Stefanos Kollias
Andreas Stafylopatis
Włodzisław Duch
Erkki Oja (Eds.)

Artificial Neural Networks – ICANN 2006

LNCS 4132

16th International Conference
Athens, Greece, September 2006
Proceedings, Part II

2 Part II

Springer

Lecture Notes in Computer Science 4132

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

Stefanos Kollias Andreas Stafylopatis
Włodzisław Duch Erkki Oja (Eds.)

Artificial Neural Networks – ICANN 2006

16th International Conference
Athens, Greece, September 10 – 14, 2006
Proceedings, Part II

Springer

Volume Editors

Stefanos Kollias
Andreas Stafylopatis
National Technical University of Athens
School of Electrical and Computer Engineering
157 80 Zographou, Athens, Greece
E-mail: {stefanos,andreas}@cs.ntua.gr

Włodzisław Duch
Nicolaus Copernicus University
Department of Informatics
ul. Grudziadzka 5, 87-100 Torun, Poland
E-mail: wduch@phys.uni.torun.pl

Erkki Oja
Helsinki University of Technology
Laboratory of Computer and Information Science
P.O. Box 5400, 02015 Hut, Finland
E-mail: erkki.oja@hut.fi

Library of Congress Control Number: 2006931797

CR Subject Classification (1998): F.1, I.2, I.5, I.4, G.3, J.3, C.2.1, C.1.3

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

ISSN 0302-9743
ISBN-10 3-540-38871-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-38871-5 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11840930 06/3142 5 4 3 2 1 0

Preface

This book includes the proceedings of the International Conference on Artificial Neural Networks (ICANN 2006) held on September 10-14, 2006 in Athens, Greece, with tutorials being presented on September 10, the main conference taking place during September 11-13 and accompanying workshops on perception, cognition and interaction held on September 14, 2006.

The ICANN conference is organized annually by the European Neural Network Society in cooperation with the International Neural Network Society, the Japanese Neural Network Society and the IEEE Computational Intelligence Society. It is the premier European event covering all topics concerned with neural networks and related areas. The ICANN series of conferences was initiated in 1991 and soon became the major European gathering for experts in these fields.

In 2006 the ICANN Conference was organized by the Intelligent Systems Laboratory and the Image, Video and Multimedia Systems Laboratory of the National Technical University of Athens in Athens, Greece.

From 475 papers submitted to the conference, the International Program Committee selected, following a thorough peer-review process, 208 papers for publication and presentation to 21 regular and 10 special sessions. The quality of the papers received was in general very high; as a consequence, it was not possible to accept and include in the conference program many papers of good quality.

A variety of topics constituted the focus of paper submissions. In regular sessions, papers addressed topics such as learning algorithms, hybrid architectures, neural dynamics and complex systems, self-organization, computational neuroscience, connectionist cognitive science, neural control, robotics and planning, data analysis, signal and time series processing, image and vision analysis, pattern recognition and applications to bioinformatics, market analysis and other real-world problems.

Special sessions, organized by distinguished researchers, focused on significant aspects of current neural network research, including cognitive machines, Semantic Web technologies and multimedia analysis, bridging the semantic gap in multimedia machine learning approaches, feature selection and dimension reduction for regression, learning random neural networks and stochastic agents, visual attention algorithms and architectures for perceptional understanding and video coding, neural computing in energy engineering, bio-inspired neural network on-chip implementation and applications, computational finance and economics.

Prominent lecturers provided key-note speeches for the conference. Moreover, tutorials were given by well-known researchers. John Taylor was the honorary Chair of the conference.

Three post-conference workshops, on intelligent multimedia, semantics, interoperability and e-culture, on affective computing and interaction and on cognitive machines, concluded the focus of ICANN 2006 on the state-of-the-art research on neural networks and intelligent technologies in relation to the domains of cognition, perception and interaction. In-depth discussion was made on the prospects and future

developments of the theoretical developments and applications of neural network models, algorithms and systems in the fields of cognition, neurobiology, semantics, perception and human computer interaction.

We would like to thank all members of the organizing laboratories for their contribution to the organization of the conference. In particular we wish to thank Lori Malatesta and Eleni Iskou, who greatly helped in handling a variety of technical and administrative problems related to the conference organization. Finally, we wish to thank Alfred Hofmann and Christine Guenther from Springer for their help and collaboration in the publication of the ICANN proceedings.

July 2006 Stefanos Kollias, Andreas Stafylopatis

Organization

General Chair

Stefanos Kollias,
National Technical University of Athens

Co-Chair

Andreas Stafylopatis, NTUA, Greece

Program Chair

Wlodzislaw Duch, Torun, Poland and Singapore
ENNS President
Erkki Oja, Helsinki, Finland

Honorary Chair

John G. Taylor, Kings College, London, UK; ENNS Past President

International Program Committee

- **Hojat Adeli,** Ohio State University, USA
- **Peter Andras,** University of Newcastle, UK
- **Marios Angelides,** Brunel University, UK
- **Panos Antsaklis,** University of N. Dame, USA
- **Bruno Apolloni,** University of Milan, Italy
- **Nikolaos Bourbakis,** Wright State University, USA
- **Peter Erdi,** University of Budapest, Hungary and Kalamazoo
- **Georg Dorffner,** University of Vienna, Austria
- **Patrick Gallinari,** Université Paris 6, France
- **Christophe Garcia,** France Telecom
- **Erol Gelenbe,** Imperial College, UK
- **Stan Gielen,** University of Nijmegen, The Netherlands
- **Pascal Hitzler,** University of Karlsruhe, Germany
- **Nikola Kasabov,** Kedri, Australia, New Zealand
- **Janusz Kacprzyk,** Warsaw, Poland
- **Okyay Kaynak,** Bogazici University, Turkey
- **Chris Koutsougeras,** Tulane University, USA
- **Thomas Martinetz,** University of Luebeck, Germany
- **Evangelia Micheli-Tzanakou,** Rutgers University, USA

- **Lars Niklasson,** Skøvde University, Sweden
- **Andreas Nuernberger,** University of Magdeburg, Germany
- **Marios Polycarpou,** University of Cyprus
- **Demetris Psaltis,** Caltech, USA
- **Branimir Reljin,** University of Belgrade, Serbia
- **Olli Simula,** Technical University of Helsinki, Finland
- **Alessandro Sperduti,** University of Padova, Italy
- **Lefteris Tsoukalas,** Purdue University, USA
- **Michel Verleysen,** Louv.-la-Neuve, Belgium
- **Alessandro Villa,** University of Grenoble, France

Local Organizing Committee

- **Yannis Avrithis,** ICCS-NTUA
- **Christos Douligeris,** Piraeus University
- **George Dounias,** Aegean University
- **Kostas Karpouzis,** ICCS-NTUA
- **Aris Likas,** University of Ioannina
- **Konstantinos Margaritis,** University of Macedonia
- **Vassilis Mertzios,** DUTH
- **Stavros Perantonis,** NCSR Demokritos
- **Yannis Pitas,** AUTH, Salonica
- **Costas Pattichis,** University of Cyprus
- **Apostolos Paul Refenes,** AUEB
- **Christos Schizas,** University of Cyprus
- **Giorgos Stamou,** ICCS-NTUA
- **Sergios Theodoridis,** UoA
- **Spyros Tzafestas,** NTUA
- **Nicolas Tsapatsoulis,** University of Cyprus
- **Mihalis Zervakis,** TUC, Crete

Reviewers

Abe	Shigeo	Kobe University
Adamczak	Rafal	Nicholas Copernicus University
Aiolli	Fabio	University of Pisa
Akrivas	George	National Technical University of Athens
Albrecht	Andreas	University of Hertfordshire
Alhoniemi	Esa	University of Turku
Andonie	Razvan	Central Washington University
Anguita	Davide	University of Genoa
Angulo-Bahon	Cecilio	Univ. Politecnica de Catalunya, Spain
Archambeau	Cedric	Université Catholique de Louvain
Atencia	Miguel	Universidad de Malaga

Aupetit	Michael	Commissariat à l`Energie Atomique
Avrithis	Yannis	National Technical University of Athens
Bedoya	Guillermo	Technical University of Catalonia, Spain
Bianchini	Monica	Università di Siena
Boni	Andrea	University of Trento
Caputo	Barbara	Royal Institute of Technology
Caridakis	George	National Technical University of Athens
Cawley	Gavin	University of East Anglia
Chetouani	Mohamed	Université Paris
Chortaras	Alexandros	National Technical University of Athens
Cichocki	Andrzej	RIKEN
Clady	Xavier	Université Pierre et Marie Curie
Corchado	Emilio	Applied Computational Intelligence Unit
Cottrell	Marie	Université Paris I
Crook	Nigel	Oxford Brookes University
Dablemont	Simon	Université Catholique de Louvain
Delannay	Nicolas	Université Catholique de Louvain
Derpanis	Kostas	York University
Dimitrakakis	Christos	IDIAP
Dominguez Merino	Enrique	E.T.S.I. Informatica, Spain
Dorronsoro	Jose	Universidad Autónoma de Madrid
Douligeris	Christos	Piraeus University
Dounias	George	Aegean University
Drosopoulos	Nasos	National Technical University of Athens
Duch	Wlodzislaw	Nicolaus Copernicus University
Elizondo	David	De Montfort University
Ferles	Christos	National Technical University of Athens
Flanagan	Adrian	Nokia Research Center
Francois	Damien	Université Catholique de Louvain
Fyfe	Colin	University of Paisley
Garcia-Pedrajas	Nicolas	University of Cordoba
Gas	Bruno	LISIF-UPMC
Gonzales Abril	Luis	Faculdad Ciencias Economicas y Empresari
Goser	Karl	Universitaet Dortmund
Gosselin	Bernard	Faculté Polytechnique de Mons
Grana	Manuel	Univ. Pais Vasco
Grothmann	Ralph	University of Bremen
Hammer	Barbara	University of Osnabrueck
Haschke	Robert	Bielefeld University
Hatziargyriou	Nikos	National Technical Univesity of Athens
Heidemann	Gunther	Bielefeld University
Hollmen	Jaakko	Technical University of Helsinki

Honkela	Antti	Helsinki University of Technology
Hryniewicz	Olgierd	Systems Research Institute PAS
Huang	Di	City University of Hong Kong
Huang	Te-Ming	The University of Auckland
Huelse	Martin	Fraunhofer Institut
Igel	Christian	Ruhr-Universitaet Bochum
Indiveri	Giacomo	UNI-ETH Zurich
Isasi	Pedro	Universidad Carlos III de Madrid
Ishii	Shin	Nara Institute of Science and Technology
Ito	Yoshifusa	Aichi-Gakuin University
Jirina	Marcel	Acad. of Sciences of the Czech Republic
Kaban	Ata	University of Birmingham
Kalveram	Karl Theodor	Institute of Experimental Psychology
Karpouzis	Kostas	ICCS-NTUA
Kasderidis	Stathis	Institute of Computer Science - FORTH
Kim	DaeEun	Max Planck Institute for Psychological Research
Kollias	Stefanos	National Technical University of Athens
Korbicz	Jozef	UZG
Koronacki	Jacek	IPI PAN
Koskela	Markus	Technical University of Helsinki
Kosmopoulos	Dimitris	National Centre for Scientific Research
Kounoudes	Anastasios	SignalGenerix Ltd
Kouropteva	Olga	University of Oulu
Kurfess	Franz	California Polytechnic State University
Kurzynski	Marek	Wroclaw University of Technology
Laaksonen	Jorma	Technical University of Helsinki
Lang	Elmar	University of Regensburg
Leclercq	Edouard	Université du Havre
Lee	John	Université Catholique de Louvain
Lehtimaki	Pasi	Helsinki University of Technology
Leiviska	Kauko	University of Oulu
Lendasse	Amaury	Helsinki University of Technology
Likas	Aris	University of Ioannina
Loizou	Christos	Intercollege, Limassol Campus
Madrenas	Jordi	Technical University of Catalunya
Malatesta	Lori	National Technical Univesity of Athens
Mandziuk	Jacek	Warsaw University of Technology
Marchiori	Elena	Vrije Universiteit Amsterdam
Marcu	Teodor	University of Duisburg-Essen
Markellos	Raphael	Athens University of Economics and Business
Markowska-Kaczmar	Urszula	Wroclaw University of Technology

Martin-Merino	Manuel	University Pontificia of Salamanca
Masulli	Francesco	Polo Universitario di La Spezia G.Marco
Micheli	Alessio	University of Pisa
Morra	Lia	Politecnico di Torino
Moutarde	Fabien	Ecole des Mines de Paris
Mueller	Klaus-Robert	University of Potsdam
Muresan	Raul	SC. NIVIS SRL
Nakayama	Minoru	CRADLE
Nikolopoulos	Konstantinos	Lancaster University Management School
Ntalianis	Klimis	National Technical University of Athens
Oja	Erkki	Helsinki University of Technology
Olteanu	Madalina	Université Paris 1
Ortiz Boyer	Domingo	University of Cordoba
Osowski	Stanislaw	Warsaw University of Technology
Parra	Xavier	Technical University of Catalonia
Pateritsas	Christos	National Technical University of Athens
Pattichis	Marios	University of New Mexico
Pattichis	Costas	University of Cyprus
Paugam-Moisy	Helene	Institut des Sciences Cognitives
Pedreira	Carlos	Catholic University of Rio de Janeiro
Pelckmans	Kristiaan	K.U.Leuven
Perantonis	Stavros	NCSR Demokritos
Pertselakis	Minas	National Technical University of Athens
Peters	Gabriele	Universitaet Dortmund
Piegat	Andrzej	Uniwersytet Szczecinski
Pitas	Yannis	Aristotle University of Thessaloniki
Polani	Daniel	University of Hertfordshire
Porrmann	Mario	Heinz Nixdorf Institute
Prevost	Lionel	Lab. Instr. et Systèmes d'Ile de France
Prevotet	Jean-Christophe	Université Pierre et Marie Curie, Paris
Raivio	Kimmo	Helsinki University of Technology
Raouzeou	Amaryllis	National Technical University of Athens
Rapantzikos	Konstantinos	National Technical University of Athens
Refenes	Apostolos Paul	Athens University Economics & Business
Risto	Risto	Tampere University of Technology
Rocha	Miguel	Universidade do Minho
Romariz	Alexandre	Universidade de Brasilia
Rossi	Fabrice	INRIA Rocquencourt
Rovetta	Stefano	University of Genova
Rutkowska	Danuta	Technical University of Czestochowa
Rynkiewicz	Joseph	Université Paris 1
Salojarvi	Jarkko	Technical University of Helsinki

Schrauwen	Benjamin	Universiteit Gent
Schwenker	Friedhelm	University of Ulm
Seiffert	Udo	Leibniz Institute of Plant Genetics
Sfakiotakis	Michael	Institute of Computer Science FORTH
Sierra	Alejandro	Universidad Autónoma de Madrid
Siivola	Vesa	Technical University of Helsinki
Skodras	Thanos	University of Patras
Stafylopatis	Andreas	National Technical University of Athens
Stamou	Giorgos	ICCS-NTUA
Steil	Jochen J.	University of Bielefeld
Steuer	Michal	University of the West of England
Stoilos	Giorgos	National Technical Univesity of Athens
Strickert	Marc	University of Osnabrueck
Suárez	Alberto	Universidad Autónoma de Madrid
Sugiyama	Masashi	Fraunhofer FIRST
Suykens	Johan	Katholieke Universiteit Leuven
Szczepaniak	Piotr	TUL
Tadeusiewicz	Ryszard	AGH
Tagliaferri	Roberto	Univ. Salerno
Taylor	John	King's College London
Terra	Marco	University of Sao Paulo
Theodoridis	Sergios	UoA
Tomas	Ana Maria	Universidade Aveiro
Trentin	Edmondo	Università di Siena
Tsakiris	Dimitris	University of Crete
Tsapatsoulis	Nicolas	University of Cyprus
Tsotsos	John	York University
Tzouvaras	Vassilis	National Technical Univesity of Athens
Usui	Shiro	RIKEN
Van Looy	Stijn	Universiteit Gent
Vannucci	Marco	Scuola Superiore Sant`Anna
Venetis	Anastassios	National Technical Univesity of Athens
Venna	Jarkko	Helsinki University of Technology
Verbeek	Jakob	University of Amsterdam
Viet	Nguyen Hoang	Polish Acacdemy of Sciences
Villmann	Thomas	Clinic for Psychotherapy
Vitay	Julien	INRIA
Wallace	Manolis	National Technical Univesity of Athens
Watanabe	Norifumi	Keio University
Wennekers	Thomas	University of Plymouth
Wiegerinck	Wim	Radboud University Nijmegen
Wira	Patrice	Universitede Haute-Alsace

Wyns	Bart	Ghent University
Yang	Zhijun	University of Edinburgh
Yearwood	John	University of Ballarat
Zervakis	Mihalis	TUC
Zimmermann	Hans-Georg	Siemens AG

Table of Contents – Part II

Neural Networks, Semantic Web Technologies and Multimedia Analysis (Special Session)

Bridging the Semantic Gap in Multimedia Machine Learning Approaches (Special Session)

Signal and Time Series Processing (I)

Signal and Time Series Processing (II)

Data Analysis (I)

Data Analysis (II)

Pattern Recognition

Visual Attention Algorithms and Architectures for Perceptional Understanding and Video Coding (Special Session)

Vision and Image Processing (I)

Vision and Image Processing (II)

Computational Finance and Economics (Special Session)

Neural Computing in Energy Engineering (Special Session)

Applications to Biomedicine and Bioinformatics

Applications to Security and Market Analysis

Real World Applications (I)

Real World Applications (II)

Table of Contents – Part I

Feature Selection and Dimension Reduction for Regression (Special Session)

Learning Algorithms (I)

Learning Algorithms (II)

Advances in Neural Network Learning Methods (Special Session)

Ensemble Learning

Learning Random Neural Networks and Stochastic Agents (Special Session)

Hybrid Architectures

Self Organization

Connectionist Cognitive Science

Cognitive Machines (Special Session)

Neural Dynamics and Complex Systems

Computational Neuroscience

Neural Control, Reinforcement Learning and Robotics Applications

Robotics, Control, Planning

Bio-inspired Neural Network On-Chip Implementation and Applications (Special session)

The Core Method: Connectionist Model Generation

Sebastian Bader* and Steffen Hölldobler

International Center for Computational Logic
Technische Universität Dresden
01062 Dresden, Germany
Sebastian.Bader@inf.tu-dresden.de, sh@iccl.tu-dresden.de

Abstract. Knowledge based artificial networks networks have been applied quite successfully to propositional knowledge representation and reasoning tasks. However, as soon as these tasks are extended to structured objects and structure-sensitive processes it is not obvious at all how neural symbolic systems should look like such that they are truly connectionist and allow for a declarative reading at the same time. The core method aims at such an integration. It is a method for connectionist model generation using recurrent networks with feed-forward core. After an introduction to the core method, this paper will focus on possible connectionist representations of structured objects and their use in structure-sensitive reasoning tasks.

1 Introduction

From the very beginning artificial neural networks have been related to propositional logic. McCulloch-Pitts networks are finite automata and vice versa [22]. Finding a global minima of the energy function modelling a symmetric network corresponds to finding a model of a propositional logic formula and vice versa [23]. These are just two examples that illustrate what McCarthy has called a *propositional fixation* of connectionist systems in [21].

On the other hand, there have been numeruous attempts to model first-order fragments in connectionist systems. In [3] energy minimization was used to model inference processes involving unary relations. In [19] and [27] multi-place predicates and rules over such predicates are modelled. In [16] a connectionist inference system for a limited class of logic programs was developed. But a deeper analysis of these and other systems reveals that the systems are in fact propositional. Recursive auto-associative memories based on ideas first presented in [25], holographic reduced representations [24] or the networks used in [9] have considerable problems with deeply nested structures. We are unaware of any connectionist system that fully incorporates structured objects and structure-sensitive processes and, thus, naturally incorporates the power of symbolic computation as argued for in e.g. [28].

* The first author is supported by the GK334 of the German Research Foundation.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 1–13, 2006.

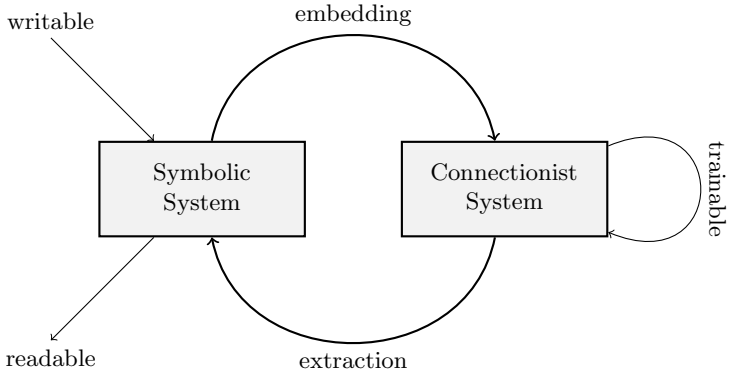


Fig. 1. The Neural-Symbolic Cycle

In this paper we are mainly interested in knowledge based artificial neural networks, i.e., networks which are initialized by available background knowledge before training methods are applied. In [29] it has been shown that such networks perform better than purely empirical and hand-built classifiers. [29] used background knowledge in the form of propositional rules and encodes these rules in multi-layer feed-forward networks. Independently, we have developed a connectionist system for computing the least model of propositional logic programs if such a model exists [14]. This system has been further developed to the so-called *core method*: background knowledge represented as logic programs is encoded in a feed-forward network, recurrent connections allow for a computation or approximation of the least model of the logic program (if it exists), training methods can be applied to the feed-forward kernel in order to improve the performance of the network, and, finally, an improved program can be extracted from the trained kernel closing the neural-symbolic cycle as depicted in Fig. 1.

In this paper we will present the core method in Section 3. In particular, we will discuss its propositional version including its relation to [29] and its extensions. The main focus of this paper will be on extending the core method to deal with structured objects and structure-sensitive processes in Section 4. In particular, we will give a feasability result, present a first practical implementation, and discuss preliminary experimental data. These main sections are framed by introducing basic notions and notations in Section 2 and an outlook in Section 5.

2 Preliminaries

We assume the reader to be familiar with basic notions from artificial neural networks and logic programs and refer to e.g. [4] and [20], resp. Nevertheless, we repeat some basic notions.

A *logic program* is a finite set of *rules* $H \leftarrow L_1 \wedge \cdots \wedge L_n$, where H is an atom and each L_i is a literal. H and $L_1 \wedge \cdots \wedge L_n$ are called the *post-* and *precondition* of

$$\begin{aligned}\mathcal{P}_1 = \{\ & p, && \%\ p \text{ is always true.}\\ & r \leftarrow p \wedge \neg q, && \%\ r \text{ is true if } p \text{ is true and } q \text{ is false.}\\ & r \leftarrow \neg p \wedge q\ \} && \%\ r \text{ is true if } p \text{ is false and } q \text{ is true.}\end{aligned}$$

Fig. 2. A simple propositional logic program. The intended meaning of the rules is given on the right.

the rule, resp. Fig. 2 and 4 show a propositional and a first-order logic program, resp. These programs will serve as running examples. The knowledge represented by a logic program $\mathcal{P}$ can essentially be captured by the *meaning function* $T_\mathcal{P}$, which is defined as a mapping on the space of interpretations where for any interpretation I we have that $T_\mathcal{P}(I)$ is the set of all H for which there exists a ground instance $H \leftarrow A_1 \wedge \cdots \wedge A_m \wedge \neg B_1 \wedge \cdots \wedge \neg B_n$ of a rule in $\mathcal{P}$ such that for all i we have $A_i \in I$ and for all j we have $B_j \notin I$, where each A_i and each B_j is an atom. Fixed points of $T_\mathcal{P}$ are called *(supported) models* of $\mathcal{P}$, which can be understood to represent the declarative semantics of $\mathcal{P}$.

Artificial neural networks consist of simple computational units (neurons), which receive real numbers as inputs via weighted connections and perform *simple* operations: the weighted inputs are added and simple functions (like threshold, sigmoidal) are applied to the sum. We will consider networks, where the units are organized in layers. Neurons which do not receive input from other neurons are called *input neurons*, and those without outgoing connections to other neurons are called *output neurons*. Such so-called *feed-forward networks* compute functions from $\mathbb{R}^n$ to $\mathbb{R}^m$, where n and m are the number of input and output units, resp. Fig. 3 on the right shows a simple feed-forward network. In this paper we will construct recurrent networks by connecting the output units of a feed-forward network N to the input units of N. Fig. 3 on the left shows a blueprint of such a recurrent network.

3 The Core Method

In a nutshell, the idea behind the core method is to use feed-forward connectionist networks – called core – to compute or approximate the meaning function of logic programs. If the output layer of a core is connected to its input layer then these recurrent connections allow for an iteration of the meaning function leading to a stable state, corresponding to the least model of the logic program provided that such a least model exists (see Fig. 3 on the left). Moreover, the core can be trained using standard methods from connectionist systems. In other words, we are considering connectionst model generation using recurrent networks with feedforward core.

The ideas behind the core method were first presented in [14] for propositional logic programs (see also [13]). Consider the logic program shown in Fig. 2. A translation algorithm turns such a program into a core of logical threshold units. Because the program contains the predicate letters p, q and r only, it suffices

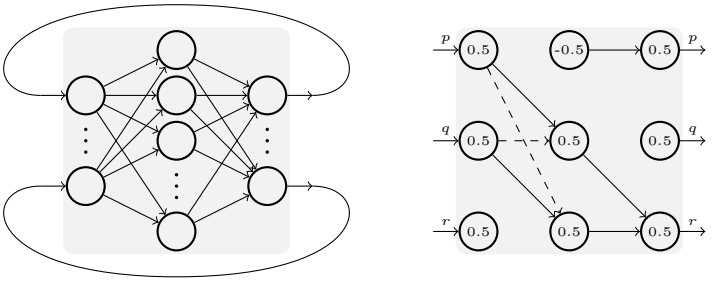


Fig. 3. The blueprint of a recurrent network used by the core method on the left. The core corresponding to $\mathcal{P}_1 = \{p,\ r \leftarrow p \wedge \neg q,\ r \leftarrow \neg p \wedge q\}$ is shown on the right. Solid connections have weight 1.0, dashed connections weight -1.0. The numbers within the units denote the thresholds.

to consider interpretations of these three letters. Such interpretations can be represented by triples of logical threshold units. The input and the output layer of the core consist exactly of such triples. For each rule of the program a logical threshold unit is added to the hidden layer such that the unit becomes active iff the preconditions of the rule are met by the current activation pattern of the input layer; moreover this unit activates the output layer unit corresponding to the postcondition of the rule. Fig. 3 on the right shows the network obtained by the translation algorithm if applied to $\mathcal{P}_1$.

In [14] we proved – among other results – that for each propositional logic program $\mathcal{P}$ there exists a core computing its meaning function $T_\mathcal{P}$ and that for each acyclic logic program $\mathcal{P}$ there exists a core with recurrent connections such that the computation with an arbitray intitial input converges and yields the unique fixed point of $T_\mathcal{P}$.

The use of logical threshold units in [14] made it easy to prove these results. However, it prevented the application of standard training methods like back-propagation to the kernel. This problem was solved in [8] by showing that the same results can be achieved if bipolar sigmoidal units are used instead (see also [5]). [8] also overcomes a restriction of the KBANN method originally presented in [29]: rules may now have arbitrarily many preconditions and programs may have arbitrarily many rules with the same postcondition.

In the meantime the propositional core method has been extended in many directions. In [18] three-valued logic programs are discussed; This approach has been extended in [26] to finitely determined sets of truth values. Modal logic programs have been considered in [6]. Answer set programming and metalevel priorities are discussed in [5]. The core method has been applied to intuitionistic logic programs in [7].

To summarize, the propositional core method allows for model generation with respect to a variety of logics in a connectionist setting. Given logic programs are translated into recurrent connectionist networks with feed-forward cores, such that the cores compute the meaning functions associated with the programs.

The cores can be trained using standard learning methods leading to improved logic programs. These improved programs must be extracted from the trained cores in order to complete the neural-symbolic cycle. The extraction process is outside the scope of this paper and interested readers are refered to e.g. [1] or [5].

4 The Core Method and Structured Objects

If structured objects and structure-sensitive processes are to be modelled, then usually higher-order logics are considered. In particular, first-order logic plays a prominent role because any computable function can be expressed by first-order logic programs. The extension of the core method to first-order logic poses a considerable problem because first-order interpretations usually do not map a finite but a countably infinite set of ground atoms to the set the truth values. Hence, they cannot be represented by a finite vector of units, each of which represents the value assigned to a particular ground atom.

In this section we will first show that an extension of the core method to first-order logic programs in feasible. However, the result will be purely theoretical and thus the question remains how cores can be constructed for first-order programs. In Subsection 4.2 a practical solution is discussed, which approximates the meaning functions of logic programs by means of piecewise constant functions. Some preliminary experimental data are presented in Subsection 4.3.

4.1 Feasibility

It is well known that multilayer feed-forward networks are universal approximators [17,12] of functions $\mathbb{R}^n \to \mathbb{R}^m$. Hence, if we find a way to represent interpretations of first-order logic programs by finite vector of real numbers, then feed-forward networks can be used to approximate the meaning function of such programs.

Consider a countably infinite set of ground atoms and assume that there is a bijection l uniquely assigning a natural number to each ground atom and vice versa; l is called *level mapping* and $l(A)$ *level* of the ground atom A. Furthermore, consider an interpretation I assigning to each ground atom A either 0 (representing falsehood) or 1 (representing truth) and let b be a natural number greater than 2. Then,

$$\iota(I) = \sum_{j=1}^{\infty} I(l^{-1}(j)) \cdot b^{-j},$$

is a real number encoding the interpretation I. With

$$\mathcal{D} = \{r \in \mathbb{R} \mid r = \sum_{j=1}^{\infty} a_j b^{-j}, a_j \in \{0, 1\}\}$$

we find that ι is a bijection between the set of all interpretions and $\mathcal{D}$. Hence, we have a sound and complete encoding of interpretations.

Let $\mathcal{P}$ be a logic program and $T_\mathcal{P}$ its associated meaning operator. We define a sound and complete encoding $f_\mathcal{P} : \mathcal{D} \to \mathcal{D}$ of $T_\mathcal{P}$ as follows:

$$f_\mathcal{P}(r) = \iota(T_\mathcal{P}(\iota^{-1}(r))).$$

In [15] we proved – among other results – that for each logic program $\mathcal{P}$ which is acylic wrt. a bijective level mapping the function $f_\mathcal{P}$ is contractive, hence continuous. This has various implications: (i) We can apply Funahashi's result, viz. that every continuous function on (a compact subset of) the reals can be uniformly approximated by feed-forward networks with sigmoidal units in the hidden layer [12]. This shows that the meaning function of a logic program (of the kind discussed before) can be approximated by a core. (ii) Considering an appropriate metric, which will be discussed in a moment, we can apply Banach's contraction mapping theorem (see e.g. [30]) to conclude that the meaning function has a unique fixed point, which is obtained from an arbitrary initial interpretation by iterating the application of the meaning function. Using (i) and (ii) we were able to prove in [15] that the least model of logic programs which are acyclic wrt. a bijective level mapping can be approximated arbitrarily well by recurrent networks with feed-forward core.

But what exactly is the approximation of an interpretion or a model in this context? Let $\mathcal{P}$ be a logic program and l a level mapping. We can define a metric d on interpretations as follows:

$$d(I,J) = \begin{cases} 0 & \text{if } I = J, \\ 2^{-n} & \text{if } n \text{ is the smallest level on which } I \text{ and } J \text{ disagree.} \end{cases}$$

As shown in [10] the set of all interpretations together with d is a complete metric space. Moreover, an interpretation I *approximates* an interpretation J *to degree* $n \in \mathbb{N}$ iff $d(I,J) \leq 2^{-n}$. In other words, if a recurrent network approximates the least model I of an acylic logic program to a degree $n \in \mathbb{N}$ and outputs $r \in \mathcal{D}$ then for all ground atoms A whose level is equal or less than n we find that $I(A) = \iota^{-1}(r)(A)$.

4.2 A First Approach

In this section, we will show how to construct a core network approximating the meaning operator of a given logic program. As above, we will consider logic programs $\mathcal{P}$ which are acyclic wrt. an bijective level mapping. We will construct sigmoidal networks and RBF networks with a raised cosine activation function. All ideas presented here can be found in detail in [2]. To illustrate the ideas, we will use the program $\mathcal{P}_2$ shown in Fig. 4 as a running example. The construction consists of five steps:

1. Construct $f_\mathcal{P}$.
2. Approximate $f_\mathcal{P}$ using a piecewise constant functions $\bar{f}_\mathcal{P}$.
3. Implement $\bar{f}_\mathcal{P}$ using (a) step and (b) triangular functions.

$$\begin{aligned}\mathcal{P}_2 = \{\ & even(0). && \%\ 0 \text{ is an even number.}\\ & even(succ(X)) \leftarrow odd(X). && \%\ \text{The successor of an odd } X \text{ is even.}\\ & odd(X) \leftarrow \neg even(X).\ \} && \%\ \text{If } X \text{ is not even then it is odd.}\end{aligned}$$

Fig. 4. The first-order logic program $\mathcal{P}_2$ describing even and odd numbers. The intended meaning of the rules is given on the right.

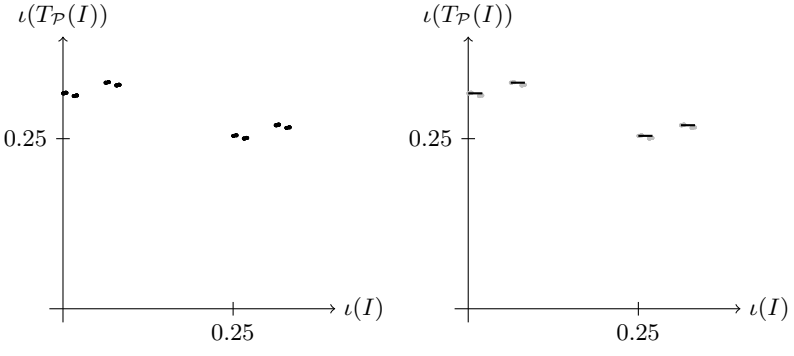


Fig. 5. On the left is the plot of $f_{\mathcal{P}_2}$. On the right a piecewise constant approximation $\bar{f}_{\mathcal{P}_2}$ (for level $n = 2$) of $f_{\mathcal{P}_2}$ is shown. The base $b = 4$ was used for the embedding.

4. Replace those by (a) sigmoidal and (b) raised cosine functions.
5. Construct the core network approximating $f_{\mathcal{P}}$.

In the sequel we will describe the ideas underlying the construction. A rigorous development including all proofs can be found in [2,31]. One should observe that $f_{\mathcal{P}}$ is a function on $\mathcal{D}$ and not on $\mathbb{R}$. Although the functions constructed below will be defined on intervals of $\mathbb{R}$, we are concerned with accuracy on $\mathcal{D}$ only.

1. Construct $f_{\mathcal{P}}$: $f_{\mathcal{P}}$ is defined as before, i.e., $f_{\mathcal{P}}(r) = \iota(T_{\mathcal{P}}(\iota^{-1}(r)))$. Fig. 5 on the left shows the plot of $f_{\mathcal{P}_2}$.

2. Constructing a Piecewise Constant Function $\bar{f}_{\mathcal{P}}$: Because $\mathcal{P}$ is acyclic, we conclude that all variables occurring in the precondition of a rule are also contained in its postcondition. Hence, for each level n we find that whenever $d(I, J) \leq 2^{-n}$ then $d(T_{\mathcal{P}}(I), T_{\mathcal{P}}(J)) \leq 2^{-n}$, where I and J are interpretations. Therefore, we can approximate $T_{\mathcal{P}}$ to degree n by some function $\bar{T}_{\mathcal{P}}$ which considers ground atoms with a level less or equal n only. As a consequence, we can approximate $f_{\mathcal{P}}$ by a piecewise constant function $\bar{f}_{\mathcal{P}}$ where each piece has a length of $\lambda = \frac{1}{(b-1)b^n}$, with b being the base used for the embedding. Fig. 5 shows $f_{\mathcal{P}_2}$ and $\bar{f}_{\mathcal{P}_2}$ for $n = 2$.

3. Implementation of $\bar{f}_{\mathcal{P}}$ using Linear Functions: As a next step, we will show how to implement $\bar{f}_{\mathcal{P}}$ using (a) step and (b) triangular functions. Those functions are the linear counterparts of the functions actually used in the networks

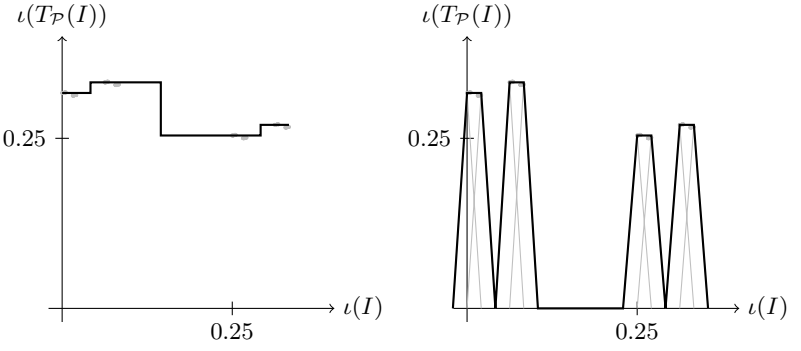


Fig. 6. Two linear approximation of $\bar{f}_{\mathcal{P}_2}$. On the left, three step functions were used; On the right, eight triangular functions (depicted in gray) add up to the approximation, which is shown using thick lines.

constructed below. If $\bar{f}_{\mathcal{P}}$ consists of k intervals, then we can implement it using $k-1$ step functions which are placed such that the steps are between two neighbouring intervals. This is depicted in Fig. 6 on the left.

Each constant piece of length λ could also be implemented using two triangular functions with width λ and centered at the endpoints. Those two triangles add up to the constant piece. For base $b = 4$, we find that the gaps between two intervals have a length of at least 2λ. Therefore, the triangular functions of two different intervals will never interfere. The triangular implementation is depicted in Fig. 6 on the right.

4. Implementation of $\bar{f}_{\mathcal{P}}$ using Nonlinear Functions: To obtain a sigmoidal approximation, we replace each step function with a sigmoidal function. Unfortunately, those add some further approximation error, which can be dealt with by increasing the accuracy in the constructions above. By dividing the desired accuracy by two, we can use one half as accuracy for the constructions so far and the other half as a margin to approximate the constant pieces by sigmoidal functions. This is possible because we are concerned with the approximation on $\mathcal{D}$ only.

The triangular functions described above can simply be replaced by raised cosine activation functions, as those add up exactly as the triangles do and do not interfere with other intervals either.

5. Construction of the Network: A standard sigmoidal core approximating the $T_{\mathcal{P}}$-operator of a given program $\mathcal{P}$ consists of:

- An input layer containing one input unit whose activation will represent an interpretation I.
- A hidden layer containing a unit with sigmoidal activation function for each sigmoidal function constructed above.
- An output layer containing one unit whose activation will represent the approximation of $T_{\mathcal{P}}(I)$.

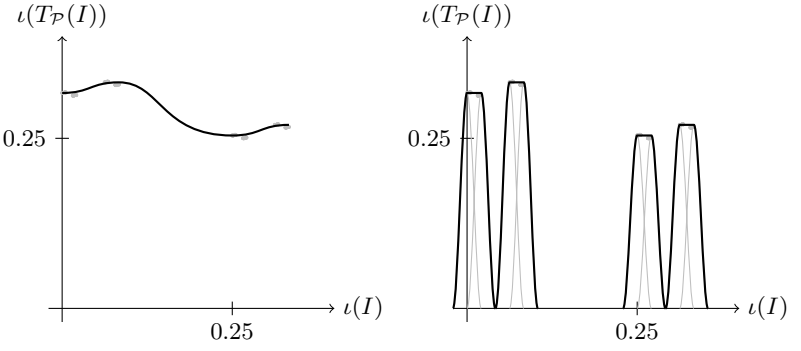


Fig. 7. Two non-linear approximation of $\bar{f}_{\mathcal{P}_2}$. On the left, sigmoidal functions were used and on the right, raised cosines.

The weights from input to hidden layer together with the bias of the hidden units define the positions of the sigmoidals. The weights from hidden to output layer represent the heights of the single functions. An RBF network can be constructed analogously, but will contain more hidden layer units, one for each raised cosine functions. Detailed constructions can be found in [2].

4.3 Evaluation and Experiments

In the previous section, we showed how to construct a core network for a given program and some desired level of accuracy. We used a one-dimensional embedding to obtain a unique real number $\iota(I)$ for a given interpretation I. Unfortunately, the precision of a real computer is limited, which implies, that using e.g. a 32-bit computer we could embed the first 16 atoms only. This limitation can be overcome by distributing an interpretation over more than one real number. In our running example $\mathcal{P}_2$, we could embed all *even*-atoms into one real number and all *odd*-atoms into another one, thereby obtaining a two-dimensional vector for each interpretation, hence doubling the accuracy. For various reasons, spelled out in [32], the sigmoidal approach described above does not work for more than one dimension. Nevertheless, an RBF network approach, similar to the one described above, does work. By embedding interpretations into higher-dimensional vectors, we can approximate meaning functions of logic programs arbitrarily well.

Together with some theoretical results, Andreas Witzel developed a prototype system in [32]. By adapting ideas from [11], he designed appropriate learning techniques utilizing the knowledge about a given domain, viz. the space of embedded interpretations. In the sequel, we will briefly present some of the results.

To adapt the networks behaviour during learning, the algorithm changes the weights, thereby changing the position and height of the constant pieces described above. Furthermore, new units are added if required, i.e., if a certain unit produces a large error, new units are added to support it. If a unit becomes inutile it will be removed from the network. These ideas are adaptations

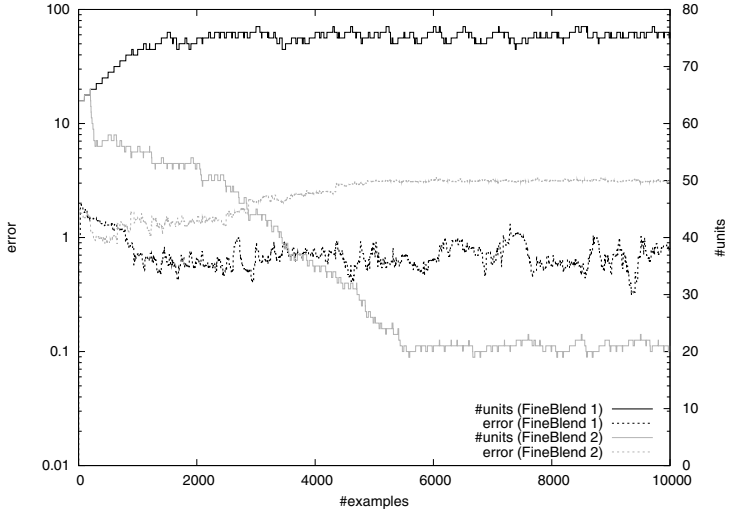


Fig. 8. Two different setups of the system during learning. Note that the error is shown on a logarithmic scale with respect to some given ε (1 means that the error is ε).

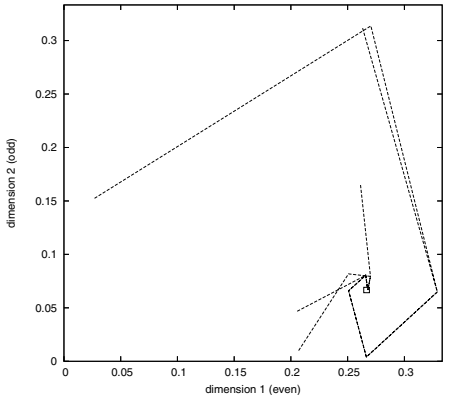


Fig. 9. Iterating random inputs

of concepts originally developed in the so called *growing neural gas* approach [11]. Fig. 8 shows a comparison of two different setups called FineBlend 1 and 2. FineBlend 1 is configured to keep the error below 1, whereas FineBlend 2 is configured to reduce the number of units resulting in a slightly higher error.

As mentioned above, a recurrent network is obtained by connecting output and input layer of the core. This is done to iterate the application of the meaning function. Therefore, we would assume a network set up and trained to represent the meaning function of an acyclic logic program to converge to a state representing the least model. As shown in Fig. 9, the network shows this behaviour.

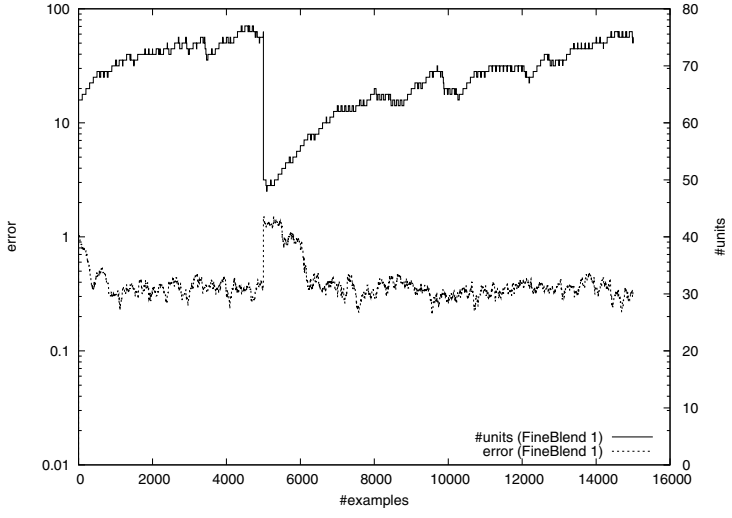


Fig. 10. The effect of unit failure. After 5000 examples, one third of the units were removed.

Shown are the two dimensions corresponding to the embedding of the even and odd predicates, resp. Also depicted is the ε-neighborhood of the least fixed point as a small square. Five random inputs were presented to the network and the output fed back via the recurrent connections. This process was repeated until the network reached a stable state, always being within the ϵ-neighbourhood of the fixed point.

Another advantage of connectionist systems is their robustness and their capability of repairing damage by further training. Fig. 10 shows the effect of unit failure. After presenting 5000 training samples to the network, one third of the hidden layer units were removed. As shown in the error plot, the system was able to recover quickly, thereby demonstrating its robustness. Further experiments and a more detailed analysis of the system can be found in [32,2].

5 Conclusion

We are currently implementing the first-order core method in order to further evaluate and test it using real world examples. Concerning a complete neural-symbolic cycle we note that whereas the extraction of propositional rules from trained networks is well understood, the extraction of first-order rules is an open question.

Acknowledgements. Many thanks to Sven-Erik Bornscheuer, Artur d'Avila Garcez, Pascal Hitzler, Yvonne McIntyre (formerly Kalinke), Anthony K. Seda, Hans-Peter Störr, Andreas Witzel and Jörg Wunderlich who all contributed to the core method.

References

1. R. Andrews, J. Diederich, and A. Tickle. A survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge–Based Systems*, 8(6), 1995.
2. S. Bader, P. Hitzler, and A. Witzel. Integrating first-order logic programs and connectionist systems — a constructive approach. In *Proceedings of the IJCAI-05 Workshop on Neural-Symbolic Learning and Reasoning, NeSy'05, Edinburgh, UK*, 2005.
3. D. H. Ballard. Parallel logic inference and energy minimization. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 203 – 208, 1986.
4. Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
5. A.S. d'Avila Garcez, K. Broda, and D.M. Gabbay. *Neural-Symbolic Learning Systems: Foundations and Applications*. Springer, 2002.
6. A.S. d'Avila Garcez, L. C. Lamb, and D.M. Gabbay. A connectionist inductive learning system for modal logic programming. In *Proceedings of the IEEE International Conference on Neural Information Processing ICONIP'02*, Singapore, 2002.
7. A.S. d'Avila Garcez, L.C. Lamb, and D.M. Gabbay. Neural-symbolic intuitionistic reasoning. In *Design and Application of Hybrid Intelligent Systems*, pages 399–408, IOS Press, 2003.
8. A.S. d'Avila Garcez, G. Zaverucha, and L.A.V. de Carvalho. Logic programming and inductive learning in artificial neural networks. In Ch. Herrmann, F. Reine, and A. Strohmaier, editors, *Knowledge Representation in Neural Networks*, pages 33–46, Berlin, 1997. Logos Verlag.
9. J. L. Elman. Structured representations and connectionist models. In *Proceedings of the Annual Conference of the Cognitive Science Society*, pages 17–25, 1989.
10. M. Fitting. Metric methods – three examples and a theorem. *Journal of Logic Programming*, 21(3):113–127, 1994.
11. B. Fritzke. *Vektorbasierte Neuronale Netze*. Shaker Verlag, 1998.
12. K.-I. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183–192, 1989.
13. P. Hitzler, S. Hölldobler, and A.K. Seda. Logic programs and connectionist networks. *Journal of Applied Logic*, 2(3):245–272, 2004.
14. S. Hölldobler and Y. Kalinke. Towards a massively parallel computational model for logic programming. In *Proceedings of the ECAI94 Workshop on Combining Symbolic and Connectionist Processing*, pages 68–77. ECCAI, 1994.
15. S. Hölldobler, Y. Kalinke, and H.-P. Störr. Approximating the semantics of logic programs by recurrent neural networks. *Applied Intelligence*, 11:45–59, 1999.
16. S. Hölldobler and F. Kurfess. CHCL – A connectionist inference system. In B. Fronhöfer and G. Wrightson, editors, *Parallelization in Inference Systems*, pages 318 – 342. Springer, LNAI *590*, 1992.
17. K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.
18. Y. Kalinke. Ein massiv paralleles Berechnungsmodell für normale logische Programme. Master's thesis, TU Dresden, Fakultät Informatik, 1994. (in German).
19. T. E. Lange and M. G. Dyer. High-level inferencing in a connectionist network. *Connection Science*, 1:181 – 217, 1989.
20. J. W. Lloyd. *Foundations of Logic Programming*. Springer, Berlin, 1993.

21. J. McCarthy. Epistemological challenges for connectionism. *Behavioural and Brain Sciences*, 11:44, 1988.
22. W. S. McCulloch and W. Pitts. A logical calculus and the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.
23. G. Pinkas. Symmetric neural networks and logic satisfiability. *Neural Computation*, 3:282–291, 1991.
24. T. A. Plate. Holographic reduced networks. In C. L. Giles, S. J. Hanson, and J. D. Cowan, editors, *Advances in Neural Information Processing Systems 5*. Morgan Kaufmann, 1992.
25. J. B. Pollack. Recursive distributed representations. *Artificial Intelligence*, 46:77–105, 1990.
26. A.K. Seda and M. Lane. Some aspects of the integration of connectionist and logic-based systems. In *Proceedings of the Third International Conference on Information*, pages 297–300, International Information Institute, Tokyo, Japan, 2004.
27. L. Shastri and V. Ajjanagadde. From associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony. *Behavioural and Brain Sciences*, 16(3):417–494, September 1993.
28. P. Smolensky. On variable binding and the representation of symbolic structures in connectionist systems. Technical Report CU-CS-355-87, Department of Computer Science & Institute of Cognitive Science, University of Colorado, Boulder, CO 80309-0430, 1987.
29. G.G. Towell and J.W. Shavlik. Extracting refined rules from knowledge–based neural networks. *Machine Learning*, 131:71–101, 1993.
30. S. Willard. *General Topology*. Addison–Wesley, 1970.
31. A. Witzel. Integrating first-order logic programs and connectionist networks. Project Thesis, Technische Universität Dresden, Informatik, 2005.
32. A. Witzel. Neural-symbolic integration – constructive approaches. Master's thesis, Technische Universität Dresden, Informatik, 2006.

A Neural Scheme for Robust Detection of Transparent Logos in TV Programs

Stefan Duffner and Christophe Garcia

France Telecom Division Research & Development,
4, rue du Clos Courtel, 35512 Cesson-Sévigné, France
{stefan.duffner, christophe.garcia}@francetelecom.com

Abstract. In this paper, we present a connectionist approach for detecting and precisely localizing transparent logos in TV programs. Our system automatically synthesizes simple problem-specific feature extractors from a training set of logo images, without making any assumptions or using any hand-made design concerning the features to extract or the areas of the logo pattern to analyze. We present in detail the design of our architecture, our learning strategy and the resulting process of logo detection. We also provide experimental results to illustrate the robustness of our approach, that does not require any local preprocessing and leads to a straightforward real time implementation.

1 Introduction

In the last decade, we have entered the digital era, with the convergence of telecommunication, video and informatics. Our society (press agencies, television channels, customers) is producing daily extremely large and increasing amounts of digital images and videos, making it more and more difficult to track and access this content, with traditional database search engines, requiring tedious manual annotation of keywords or comments. Therefore, automatic content-based indexing has become one of the most important and challenging issues for the years to come, in order to face the limitations of traditional information systems. Some expected applications are [6,7,9]: Information and entertainment, video production and distribution, professional video archive management including legacy footages, teaching, training, enterprise or institutional communication, TV program monitoring, self-produced content management, internet search engines and video conference archiving and management.

The recent progresses in the field of object detection and recognition tend to make possible a large range of applications that require accessing the semantic content and identifying high-level indices, regardless of the global context of the image, in order to ease automatic indexing and provide more intuitive formulation of user requests. For instance, human face detection can now be considered as a very mature tool, even though progresses have still to be made for full-profile view detection and accurate facial feature detection, for allowing robust face recognition. Recently, Garcia and Delakis [2] proposed a near-real time neural-based face detection scheme, named "Convolutional Face Finder"

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 14–23, 2006.

(CFF) that has been designed to precisely locate multiple faces of minimum size 20x20 pixels and variable appearance, rotated up to 30 degrees in image plane and turned up to 60 degrees, in complex real world images. A detection rate of 90.3% with 8 false positives have been reported on the CMU test set, which are the best results published so far on this test set. Locating a face and recognizing it [10] tend to appear as a required functionality in state-of-the-art systems, working with professional videos or personal digital image collections. Another important expected functionality is superimposed text detection and recognition.

Even though lots of progresses have been recently made in the field of object detection and recognition, most approaches have focused on image of objects variable in scale and orientation, but with small variation in shape or global gray level appearance. There is a still a lot to be done in the case of deformable 3D object detection but also in the case of very variable object texture appearance.

In this paper, we will focus on the specific case of transparent object detection in images, which is a very challenging problem. We propose a general solution that will be evaluated on the problem of transparent logo detection in video programs. For illustration purposes, we will focus on the detection of the logo of the France 2 television channel (FR2 logo), as shown in Fig. 1. Note that the proposed method is very generic and can be applied to other logos in a straightforward way. Most logo detection approach consider opaque logos superimposed on video frames. In that case, pixels inside the logo boundaries keep approximately the same values from one frame to the next, with a certain amount of noise due to video coding. Only pixels outside the logo boundaries are variable. In the case of transparent logo, pixels inside the logo boundaries also change depending on the video underneath. If temporal constancy of pixel inside opaque logo can ease the detection process, by temporal gradient analysis [4] or low level based pattern matching techniques [4,1,11], this is not the case for transparent logos, where all pixels strongly vary at the same time depending on the background.

To face this challenge, we propose an image-based approach that is designed to precisely detect transparent patterns of variable size, in complex real world video images. Our system is based on a convolutional neural network architecture [3], directly inspired from our face detector, the Convolutional Face Finder (CFF) described in [2]. It automatically derives problem-specific feature extractors, from a large training set of logo and non-logo patterns, without making any assumptions about the features to extract or the areas of the logo patterns to analyze. Once trained, our system acts like a fast pipeline of simple convolutions and subsampling modules, that treat the raw input image as a whole, for each analyzed scale, and does not require any costly local preprocessing before classification. Such a scheme provides very high detection rate with a particularly low level of false positives, demonstrated on difficult videos, maintaining a near real time processing speed.

The remainder of the paper is organized as follows. In section 2, we describe the architecture of the proposed transparent logo detection system. In sections

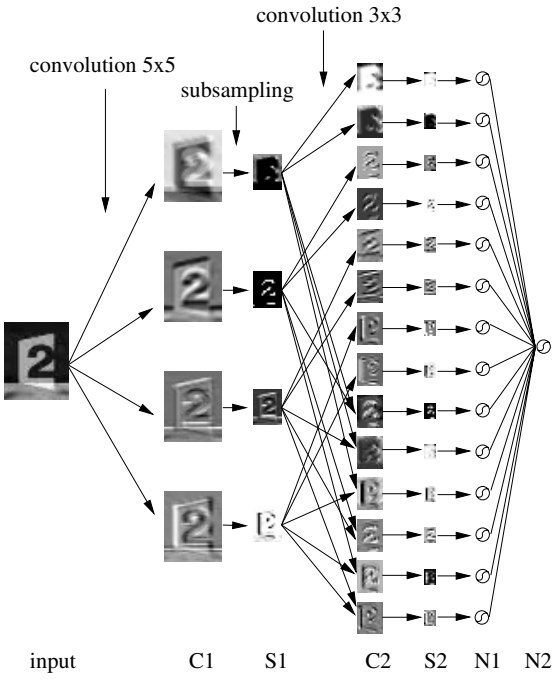


Fig. 1. The convolutional architecture

3 and 4, we explain in detail the way we train and apply the built detector. In section 5, we assess the performance of our approach by analyzing its precision. Some experimental results obtained on images of complex scenes are also presented to demonstrate the effectiveness and the robustness of the proposed approach. Finally, conclusions are drawn.

2 System Architecture

The convolutional neural network, shown in Fig. 1, consists of a set of three different kinds of layers. Layers Ci are called convolutional layers, which contain a certain number of planes. Layer C1 is connected to the retina, receiving the image area to classify as logo or non-logo. Each unit in a plane receives input from a small neighborhood (biological local receptive field) in the planes of the previous layer. The trainable weights (convolutional mask) forming the receptive field for a plane are forced to be equal at all points in the plane (weight sharing). Each plane can be considered as a feature map that has a fixed feature detector that corresponds to a pure convolution with a trainable mask, applied over the planes in the previous layer. A trainable bias is added to the results of each convolutional mask. Multiple planes are used in each layer so that multiple features can be detected.

Once a feature has been detected, its exact location is less important. Hence, each convolutional layer Ci is typically followed by another layer Si that performs local averaging and subsampling operations. More precisely, a local averaging over a neighborhood of four inputs is performed followed by a multiplication by a trainable coefficient and the addition of a trainable bias. This subsampling operation reduces by two the dimensionality of the input and increases the degrees of invariance to translation, scale, and deformation of the learnt patterns.

The different parameters governing the proposed architecture, i.e., the number of layers, the number of planes and their connectivity, as well as the size of the receptive fields, have been experimentally chosen. Practically, different architectures have been iteratively built, trained, and tested over training sets. We retained the architecture that performed efficiently in terms of good detection rates and especially in terms of false alarm rejection, while still containing an acceptable number of free parameters.

Layers C1 and C2 perform convolutions with trainable masks of dimension 5x5 and 3x3 respectively. Layer C1 contains four feature maps and therefore performs four convolutions on the input image. Layers S1 and C2 are partially connected. Mixing the outputs of feature maps helps in combining different features, thus in extracting more complex information. In our system, layer C2 has 14 feature maps. Each of the four subsampled feature maps of S1 is convolved by two different trainable masks 3x3, providing eight feature maps in C2. The other six feature maps of C2 are obtained by fusing the results of two convolutions on each possible pair of feature maps of S1. Layers N1 and N2 contain simple sigmoid neurons. The role of these layers is to perform classification, after feature extraction and input dimensionality reduction are performed. In layer N1, each neuron is fully connected to exactly one feature map of layer S2. The unique neuron of layer N2 is fully connected to all the neurons of the layer N1. The output of this neuron is used to classify the input image as logo or non-logo. For training the network, we used the classical backpropagation algorithm with momentum modified for being used in convolutional networks as described in [3]. Desired responses are set to -1 for non-logo and to +1 for logo.

In our system, the dimension of the retina is 38x46. Because of weight sharing, the network has only 1147 trainable parameters. Local receptive fields, weight sharing and subsampling provide many advantages to solve two important problems at the same time: the problem of robustness and the problem of good generalization, which is critical given the impossibility of gathering in one finite-sized training set all the possible variations of the logo pattern. This topology has another decisive advantage. In order to search for a specific pattern, the network must be replicated (or scanned) at all locations in the input image, as classically done in detection approaches [5,8]. In our approach, since each layer essentially performs a convolution with a small-size kernel, a very large part of the computation is in common between two neighboring logo window locations in the input images. This redundancy is naturally eliminated by performing

Fig. 2. Some samples of the training set. The last row shows initial negative examples.

the convolutions corresponding to each layer on the entire input image at once. The overall computation amounts to a succession of convolutions and non-linear transformations over the entire images.

3 Training Methodology

The FR2 logo examples used to train the network were collected from various video segments, during a 12 hour broadcast of the FR2 TV channel. Some of the $1,993$ collected FR2 logo images are shown in the first row of Fig. 2. Collecting a representative set of non-logos is more difficult as virtually any random image could belong to it. A practical solution to this problem consists in a bootstrapping strategy [8], in which the system is iteratively re-trained with false alarms produced when applied to a set of video images, that do not contain the targeted logo. In the proposed approach, we improved this strategy. Before proceeding with the bootstrapping, an initial training set of $2,313$ non-logo patterns was built by randomly cropping images from video frames. Some non-logo patterns (negative examples) are shown in Fig. 2. The proposed bootstrapping procedure is presented in table 1. In step 1, a validation set is built and used for testing the generalization ability of the network during learning and, finally, selecting the

Table 1. The proposed bootstrapping scheme

1. Create a validation set of 400 logo images and 400 non-logo images randomly extracted and excluded from the initial training set. It will be used to choose the best performing weight configuration during steps 3 and 8.
2. Set $BIter = 0,\ ThrFa = 0.8$.
3. Train the network for 60 learning epochs. Use an equal number of positive and negative examples in each epoch. Set $BIter = BIter + 1$.
4. Gather false alarms from a set of 300 video frames with network answers above $ThrFa$. Collect at maximum $5,000$ new examples.
5. Concatenate the newly created examples to the non-logo training set.
6. If $ThrFa \geq 0.2$ set $ThrFa = ThrFa - 0.2$.
7. If $BIter < 6$ go to step 3.
8. Train the network for 60 more learning epochs and exit.

weight configuration that performs best on it. This validation set is kept constant through all the bootstrapping iterations, in contrast with the training set which is updated. In step 3, the backpropagation algorithm is used with the addition of a momentum term for neurons belonging to the N1 and N2 layers. Stochastic learning was preferred versus batch learning. For each learning epoch, an equal number of examples from both classes are presented to the network giving no bias toward one of the two classes.

The generation of the new patterns that will be added to the non-logo training set is carried out by step 4. The false alarms produced in this step force the network, in the next iteration, to refine its current decision boundary for the FR2 logo class. At each iteration, the false alarms, giving network answers greater than *ThrFa*, and therefore strongly misclassified, are selected. As the network generalizes from these examples, *ThrFa* is gradually reduced until reaching 0. In this way, some redundancy is avoided in the training set. The learning process is stopped after six iterations, when convergence is noticed, i.e. when the number of false alarms remains roughly constant. This procedure helps in correcting problems arising in the original algorithm proposed in [8] where false alarms were grabbed regardless of the strength of the network answers. Finally, the controlled bootstrapping process added around $21,000$ non FR2 logo examples to the training set.

4 Logo Localization

Fig. 3. depicts the process of logo localization. In order to detect FR2 logo patterns of different sizes, the input image is repeatedly subsampled via a factor of 1.2, resulting in a pyramid of images.

As mentioned earlier, each image of the pyramid is entirely convolved at once by the network. For each image of the pyramid, an image containing the network results is obtained. Because of the successive convolutions and subsampling operations, this image is approximately four times smaller than the original one. This fast procedure may be seen as corresponding to the application of the network retina at every location of the input image with a step of four pixels in both axis directions, without computational redundancy.

After processing by this detection pipeline, logo candidates (pixels with positive values in the result image) in each scale are mapped back to the input image scale (step 3). They are then grouped according to their proximity in image and scale spaces. Each group of logo candidates is fused in a representative logo whose center and size are computed as the centroids of the centers and sizes of the grouped logos, weighted by their individual network responses. After applying this grouping algorithm, the set of remaining representative logo candidates serve as a basis for the next stage of the algorithm, in charge of fine logo localization and eventually false alarm dismissal.

To do so, a local search procedure is performed in an area around each logo candidate center in image scale-space (step 4). A reduced search space centered at the logo candidate position is defined in image scale-space for precise localization

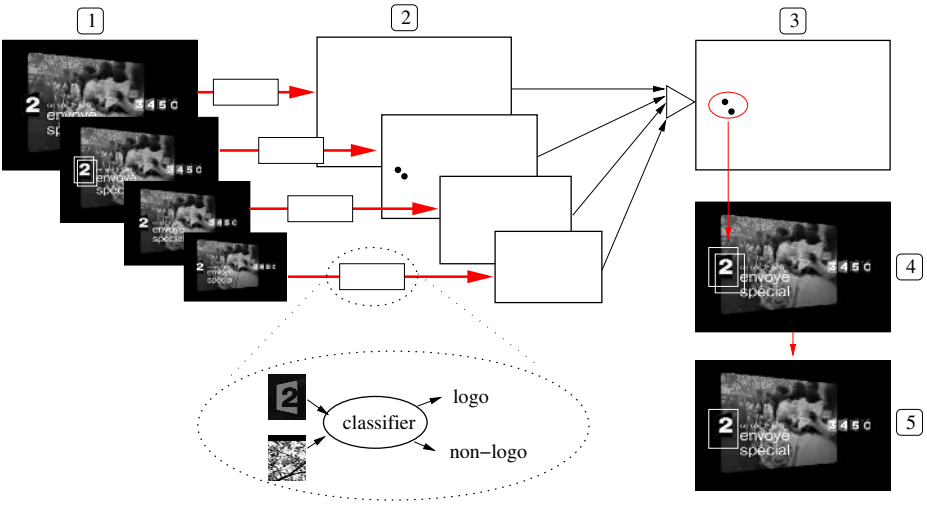


Fig. 3. Multi-scale logo localization

of the logo candidate. It corresponds to a small pyramid centered at the logo candidate center position covering ten equally distant scales varying from 0.8 to 1.5 times the scale of the logo candidate. For every scale, the presence of a logo is evaluated on a rescaled grid of 16×16 pixels around the corresponding logo candidate center position. We observed that true logos usually give a significant number of high positive responses in consecutive scales, which is not often the case for non logos. In order to discriminate true logos from false alarms, it resulted efficient to take into account both number and values of positive answers. We therefore consider the volume of positive answers (the sum of positive answer values) in the local pyramid in order to take the classification decision. Based on the experiments described in the next section, a logo candidate is classified as logo if its corresponding volume is greater than a given threshold *ThrVol* (step 5). The bottom-right image of Fig.3 shows the position and size of the detected logo after local search.

5 Experimental Results

We tested the trained logo detection system on two sets containing images extracted from TV programs. The first set consists of 800 images each containing one FR2 logo. The other consists of 257 images not containing any FR2 logo. Fig. 4 shows a ROC curve for the first set. This curve presents the detection rate as a function of the number of false alarms while varying the volume threshold *ThrVol*. One can clearly notice that, for a low number of false alarms, the detector attains a high detection rate. For example, if we allow 10 false alarms (for

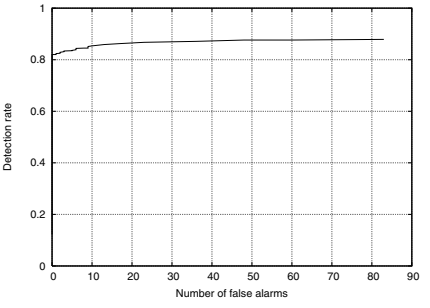


Fig. 4. Detection rate on the first test set versus the number of false alarms for varying volume threshold *ThrVol*

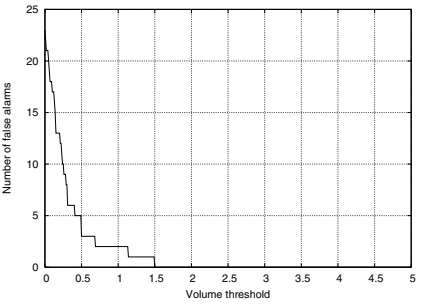


Fig. 5. Number of false alarms on the second test set versus the volume threshold *ThrVol*

Fig. 6. Some results with false alarms

the 800 images) the system detects about 85% of the logos, which seems to be the maximal detection rate that can be reached on these test images, with the proposed architecture. An important point is that we obtain a good detection rate of 82% with no false alarm, for values of *ThrVol* above 1.5. Note that for 13 test images ($\approx 1.6\%$), we judged the logo invisible to the human eye, but we still counted these examples as undetected. Fig. 6 shows some images with false alarms for a very low *ThrVol*.

In the second experiment, we applied the FR2 logo detector on the second test set that does not contain any image displaying the logo. The curve in Fig. 5 shows the number of false alarms as a function of the volume threshold *ThrVol*. One can notice that this number of false alarm decreases very quickly as *ThrVol* increases, and that no false alarm are produced for *ThrVol* above 1.5. For illustration purposes, Fig. 7 shows some images of the first test set with detected transparent logos. There are examples containing logos of very low contrast due to light background. Other examples show logos over a high contrasted non-uniform background which considerably falsifies the logo contours in the image region. There are also some examples of FR2 logos of different sizes at different positions in the image.

Fig. 7. Some results of logo detection on "France 2" TV programs

6 Conclusion

Our experiments have shown that a multi-resolution scheme based on convolutional neural networks is very powerful for transparent logo detection. Indeed, this approach does not require any heuristic regarding image preprocessing, low level measures to extract or segmented shape analysis. The detection rate is very high even in cases where the transparent logo is very poorly contrasted because of the video background. Due to its convolutional nature and the use of a single network, our approach is very fast and can be easily embedded in

real time on various platforms. Moreover, recent experimental results tend to show that multiple transparent logos can be handled through the use of a single light convolutional architecture. As an extension of this work, we are currently considering the detection of animated deformable transparent logos.

References

1. R.J.M. den Hollander and A. Hanjalic. Logo recognition in video stills by string matching. In *Proceedings of International Conference on Image Processing (ICIP)*, pages 517–520, 2003.
2. C. Garcia and M. Delakis. Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1408–1423, 2004.
3. Y. LeCun, L. Bottou, , Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
4. H. Pan, B. Li, and M. Ibrahim Sezan. Automatic detection of replay segments in broadcast sports programs by detection of logos in scene transitions. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2002.
5. H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.
6. H. Sanson. Video indexing: Myth and reality. In *Proceedings of International Workshop on Content-Based Multimedia Indexing*, 2005.
7. C.G.M. Snoek and M. Worring. A state-of-the-art review on multimodal video indexing. In *Proceedings of the 8th Annual Conference of the Advanced School for Computing and Imaging*, 2002.
8. K.K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51, 1998.
9. R. C. Veltkamp and M. Tanase. Content-based image retrieval systems: A survey. *IEEE Image Processing*, 1(1):100–148, 2001.
10. M. Visani, C. Garcia, and J.M. Jolion. Bilinear discriminant analysis for face recognition. In *Proceedings of International Conference on Advances in Pattern Recognition (ICAPR 2005)*, 2005.
11. K. Zyga, R. Price, and B. Williams. A generalized regression neural network for logo recognition. In *Proceedings of International Conference on Knowledge-Based Engineering Systems and Allied Technologies*, 2000.

A Neural Network to Retrieve Images from Text Queries

David Grangier[1,2] and Samy Bengio[1]

[1] IDIAP Research Institute, Martigny, Switzerland
firstname.lastname@idiap.ch
[2] Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Abstract. This work presents a neural network for the retrieval of images from text queries. The proposed network is composed of two main modules: the first one extracts a global picture representation from local block descriptors while the second one aims at solving the retrieval problem from the extracted representation. Both modules are trained jointly to minimize a loss related to the retrieval performance. This approach is shown to be advantageous when compared to previous models relying on unsupervised feature extraction: average precision over *Corel* queries reaches 26.2% for our model, which should be compared to 21.6% for PAMIR, the best alternative.

1 Introduction

A system for the retrieval of images from text queries is essential to take full benefit from large picture databases such as stock photography catalogs, newspaper archives or website images. A widely used solution to this problem is to manually annotate each image in the targeted database and then use a text search engine over the annotations. However, this approach is time-consuming and hence costly, moreover it often results in incomplete and biased annotations which degrades retrieval performance. Therefore, several approaches to avoid this manual step have been proposed in the literature [1,2,3,4]. These approaches are either generative auto-captioning models or discriminative retrieval models. Generative auto-captioning models aims at inferring textual captions from pictures that can then be searched with a text retrieval system [1,3,4], while discriminative retrieval models do not introduce an intermediate captioning step and are directly trained to optimize a criterion related to retrieval performance [2].

In this work, a discriminative approach is proposed. This approach relies on a neural network composed of two main modules: the first module extracts global image features from a set of block descriptors, while the second module aims at solving the retrieval task from the extracted features. The training of both modules is performed simultaneously through gradient descent, meaning that image feature extraction and global decision parameters are inferred to optimize a retrieval criterion. This block-based neural network (BBNN) contrasts with previous discriminative models, such as [2], in which the extraction of image representation is chosen prior to training. This difference is shown to yield

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 24–34, 2006.

significant improvement in practice and BBNN is reported to outperform both generative and discriminative alternatives over the benchmark *Corel* dataset [5] (e.g. BBNN reaches 26.2% average precision over evaluation queries which should be compared to 21.6% for PAMIR, the best alternative, see Section 5).

The remainder of this paper is organized as follows: Section 2 briefly describes the related work, Section 3 introduces the proposed approach, Section 4 describes the text and visual features used to represent queries and images. Next, Section 5 presents the experiments performed over the benchmark *Corel* dataset. Finally, Section 6 draws some conclusions.

2 Related Work

As mentioned in introduction, most of the work in image retrieval from text queries focussed on generative models that attempt to solve the image auto-annotation task. These models include Cross-Media Relevance Models (CMRM) [3], Probabilistic Latent Semantic Analysis (PLSA) [4] and Latent Dirichlet Annotation (LDA) [1]. In general, these models introduce different conditional independence assumptions between the observation of text and visual features in an image and the parameters of the model, θ, are selected to maximize the (log) likelihood of some annotated training images, i.e.

$$\theta^* = \text{argmax} \sum_{i=1}^{N} \log P(p_i, c_i|\theta),$$

where $(p_1, \ldots, p_N)$ and $(c_1, \ldots, c_N)$ correspond to the N available training pictures and their captions. The trained models are then applied to associate a caption (or a distribution over text terms) to each of the unannotated test images and a text retrieval system is then applied over these textual outputs.

The training process of these models hence aims at maximizing the training data likelihood, which is not directly related to the targeted retrieval task, i.e. ranking a set of pictures P with respect to a query q such that the picture relevant to q appear above the others. Better performance can be achieved with a more suitable criterion, as recently shown by the discriminative model PAMIR [2]. To the best of our knowledge, the PAMIR approach is the first attempt to train a model to retrieve images from text queries through the optimization of a ranking criterion over a set of training queries. Previous discriminative models have only focussed on categorization ranking problems (e.g. [6,7]), i.e. the task of ranking unseen images with respect to queries known at training time, which is not a true retrieval task in which an unseen query can be submitted.

In this work, we propose to train a neural network with a criterion similar to the one introduced in [2]. This neural network consists of two modules, the first one extracts an image representation from a set of local descriptors and the second one relies on the inferred representation to solve the retrieval problem. The training of both layers is performed jointly through gradient descent (see Section 3). This approach is inspired from convolutional neural

networks (CNN) [8] which have been successfully applied to various classification/detection tasks [8,9]: these models also formulate the identification of a suitable image representation and the classification from this representation as a joint problem. The proposed neural network hence contrasts with the PAMIR model for which the image representation is a-priori chosen. Our experiments over the benchmark *Corel* corpus show that this difference actually yields a significant improvement, e.g. P10 reaches 10.2% for BBNN compared to 8.8% for PAMIR (see Section 5).

3 A Neural Network for Image Retrieval

This section presents the loss function L adopted to discriminatively train an image retrieval model. It then describes the neural network proposed for image retrieval and its training procedure.

3.1 Discriminative Training for Image Retrieval

Before introducing a loss suitable for image retrieval, we should first recall the objective of a retrieval model: given a query q and a set of pictures P, a retrieval model M should ideally rank the pictures of P such that the pictures relevant to q appear above the others, i.e.

$$\forall q, \forall p^+ \in R(q), \forall p^- \notin R(q), rk_M(q, p^+) < rk_M(q, p^-), \tag{1}$$

where $R(q)$ is the set of queries relevant to q and $rk_M(q, p)$ is the rank of picture p in the ranking outputted by M for query q.

In order to achieve such an objective, retrieval models generally introduce a scoring function F that assigns a real value $F(q, p)$ to any query/picture pair (q, p). Given a query q, this function is used to rank the pictures of P by decreasing scores. In this case, the ideal property (1) hence translates to:

$$\forall q, \forall p^+ \in R(q), \forall p^- \notin R(q), F(q, p^+) > F(q, p^-). \tag{2}$$

In order to identify an appropriate function F from a set of training data, the following loss has been introduced [10],

$$\begin{aligned} L(F; D_{train}) &= \sum_{k=1}^{N} l(F; q_k, p_k^+, p_k^-) \\ &= \sum_{k=1}^{N} \max(0, \epsilon_k - F(q_k, p_k^+) + F(q_k, p_k^-)) \end{aligned} \tag{3}$$

where $\forall k, \epsilon_k > 0$ and D_{train} is a set of N triplets $\{(q_k, p_k^+, p_k^-), \forall k = 1, \ldots, N\}$ in which q_k is a text query, p_k^+ is a picture relevant to q and p_k^- is a picture non-relevant to q. This loss L can be referred to as a margin loss since it penalizes the functions F for which there exists training examples (q_k, p_k^+, p_k^-) for which

the score $F(q_k, p_k^+)$ is not greater than $F(q_k, p_k^-)$ by at least a margin of ϵ_k. This loss has already been successfully applied to text retrieval problems [10,11] and to image retrieval problems [2].

Regarding the choice of the margin value ϵ_k, two alternatives have been proposed previously [2]. A first option, *constant-*ϵ, is to set ϵ_k to be the same for all examples, e.g. $\forall k, \epsilon_k = 1$ (the value 1 is chosen arbitrarily here, any positive value would lead to the same optimization problem). Another option, *text-*ϵ, which can be applied only if the training pictures are annotated with textual captions, is to set ϵ_k to be greater than the difference of scores outputted by a text retrieval system F^{text}, i.e.

$$\epsilon_k = \max(\epsilon, F^{text}(q_k, c_k^+) - F^{text}(q_k, c_k^-)), \tag{4}$$

where c_k^+, c_k^- are the captions of the pictures p_k^+, p_k^- and $\epsilon > 0$. This second option has previously shown to be more effective [2] and will hence be used in the following.

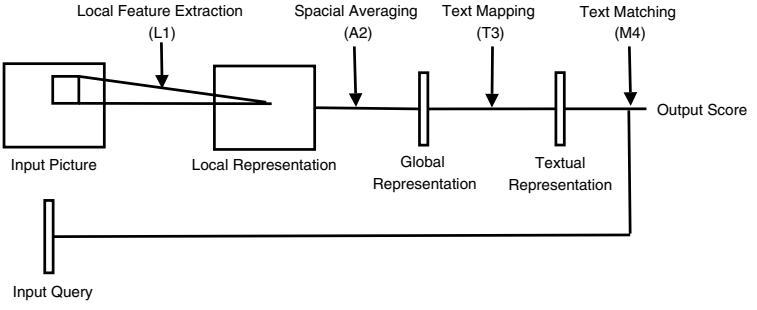


Fig. 1. The 4 successive layers of BBNN: local feature extraction (L1), spacial averaging (A2), text mapping (T3) and text matching (M4)

3.2 Block-Based Neural Network Architecture

As explained above, our goal is to identify a scoring function $q, p \rightarrow F(q, p)$ that minimizes $L(F; D_{train})$. For that purpose, we first introduce the block-based neural network (BBNN), $q, p \rightarrow F_w(q, p)$, and we then explain how the parameters w^* that minimize $w \rightarrow L(F_w; D_{train})$ are identified through stochastic gradient descent.

The proposed neural network is composed of 4 layers (see Figure 1): the local feature extraction layer $L1$, the averaging layer $A2$, the text mapping layer $T3$ and the query matching layer $M4$. The first layer $L1$ extracts local feature descriptors from different positions of the input picture p. The second layer $A2$ computes the average of the local feature vectors extracted by $L1$. The text mapping layer $T3$ then projects the output of $A2$ into the text space. The layer $M4$ finally compares the obtained textual vector with the input query q leading to the output $F(q, p)$. The layers are detailed as follows:

$L1$**: Local Feature Extraction.** This layer extracts the *same* type of features at *different* positions of the input picture p through the following process: first, p is divided into B (possibly overlapping) blocks of the same size, $\{b_1, \ldots, b_B\}$, and each block is assigned a vector representation, i.e. $b_i \in \mathbb{R}^{N_0}$ (see Section 4). The same parametric function is then applied over each block vector,

$$\forall i, f_i = \tanh(W_1 b_i + B_1),$$

where tanh is the component-wise hyperbolic tangent function, $W_1 \in \mathbb{R}^{N_1 \times N_0}$ and $B_1 \in N_1$ are the model parameters. The output dimension N_1 is a hyperparameter of the model.

$A2$**: Spacial Averaging.** This layer summarizes the B output vectors of $L1$ into a single N_1-dimensional vector through averaging:

$$f = \frac{1}{B} \sum_{i=1}^{B} f_i.$$

The succession of $L1$ and $A2$ is inspired from the bag-of-visterms (BOV) representation which has been widely used in computer vision in the recent years, e.g. [1,12]. In this case, a first quantization layer maps each vector b_i to a single discrete value among N_v, which is equivalent to map b_i to a N_v dimensional binary vector in which only one component is 1. In a second step, the input image is represented by a histogram through the averaging of its binary vectors. Here, we replace the quantization step by $L1$, which has two main advantages: first, the vectors f_i are continuous non-sparse vectors which allows to better model correlation between blocks. Second, the parameters of $L1$ are inferred jointly with the next layer parameters to solve the retrieval problem. This contrasts with the BOV approach in which the quantization parameters are generally inferred through generative learning (e.g. k-means clustering).

$T3$: **Text Mapping.** This layer takes as input the representation f of picture p as outputted by $A2$. It then outputs a *bag-of-words* (BOW) vector t, i.e. a vocabulary-sized vector in which each component i represents the weight of term i in picture p (see Section 4 for further description on the BOW representation). This mapping from f to t is performed according to the parametric function:

$$t = W_3 \tanh(W_2 f + B_2) + B_3$$

where $W_2 \in \mathbb{R}^{N_2 \times N_1}$, $B_2 \in \mathbb{R}^{N_2}$, $W_3 \in \mathbb{R}^{V \times N_2}$ and $B_3 \in \mathbb{R}^V$ are the parameters of layer $T3$, V is the vocabulary size and N_3 is a hyperparameter to tune the capacity of $T3$.

$M4$**: Query Matching.** This layer takes two BOW vectors as input: t, the output of $T3$ that represents the input picture p, and q, the input query. It then outputs a real-valued score s. This score is the inner product of t and q,

$$s = \sum_{i=1}^{V} t_i \cdot q_i.$$

This matching layer is inspired from the text retrieval literature in which text documents and text queries are commonly compared according to the inner product of their BOW representation [13].

This neural network approach is inspired from CNN classification models [8] for its first layers ($L1$, $A2$, $T3$) and from text retrieval systems for its last layer (i.e. BOW inner product). Like CNN for classification, our model formulates the problem of image representation and retrieval in a single integrated framework. Moreover like CNN, our parameterization assumes that the final task can be performed through the application of the same local feature extractor at different locations in the image. Our BBNN approach is however not a CNN strictly speaking: the local block descriptors b_i to which the first layer is applied do not simply consist of the gray level of the block pixels like in a CNN. In our case, we extract a N_0 dimensional feature vector summarizing color and texture statistics of the block, as explained in Section 4. This difference is motivated by two main aspects of our task: color information is helpful for image retrieval (see previous works such as [2]) and, moreover, the limited amount of training data prevents us from using a purely data-driven feature extraction technique (see Section 5 which depicts the small number of relevant pictures available for each query).

3.3 Stochastic Gradient Training Procedure

Stochastic gradient descent is the most widely used training technique for neural networks applied to large corpora. Its main advantages are its robustness with respect to local minima, and its fast convergence. We therefore decided to apply this optimization technique to identify the weight vector $w = [W_1; W_2; W_3; B_1; B_2; B_3]$ that minimizes the loss $w \rightarrow L(F_w; D_{train})$, which yields the following algorithm:

Initialize w.
Repeat
  Pick $(q, p^+, p^-) \in D_{train}$ randomly with replacement.
  Compute the gradient $\frac{\partial l}{\partial w}(F_w; q, p^+, p^-)$.
  Update weights $w \leftarrow w - \lambda \frac{\partial l}{\partial w}(F_w; q, p^+, p^-)$.
Until termination criterion.

It should be noted that this version of stochastic gradient training differs from the most used implementation in its sampling process [14]: we choose to sample a training triplet with replacement at each iteration rather than processing the samples sequentially in a shuffled version of the training set. While having no impact on the distribution of the examples seen during training, this difference avoids the costly shuffle for large triplet sets (e.g. there are $\sim 10^8$ triplets for the Corel dataset presented in Section 5).

The other aspects of the training process are more classical: the weight initialization is performed according to the methodology defined in [14] and early stopping is used as the termination criterion [14], i.e. training is stopped when performance over a held-out validation set D_{valid} stops improving. The learning rate λ is selected through cross-validation, as are the other hyperparameters of the model (i.e. N_1, N_2).

4 Text and Visual Features

In this section, we describe the bag-of-words representation used to represent text queries and the edge and color statistics used to represent image blocks.

4.1 Text Features

The text queries are assigned a bag-of-words representation [13]. This representation assigns a vector to each query q, i.e. $q = (q_1, \ldots, q_V)$ where V is the vocabulary size and q_i is the weight of term i in query q. In our case, this weight is assigned according to the well known *normalized tf idf* weighting, i.e.

$$q_i = tf_{q,i} \cdot idf_i,$$

where the term frequency $tf_{q,i}$ is the number of occurrences of i in q and the inverse document frequency idf_i is defined as $idf_i = -log(r_i)$, r_i referring to the fraction of training picture captions containing term i. It should be noted that this definition of *idf* hypothesizes that each training picture is labeled with a caption. This is the case for the *Corel* data used in our experiments (see Section 5). However, were such captions to be unavailable, it would still be possible to compute *idf* over another textual corpus, such as an encyclopedia.

4.2 Image Block Features

The image block descriptors b_i, on which the first layer of our model relies (see Section 3), summarizes edges and color statistics in the following manner.

Color information is represented through a N_C-bin histogram. This histogram relies on a codebook inferred from k-means clustering of the RGB pixels of the training pictures.

Edge information is represented through uniform Local Binary Pattern (uLBP) histograms. These histograms summarize texture information through the binary comparison of pixel intensities between each pixel and its eight neighbors. These features have shown to be effective over various computer vision tasks, including retrieval [15].

Color and edge histograms are then concatenated into a single block vector. Furthermore, a log-scale is adopted in the histograms, i.e. each pixels count c is replaced by $log(1 + c)$, since such non-linear scalings have already shown to be advantageous in previous work [16,13].

5 Experiments and Results

This section first describes the experimental setup and then discusses the results.

5.1 Experimental Setup

The experiments presented in this section have been performed over the *Corel* dataset according to the setup defined in [5]. This setup has been widely used

Table 1. Query Set Statistics

	Q_{train}	Q_{valid}	Q_{test}
Number of queries	7,221	1,962	2,241
Avg. # of rel. pic. per q.	5.33	2.44	2.37
Vocabulary size		179	
Avg. # of words per query	2.78	2.51	2.51

in the image retrieval community [3,2,4] and has become a kind of benchmark protocol for image retrieval. The data used consist of $4,500$ development pictures and 500 test pictures. The size of each picture is either 384×256 or 256×384. We further split the development set into a $4,000$-picture training set and a 500-picture validation set. This hence leads to three picture sets, P_{train}, P_{valid} and P_{test}. Each picture is further labeled with a caption relying on a 179-word vocabulary. These captions have been used for two purposes: for the definition of relevance assessments (i.e. we considered a picture to be relevant to a query q if its caption contained all query terms as explained in [2]) and for $text - \epsilon$ training (in this case, we used inner product of BOW vector as F^{text} function, see equation (4)).

The queries, Q_{train}, Q_{valid} and Q_{test}, used for training, validation and evaluation correspond to all subsets of the 179 vocabulary words for which there is at least one relevant picture within the training, validation or test pictures respectively. Table 1 summarizes query set statistics. The three query/picture datasets (Q_{train}, P_{train}), (Q_{valid}, P_{valid}) and (Q_{test}, P_{test}) have been respectively used to train the model (i.e. select the parameters that minimize the loss L), to select the model hyperparameters (i.e. the learning rate λ and the number of hidden units N_1, N_2) and to perform evaluation. For this evaluation, BBNN performance is measured with precision at top 10 (P10) and average precision (AvgP), the standard measures for information retrieval benchmarks [13]. These measures are complementary and evaluate different retrieval scenarios: P10 focuses on the first positions of the ranking, as the user of a web search engine would do, while AvgP focuses on the whole ranking, as an illustrator requiring all pictures about a specific theme would do. For any query, P10 measures the precision within top 10 positions (i.e. the percentage of relevant pictures within the 10 top-ranked pictures), while AvgP corresponds to the average of precision measured at each position where a relevant picture appears. Both measures have been averaged over the whole query set. BBNN has then been compared with the alternative models CMRM, PLSA and PAMIR which have been evaluated according to the same setup, as explained in [2].

Regarding picture preprocessing, 64×64 square blocks have been extracted every 32 pixels horizontally and vertically, leading to 77 blocks per picture. The size has been chosen as a trade-off between obtaining rich block statistics (i.e. having large blocks with many pixels) and extracting local patterns from the image (i.e. having many small blocks). The overlap of 32 pixels has been selected such that all pixels belong to the same number of blocks, which avoids the predominance of pixels located at the block borders. Concerning the color codebook

Table 2. P10 and mean average precision (%) over test queries

	CMRM	PLSA	PAMIR	BBNN
P10	5.8	7.1	8.8	**10.2**
AvgP	14.7	16.7	21.6	**26.2**

size, we defined $N_C = 50$ which allows a perceptually good picture reconstruction while keeping the block histogram size reasonable. Although it would be more appropriate to select all these parameters through cross-validation, these a-priori choices already led to promising results, as reported in the next section.

5.2 Results

Table 2 reports the results obtained over the test queries. BBNN outperforms all other evaluated techniques for both measures. For AvgP, the relative improvement over CMRM, PLSA and PAMIR is respectively +78%, +57% and +21%. For P10, BBNN reaches 10.2%, which means that, on average, ~ 1 relevant picture appears within the top 10 positions. This number corresponds to good performance considering the low number of relevant pictures per query (2.37 on average, see Table 1). It fact, P10 cannot exceed 20.2% over *Corel* evaluation queries. In order to check whether the improvements observed for P10 and AvgP on the whole query set could be due to a few queries, we further compared BBNN results to those of the other models according to the Wilcoxon signed rank test [17]. The test rejected this hypothesis with 95% confidence for all models and both measures, which is indicated by bold numbers in the table. This means that BBNN consistently outperforms the alternative approaches on the test query set.

The results reported in Table 2 outline the effectiveness of discriminative approaches (PAMIR and BBNN) which both outperform the generative alternative (CMRM and PLSA). This shows the appropriateness of the selected loss function (3) for image retrieval problems. This outcome is in agreement with the text retrieval literature that recently reported good results with models relying on similar criteria [10,16,11].

As mentioned above, a difference in performance is also observed between the two discriminative models: BBNN is reported to outperform PAMIR (26.2% vs 21.6% AvgP). Since both models rely on the optimization of the same loss function, the observed difference is certainly due to the parameterization of the models. On one hand, PAMIR takes as input a bag-of-visterms representation of images, this representation being inferred from local descriptor through unsupervised clustering [2]. On the other hand, BBNN formulates the problem of representing images from local descriptors and the image retrieval task in a single integrated framework (see Section 3). This joint formulation allows the identification of a problem-specific image representation, which seems more effective than the bag-of-visterms representation.

Since several studies report results only for single word queries (e.g. [4,5]), we also trained and evaluated the model over the subset of our train and test

Table 3. P10 and mean average precision (%) over single-word test queries

	CMRM	PLSA	PAMIR	BBNN
P10	17.8	21.3	25.3	**28.5**
AvgP	19.2	24.5	30.7	**35.0**

queries containing only 1 word. The results of this experiments are reported in Table 3. This evaluation further confirms the advantage of BBNN which yields a significant improvement is this case also. It should be noted that the difference observed between Table 2 and Table 3 does not mean that the retrieval models are more adapted to single-word queries: it only reflects the fact that single-word queries correspond to an easier retrieval problem (the average number of relevant documents per query is 2.4 for the whole Q_{test} set and 9.4 for its single-word query subset).

Overall, the results of both retrieval experiments confirm the advantage of supervised feature extraction that has already been observed with CNN over other tasks, such as classification or detection [8,9].

6 Conclusions

We have introduced a discriminative model for the retrieval of images from text queries. This model relies on a neural network architecture inspired from convolutional neural networks [8]. The proposed network, Block-Based Neural Network (BBNN), formulates the identification of global image features from local block descriptors and the retrieval of images from such features as a joint problem. This approach is shown to be effective over the benchmark *Corel* dataset [5]. In particular, BBNN is reported to outperform both generative and discriminative state-of-the-art alternatives. For instance, the mean average precision over *Corel* test queries has been improved by 21% relative compared to the second best model PAMIR [2] (26.2% vs 21.6%). These results are promising and need to be confirmed over other datasets. It could also be interesting to extend the BBNN approach such that it could be applied to other retrieval problems, such as video retrieval.

Acknowledgments. This work has been performed with the support of the Swiss NSF through the NCCR–IM2 project. It was also supported by the PASCAL European Network of Excellence, funded by the Swiss OFES.

References

1. Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D.M., Jordan, M.I.: Matching words and pictures. J. of Machine Learning Research **3** (2003)
2. Grangier, D., Bengio, S.: A discriminative approach for the retrieval of images from text queries. Technical report, IDIAP Research Institute (2006)

3. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: ACM Special Interest Group on Information Retrieval. (2003)
4. Monay, F., Gatica-Perez, D.: PLSA-based image auto-annotation: constraining the latent space. In: ACM Multimedia. (2004)
5. Duygulu, P., Barnard, K., de Freitas, N., Forsyth, D.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: European Conf. on Computer Vision. (2002)
6. Tieu, K., Viola, P.: Boosting image retrieval. Intl. J. of Computer Vision **56** (2004)
7. Wu, H., LuE, H., Ma, S.: A practical SVM-based algorithm for ordinal regression in image retrieval. In: ACM Multimedia. (2003)
8. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Handwritten digit recognition with a back-propagation network. In: Conf. on Advances in Neural Information Processing Systems. (1989)
9. Garcia, C., Delakis, M.: Convolutional face finder: A neural architecture for fast and robust face detection. T. on Pattern Analysis and Machine Intelligence **26** (2004)
10. Joachims, T.: Optimizing search engines using clickthrough data. In: Intl. Conf. on Knowledge Discovery and Data Mining. (2002)
11. Grangier, D., Bengio, S.: Exploiting hyperlinks to learn a retrieval model. In: NIPS Workshop on Learning to Rank. (2005)
12. Quelhas, P., Monay, F., Odobez, J.M., Gatica-Perez, D., Tuytelaars, T., Gool, L.J.V.: Modeling scenes with local descriptors and latent aspects. In: Intl. Conf. on Computer Vision. (2005)
13. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison Wesley, Harlow, England (1999)
14. LeCun, Y., Bottou, L., Orr, G.B., Mueller, K.R.: Efficient backprop. In Orr, G.B., Mueller, K.R., eds.: Neural Networks: Trick of the Trade. Springer (1998)
15. Takala, V., Ahonen, T., Pietikainen, M.: Block-based methods for image retrieval using local binary patterns. In: Scandinavian Conf. on Image Analysis. (2005)
16. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: Intl. Conf. on Machine Learning. (2005)
17. Rice, J.: Rice, Mathematical Statistics and Data Analysis. Duxbury Press, Belmont, California (1995)

Techniques for Still Image Scene Classification and Object Detection*

Ville Viitaniemi and Jorma Laaksonen

Laboratory of Computer and Information Science, Helsinki University of Technology,
P.O. Box 5400, FIN-02015 TKK, Finland
{ville.viitaniemi, jorma.laaksonen}@tkk.fi

Abstract. In this paper we consider the interaction between different semantic levels in still image scene classification and object detection problems. We present a method where a neural method is used to produce a tentative higher-level semantic scene representation from low-level statistical visual features in a bottom-up fashion. This emergent representation is then used to refine the lower-level object detection results. We evaluate the proposed method with data from Pascal VOC Challenge 2006 image classification and object detection competition. The proposed techniques for exploiting global classification results are found to significantly improve the accuracy of local object detection.

1 Introduction

In today's world large amounts of digital video and image material are constantly produced. Furthermore, the rate seems to be constantly increasing. Automatic methods are thus called for to analyse and index these overwhelmingly large data masses. Especially useful would be methods that could automatically analyse the semantic contents of images and videos as it is just the content that determines the relevance in most of the potential uses.

The major challenge in semantic image content analysis is the gap between high-level semantic analysis that would be most beneficial for the potential applications and the low-level visual characterisations produced by bottom-up image analysis systems. The correspondence between entities on different semantic levels can be studied from the viewpoint of emergence [6]. Emergence is a process where a new, higher-level phenomenon results from co-operation of a large number of elementary processes. Neural networks, fed with large amounts of semantically low-level visual data, have turned out to produce useful emergent representations of higher-level semantic concepts, e.g [4]. The bottom-up image content analysis approach with neural methods is thus able to overcome the semantic gap to some degree. Admittedly, the depth and accuracy of the achieved analysis leaves lots to be desired.

* Supported by the Academy of Finland in the projects *Neural methods in information retrieval based on automatic content analysis and relevance feedback* and *Finnish Centre of Excellence in Adaptive Informatics Research.*

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 35–44, 2006.

Image analysis can be seen as interaction between top-down and bottom-up processes. The top-down process generates hypotheses of the visual scene contents and tries to verify them. On the other hand, the bottom-up process starts from low-level visual features and tries to interpret and combine them to form higher-level representations. The hypotheses made by the top-down component guide the bottom-up component's interpretation of the low-level features. On the other hand the hypotheses can be made more accurate and appropriate with help of better low and intermediate level representations.

In this paper we consider forming tentative representations of higher-level semantic concepts by a bottom-up neural classifier and then refining the results of lower-level object detection. We experimentally study the idea in the concrete task of classification of scenes and detection of objects. In the experiments the object detection results will be refined by using the produced scene classifications. The scenes are classified according to whether they contain the target objects. Although the semantic concepts, say, "cow" and "a scene likely to contain a cow" are related, the latter is clearly a richer and thus higher-level concept than the former.

In Section 2 the considered concrete image analysis tasks are described in detail. Section 3 outlines our neural PicSOM image analysis and classification framework. In Section 4 we describe how the framework is applied to the scene classification task. In Section 5 we apply the framework to the object detection task. We look at the detection results both on the first bottom-up iteration and refined by the classification results. In Section 6 we present conclusions from the experiments.

2 Image Classification and Object Detection Tasks

The scene classification and object detection techniques addressed in this paper have been used to participate in the Pascal Visual Object Classes Challenge 2006[1]. In the challenge, machine learning systems are compared by their ability to recognise objects from a number of visual object classes in realistic scenes. The problem is formulated as a supervised learning problem in which a training set of labelled images is provided.

2.1 Image Data

As the test set images provided by the VOC Challenge organisers were not yet released at the time of this writing, we use approximately half of the available images for training and the rest for evaluating the performance of the proposed techniques. This results in an image set consisting of 2618 images which contain 4754 objects of ten selected object classes. The statistics of the image sets are shown in Table 1. From the numbers we see that the images typically contain several objects. They may be either of same or different classes.

[1] http://www.pascal-network.org/challenges/VOC/

Table 1. Statistics of image sets. Columns correspond to different object classes. For each class number of objects (obj) and number of images containing objects of the class are listed.

		bicycle	bus	car	cat	cow	dog	horse	motorbike(mb)	person	sheep	total
training	img	127	93	271	192	102	189	129	118	319	119	1277
set	obj	161	118	427	214	156	211	164	138	577	211	2377
test	img	143	81	282	194	104	176	118	117	347	132	1341
set	obj	162	117	427	215	157	211	162	137	579	210	2377

The annotations of the images contain manually-specified bounding boxes of the objects in the images. Additionally the objects may be tagged to be "truncated" or "difficult". Some object classes are further divided into subclasses by the pose of the objects. At present we ignore this extra information apart from the "difficult" tag. Just as in the VOC challenge, the difficult images (less than 10% of the images in each class) are excluded from the performance evaluation.

2.2 Learning Tasks and Performance Measures

We consider here two different types of tasks on the image sets. In the classification task the goal is to predict the presence/absence of an object in the test images. In the detection task the goal is to predict the bounding boxes of objects in the test set images.

In this paper we use the same quantitative performance measures as in the VOC Challenge. Classification performance is evaluated in terms of the Area Under Curve (AUC) attribute of the Receiver Operating Characteristic (ROC) curves. The performance in detection tasks is assessed by means of the precision/recall (PR) curve. The average precision (AP) in the PR-curve is used as the quantitative measure. Detections are considered as true or false positives based on the relative area of overlap with ground truth bounding boxes. To be regarded as a correct detection, the area of overlap a_{o} between the predicted bounding box B_{p} and ground truth bounding box B_{gt} must exceed 50% by the formula

$$a_{\mathrm{o}} = \frac{\mathrm{area}(B_{\mathrm{p}} \cap B_{\mathrm{gt}})}{\mathrm{area}(B_{\mathrm{p}} \cup B_{\mathrm{gt}})}. \tag{1}$$

Multiple detections of the same object are considered false detections.

3 PicSOM Framework for Object Classification

In the forthcoming sections we use the term target object to denote training and test set images for the scene classification task and automatically extracted image segments for the object detection task. Our proposed method for tackling the VOC Challenge tasks is based on assessing the similarity of the visual properties of the test set target objects to the properties of the training set target objects. Section 3.1 outlines the neural framework used for the similarity assessment. For

more details, see e.g. [5]. Sections 3.2 and 3.3 specify some details of the used automatic image segmentation and visual feature extraction methods.

3.1 Outline

In the PicSOM image analysis framework, target objects are ranked according to their similarity with a given set of positive example objects, simultaneously combined with the dissimilarity with a set of negative example objects. The objects are correlated in terms of a large set of visual features of statistical nature. For this purpose training and test set images are pre-processed in the same manner: the images are first automatically segmented and a large body of statistical features is extracted from both the segments and whole images. Several different segmentations of the same images can be used in parallel. In the current experiments we consider only visual features, but the framework has been used to index also e.g. multimedia messages and videos [4].

After feature extraction, a tree-structured variant of Self-Organising Map [2], a TS-SOM [3], is trained in an unsupervised manner to quantise each of the formed feature spaces. The quantisation forms representations of the feature spaces where points on the TS-SOM surfaces correspond to images and image segments. Due to the topology preserving property of the TS-SOM mapping, the classification in each of the individual feature spaces can be performed by evaluating the distance of representation of an object on the TS-SOM grid to the representations of positive and negative example objects. A single combined similarity measure is formed by summing the contributions of the individual feature spaces. Because of the performed normalisations, the combining algorithm automatically emphasises feature spaces that perform best in discriminating the objects. In the classification task where the target objects are images, the segment-wise similarities are finally combined within an image by summing the contributions of all the segments in it.

For the current experiments the set of features is selected separately for each classification task to maximise classification performance in terms of ROC AUC in the training set. We use a greedy sequential forward search procedure to grow the set of used features until the used performance criterion stops improving.

3.2 Segmentation

Analogously to the principle of allowing the system to automatically select the most beneficial combination of feature TS-SOMs, we address the question of selecting the automatic segmentation method by making a number of alternative segmentations available, and letting the system automatically choose the appropriate ones. According to our earlier experience, the PicSOM algorithm for statistical integration of visual features seems to be quite robust against presence of irrelevant extraneous information. Such unnecessary visual features and segmentations seem to increase the level of noise in the problem but do not seriously compromise the performance as long as the noise level remains moderate. This can be understood by considering the distributions of the target objects in

Table 2. Visual features extracted from image segments

MPEG-7 descriptors	non-standard descriptors
Color Layout	average colour in CIE L*a*b* colour space
Dominant Color	central moments of colour distribution
Region Shape	Fourier descriptors of segment contours
Scalable Color	histogram of Sobel edge directions
	co-occurence matrix of Sobel edge directions
	8-neighbourhood binarised intensity texture

the corresponding feature spaces. In the irrelevant feature spaces the distributions of example objects are uniform and act as noise. On the average, no object in the test set is favoured. In contrast, the distributions of example objects in the relevant feature spaces are strongly peaked and able to overcome moderate levels of noise.

Against this background, the exact selection of the segmentation methods used does not seem critical to the performance. For example, selecting the partitions randomly still leads to non-trivial results. Different visual features have different levels of sensitivity to the segmentation method. For instance, colour and texture features are quite robust in terms of segmentation, whereas properly benefiting from segment shape features requires in some sense successful segmentation. We use two means of generating alternative segmentations to be used in the system. First one is to simply use several different segmentation methods. Another mechanism is to record full segmentation hierarchies of the images and simultaneously consider all levels of the hierarchy in the algorithm. In practice, the results we report here use, due to time limitations, only two alternative segmentations in addition to whole images.

For the current experiments we have used a generic image segmentation method which is simple and somewhat rudimentary. The method employs an area-based region merging algorithm based on homogeneity in terms of colour and texture. In addition to the basic segments, we also record the hierarchical segmentation that results from continuing the region-merging algorithm until only three regions remain.

3.3 Statistical Visual Features

A number of statistical visual features is extracted and made available for the similarity assessment algorithm. The features include MPEG-7 standard descriptors [1] as well as some non-standard descriptors. The features are extracted from image segments as well as from whole images when appropriate. Table 2 lists the used visual features.

4 Scene Classification

The scene classification task straightforwardly employs the PicSOM framework described in Section 3. Slight variations of the method are obtained by selecting

Table 3. Test set ROC AUC resulting from different strategies of choosing the set of positive examples

	bicycle	bus	car	cat	cow	dog	horse	mb	person	sheep
all segments	**0.846**	**0.950**	**0.930**	**0.837**	**0.886**	0.754	**0.792**	0.815	**0.761**	**0.890**
touching	0.828	0.942	**0.930**	0.829	0.881	**0.767**	0.773	**0.829**	0.759	**0.890**
min. overlap	0.820	0.912	0.930	0.817	0.873	0.725	0.781	0.807	0.731	0.880

different automatically obtained segments (Section 3.2) as positive examples in the framework. In our experiments either (i) all segments of positive training images, (ii) only those segments touching the manually specified bounding boxes, or (iii) those segments having minimum overlap of 45% with the bounding boxes were used as positive examples. All segments in the negative images were used as negative examples. For these experiments we used all the features available in the system.

Table 3 displays the resulting test set performances. Different object classes have different optimal strategies of selecting example segments, which is understandable in the light of the images being of different type. Different scene classes vary in the degree in which they depend on the context information outside the object defining the scene class. Even semantically seemingly similar classes, such as "cat" and "dog", may appear in different types of contexts in the training images and thus have different optimal strategies for selecting the example segments. In the light of these examples, it is worthwhile, however, to include all segments of positive images in the positive example set in most cases. There the slightly increased background noise level is more than compensated by the introduced additional context information.

5 Object Detection

Our approach to the object detection problem is based on ranking the segments in the pre-calculated segmentations according to their likelihood to present the target object. In the rest of this section we describe the techniques we have developed for object detection. We take the straightforward application of the PicSOM framework as our baseline (Section 5.1) and then describe two improvements thereupon: incorporation of global scene classification results (Section 5.2) and heuristics for redistributing detector outcomes within images (Section 5.3). Figure 1 shows some correct and incorrect detections on the present image data.

5.1 Baseline Method

In our baseline method we consider all the test image segments given by a certain segmentation method. The segments are ranked using the PicSOM framework of Section 3 according to their similarity of target object segments in the training images. Rows B in Table 4 show the average precision (AP) measure for the test set object detection. For comparison, we display also the performance figures

(a) (b) (c) (d)

Fig. 1. Examples of correct (a) and different types of incorrect (b through d) detections of object class "sheep"

Table 4. Average precision of the algorithm variants in object detection tasks for all the objects classes and different algorithm variants. Rows designated with B correspond to the baseline method, G the incorporation of global classification results and P the nonlinear propagation algorithm. In these experiments all segments of the negative images were used as negative examples. Within an algorithm the different rows correspond to the different strategies of choosing the set of positive examples: all segments, segments overlapping with the ground truth bounding boxes of the objects (touching) and segments having at least 45% overlap with the bounding boxes (min. overlap). The performance of VOC example detector is shown on the uppermost row for reference.

	bicycle	bus	car	cat	cow	dog	horse	mb	person	sheep
VOC example	0.013	0.005	0.013	0.016	0.091	0.013	0.001	0.005	0.004	0.010
B, all segments	0.050	**0.121**	**0.212**	0.037	0.110	0.101	0.099	0.137	0.019	0.035
B, touching	0.060	0.085	0.188	0.045	0.069	**0.106**	**0.102**	0.141	0.017	0.028
B, min. overlap	**0.122**	0.093	0.210	**0.094**	**0.186**	0.051	0.074	**0.155**	**0.054**	**0.176**
G, touching	**0.215**	0.142	0.194	0.100	0.153	0.075	0.110	**0.141**	0.014	0.110
G, min. overlap	0.199	**0.148**	0.206	**0.179**	**0.228**	0.074	0.052	0.090	**0.039**	**0.232**
G, best	0.178	0.139	**0.223**	0.130	0.226	**0.131**	**0.131**	0.058	0.025	0.223
P, touching	**0.244**	**0.172**	0.022	0.162	0.151	**0.124**	0.058	0.129	0.007	0.083
P, min. overlap	0.237	0.163	0.210	0.162	**0.240**	0.122	0.061	**0.132**	0.012	**0.242**
P, best	0.232	0.164	**0.240**	**0.173**	0.226	0.130	**0.076**	0.069	**0.022**	0.220

of simplistic VOC example implementation The results show that for almost all of the ten object classes we clearly achieve non-trivial performance level. For the class "person" and to some extent for the classes "cat " and "dog", however, the detection accuracy is low. This is mainly due to the coarseness of the selected fixed segmentation. To capture these objects, a more fine-grained segmentations of the images would be required. Variants of the basic method obtained by choosing slightly different strategies for selecting the positive and negative example segments for the classification algorithm are also shown in the table. The reasons for the performance differences are the same as discussed in Section 4.

In the current experiments we do not adequately address the issue of suppressing multiple detections of a single object. This issue is relevant for our approach as the same image areas are contained in several levels of hierarchical segmentation. As a heuristic cure, we have exponentially discounted subsequent

detections when an object has already been detected in the same image. This procedure consistently improves the detection accuracy.

5.2 Incorporation of Global Classification Results

In this section we employ the natural connection between the scene classification and object detection tasks. If the scene does not contain an object of a certain class as whole, neither can any of its parts correspond to a true positive detection of that class. This suggests factorisation of the detector outcome into a product where the conditional object detector confidence is modulated by the overall probability of the scene containing objects of the class. As the overall scene classification is more reliable than the object detection both in case of our system and in general, this may provide a practical avenue in constructing an object detector.

More formally, let r be an image segment, I_s and I_i binary indicator variables for the segment r and the corresponding image, respectively, belonging to certain object class. The trained classifier outputs two confidence values: c_i for the segment r being a true detection, and c_s for r being contained in an image belonging to the class. Now we write for the probability of r being a true detection

$$p(I_\mathrm{s} = 1|c_\mathrm{i}, c_\mathrm{s}) = p(I_\mathrm{s} = 1|I_\mathrm{i} = 1, c_\mathrm{i}, c_\mathrm{s})p(I_\mathrm{i} = 1|c_\mathrm{i}, c_\mathrm{s}). \quad (2)$$

With rather plausible independence assumptions this can be approximated as

$$p(I_\mathrm{s} = 1|c_\mathrm{i}, c_\mathrm{s}) \approx p(I_\mathrm{s} = 1|I_\mathrm{i} = 1, c_\mathrm{s})p(I_\mathrm{i} = 1|c_\mathrm{i}). \quad (3)$$

The two mappings from classifier scores to probabilities in this product are estimated by applying the PicSOM framework to the training data. The first classifier is a discriminative classifier trained with only segments in the images belonging to the object class. The true positive segments are used as positive examples and other segments as negative examples. The second classifier is directly the same as the one used for the classification task. For the current experiments, the mappings from scores to probabilities are estimated by fitting logistic sigmoids to the training data. Rows G in Table 4 show the object detection performance of the described method.

5.3 Propagation of Detector Outcomes along Segmentation Hierarchy

To augment the statistical detection of object segments with geometric considerations, we implement a mechanism for propagating relevance scores along the segmentation hierarchy within a single image. In the PicSOM algorithm the propagation takes place after the relevance of the individual segments has been evaluated by the SOMs. The classification framework of Section 3 only statistically estimates whether an individual segment is likely to correspond to a correct detection. In particular, this assessment neglects the dependency of the segments in the same image, especially the relations in the segmentation hierarchy.

In contrast, the propagation algorithm simultaneously considers just the correctness of several of the segments and their combinations that appear in the hierarchical segmentation of an image.

Rows P in Table 4 show the object detection performance resulting from the implementation of the above mentioned considerations in form of a simple non-linear score propagation algorithm along the automatically discovered segmentation hierarchy tree. In the algorithm a set of heuristically chosen rules is used to compare the relative relevance of child and parent nodes and in some cases propagate the relevance from children to parent. From the results we see that the inclusion of the propagation step often leads to performance improvements which in some cases are significant. In some other cases the step fails. From this we conclude that there is potentially a large performance gain available in considering the segments of an image simultaneously and taking segmentation hierarchy into account, even though our simple algorithm is not always able to capitalise on it.

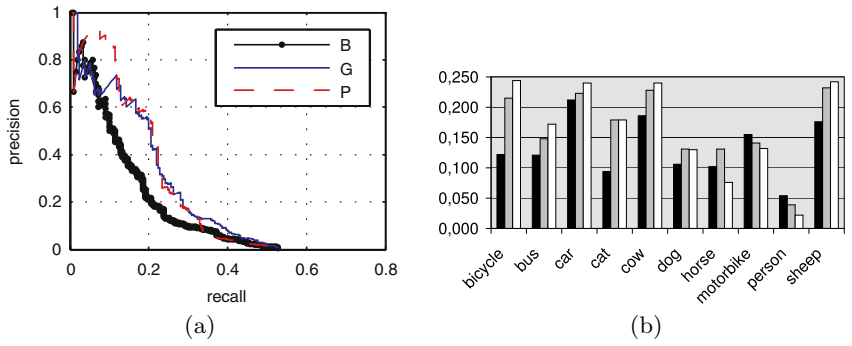
6 Conclusions and Discussion

On the practical side, Figure 2 summarises the object recognition performance achieved by using the techniques described in the previous sections. The techniques were found to be useful and they were implemented and fully evaluated in our entry to the VOC Challenge 2006 image analysis competition. Parts of the contest are still ongoing, but in the preliminary results the PicSOM performance in object classification was slightly over the average among the over 20 participants. Object detection attracted less participants. For some object classes, PicSOM was the best system but not overall.

In this paper we have considered the processing of visual entities on different semantic levels. At least three levels can be distinguished. On the lowermost level there are the statistical low-level visual features describing the images and all their imaginable parts. An intermediate-level semi-semantic representation is formed by considering a small subset of all possible image segmentations. The uppermost semantic level consists of interpretations of whole images in terms of the semantic scene classes of the type "a scene likely to contain a certain object". Of the present concrete image analysis tasks scene classification is concerned with entities on the uppermost semantic level, whereas the object detection operates on the intermediate level.

In the experiments we have seen that the intermediate-level object detection clearly benefits from interaction with the higher-level scene classification, compared with just the bottom-up approach to the detection task. The presented technique can be seen as the first round of iterative alteration of bottom-up and top-down processes. Along the chosen path, the next step we are going to take is to feed the refined intermediate level representation back to the higher-level analysis and to study the usefulness of continuing the iteration even longer.

The distance between the semantic levels which the image analysis aims to link varies, as does the total height and size of the hierarchies. In this comparison



(a) (b)

Fig. 2. Summary of the best object detection results obtained using the three algorithm variants. Subfigure (a) shows the best PR-curves for one object class, "sheep". Curve B (AP = 0.176) corresponds to the baseline algorithm, G (AP = 0.232) the incorporation of global classification results and P (AP = 0.242) the nonlinear propagation algorithm. Subfigure (b) shows average precision measure for all ten classes. Here The leftmost bars (black) represent the baseline algorithm, bars in the middle (grey) the algorithm incorporating global classification results and the rightmost bars (white) the nonlinear score propagation.

the present hierarchy is relatively small and flat, and the levels are quite close together. However, conceptually there is no obvious reason why similar approaches could not be used in case of more challenging semantic hierarchies.

References

1. ISO/IEC. Information technology - Multimedia content description interface - Part 3: Visual, 2002. 15938-3:2002(E).
2. Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer-Verlag, third edition, 2001.
3. Pasi Koikkalainen. Progress with the tree-structured self-organizing map. In *11th European Conference on Artificial Intelligence*. European Committee for Artificial Intelligence (ECCAI), August 1994.
4. Markus Koskela, Jorma Laaksonen, Mats Sjöberg, and Hannes Muurinen. PicSOM experiments in TRECVID 2005. In *Proceedings of the TRECVID 2005 Workshop*, pages 262–270, Gaithersburg, MD, USA, November 2005.
5. Jorma Laaksonen, Markus Koskela, and Erkki Oja. PicSOM—Self-organizing image retrieval with MPEG-7 content descriptions. *IEEE Transactions on Neural Networks, Special Issue on Intelligent Multimedia Processing*, 13(4):841–853, July 2002.
6. A. Ultsch. Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 33–45. Elsevier, 1999.

Adaptation of Weighted Fuzzy Programs

Alexandros Chortaras*, Giorgos Stamou, and Andreas Stafylopatis

School of Electrical and Computer Engineering
National Technical University of Athens
Zografou 157 80, Athens, Greece
{achort, gstam}@softlab.ntua.gr, andreas@cs.ntua.gr

Abstract. Fuzzy logic programs are a useful framework for handling uncertainty in logic programming; nevertheless, there is the need for modelling adaptation of fuzzy logic programs. In this paper, we first overview weighted fuzzy programs, which bring fuzzy logic programs and connectionist models closer together by associating significance weights with the atoms of a logic rule: by exploiting the existence of weights, it is possible to construct a neural network model that reflects the structure of a weighted fuzzy program. Based on this model, we then introduce the weighted fuzzy program adaptation problem and propose an algorithm for adapting the weights of the rules of the program to fit a given dataset.

1 Introduction

In many applications it is essential to reason with uncertain information. Because logic programming is a widely used reasoning framework, adding fuzziness to logic programs has already been investigated (for a review on fuzzy logic programs see e.g. [LS01]). The base of all the proposed approaches is the definition of fuzzy facts, whose truth value belongs to the interval $[0, 1]$. In [CSSK06] the expressiveness of fuzzy programs was extended with the introduction of weights that allow an atom in a rule body to have a different importance in inferencing the head. E.g. the rule $Happy(x) \leftarrow (0.3; Rich(x)) \wedge (0.8; Healthy(x))$ expresses the fact that healthiness is more important for happiness; richness is not (usually) so important, although it may add a bit to the happiness of a person.

On the other hand, neural networks are architectures characterized by adaptation and learning capabilities and fault tolerance. Therefore, the possibility to use connectionist models to perform symbolic reasoning and extract symbolic rules out of them is of considerable interest. Some such models for the representation of fuzzy logic programs have already been developed (e.g. [HHS04], [MCA04]). When the rules include weights, a connectionist representation has a significant advantage: the neural network may be trained to learn the weights that make the rules fit best the data, so that the problem of extracting symbolic rules from the network is reduced to learning the rule weights.

In this paper, after overviewing weighted fuzzy programs and describing how a connectionist model may be constructed for their representation, we investigate

* A. Chortaras is funded by the Alexander S. Onassis Public Benefit Foundation.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 45–54, 2006.

the issue of adapting the weights of a program to fit better a given dataset. For this purpose, we define a mean square error minimization criterion, which gives rise to a heuristic adaptation algorithm that adapts the weights of the neural network. The structure of the paper has as follows: Section 2 overviews the syntax and semantics of definite weighted fuzzy programs, Section 3 describes their connectionist representation, Section 4 defines the weight adaptation problem and presents the adaptation algorithm, and Section 5 concludes the paper.

2 Definite Weighted Fuzzy Programs

2.1 Syntax

Definition 1. *A* fuzzy atom *of predicate p (of arity $\langle p \rangle \geq 0$) is the formula $p(u_1, \ldots, u_n)$ where u_i for $i = 1...n$ are variables or constants. The* truth fuzzy atom t *is a special fuzzy atom of zero arity that represents absolute certainty.*

Definition 2. *A* definite weighted fuzzy program *is a finite set of* weighted fuzzy rules *that are clauses of the form:*

$$w : B \leftarrow \widetilde{\wedge}((w_1; A_1), \ldots, (w_n; A_n))$$

where $A_1, \ldots, A_n$ are fuzzy atoms and B a fuzzy atom excluding t, such that all the variables that appear in B appear also in at least one A_i for $i = 1 \ldots n$. The weight $w \in [0, 1]$ represents the strength of the rule, while the weight $w_i \in [0, 1]$ the significance of atom A_i for the rule.

We write a rule R also as $w : B \leftarrow (w_1; A_1), \ldots, (w_n; A_n)$ and use the notation $s(R) \equiv w$, $\boldsymbol{w}(R) \equiv (w_1, \ldots, w_n)$ and $\boldsymbol{a}(R) \equiv (A_1, \ldots, A_n)$. A fuzzy atom (rule) that contains only constants is called a *ground fuzzy atom* (*rule*) and is denoted by $\underline{\mathrm{A}}$ ($\underline{\mathrm{R}}$). The ground rules that may be obtained from R by substituting all its variables by constants are the *instances* of R. A rule whose body includes only t is a *fuzzy fact*. The number of fuzzy atoms that make up the body of a rule R plus 1 (for the strength) is the *weight size* q_R of R and $q_{\mathcal{P}} = \sum_{R \in \mathcal{P}} q_R$ is the *weight size* of the weighted fuzzy program $\mathcal{P}$.

2.2 Semantics

Let $\mathcal{P}$ be a definite weighted fuzzy program, $P_{\mathcal{P}}$ the set of all the predicates that appear in $\mathcal{P}$ excluding t, $V_{\mathcal{P}}$ a superset of the set of all the constants that appear in $\mathcal{P}$ (an *extended Herbrand universe*), and $B_{\mathcal{P}}(V_{\mathcal{P}})$ the set of all the ground fuzzy atoms that can be constructed from the predicates in $P_{\mathcal{P}}$ and the constants in $V_{\mathcal{P}}$ (the *extended Herbrand base*).

Given an extended Herbrand base $B_{\mathcal{P}}(V_{\mathcal{P}})$, the *base* B_p of predicate p is the subset of $B_{\mathcal{P}}(V_{\mathcal{P}})$ that consists of all the ground fuzzy atoms of predicate p. We denote a member of B_p by $\underline{B}$ or $p(\boldsymbol{c})$ for some $\boldsymbol{c} \in V_{\mathcal{P}}^{\langle p \rangle}$. The *rule base* R_p of predicate p is the set of all the rules in $\mathcal{P}$ whose head is a fuzzy atom of predicate p. The *inference base* $R_{\underline{B}}$ of a ground fuzzy atom $\underline{B}$ is the set of all

the rule instances of the rules in $\mathcal{P}$ that consist of atoms from $B_{\mathcal{P}}(V_{\mathcal{P}})$ and have $\underline{B}$ in their head. Given an extended Herbrand universe $V_{\mathcal{P}}$, the *explicit extended Herbrand base* $EB_{\mathcal{P}}(V_{\mathcal{P}})$ *of* $\mathcal{P}$ is the set of all the ground fuzzy atoms in $B_{\mathcal{P}}(V_{\mathcal{P}})$ whose predicate appears only in the body of a rule in $\mathcal{P}$.

Definition 3. *An* extended fuzzy Herbrand interpretation I *with domain* $V_{\mathcal{P}}$ *of* $\mathcal{P}$ *is a set of mappings* $p^I : V_{\mathcal{P}}^{\langle p \rangle} \to [0,1]$ *that associate each tuple of* $V_{\mathcal{P}}^{\langle p \rangle}$ *with a certainty value* $\forall p \in P_{\mathcal{P}}$*. The interpretation assigns to* t *always the value 1.*

We denote the value with which I associates $\underline{A}$ by $\underline{A}^I$. The value of the body of a rule instance $\underline{R} \equiv w : \underline{B} \leftarrow (w_1; \underline{A}_1), \ldots, (w_n; \underline{A}_n)$ may be computed by a *weighted fuzzy AND operator*, introduced in [CSSK06]; here, it suffices to say that it is non-decreasing in $\underline{A}_i^I$ for $i = 1 \ldots n$, that any operand with weight 0 may be omitted without affecting the result and that the result is bounded from above by the maximum weight and $\underline{A}_i^I$. We denote the operator by $\widetilde{\wedge}_{[\cdot]}$, so that the value of $\widetilde{\wedge}((w_1; \underline{A}_1^I), \ldots, (w_n; \underline{A}_n^I))$ is $\widetilde{A}^I = \widetilde{\wedge}_{[w_1,\ldots,w_n]}(\underline{A}_1^I, \ldots, \underline{A}_n^I)$. If $\bar{w} = \max_{i=1\ldots n} w_i$ and T a t-norm and S an s-norm, an example is the operator:

$$\widetilde{\wedge}_{[w_1,\ldots,w_n]}(a_1, \ldots, a_n) = \min_{i=1\ldots n} S(\bar{w} - w_i, T(\bar{w}, a_i))$$

For the values of the atoms in the body of $\underline{R}$ we will also write $\boldsymbol{a}^I(\underline{R})$ instead of $(\underline{A}_1^I, \ldots, \underline{A}_n^I)$. The rules are interpreted as fuzzy r-implications (see [KY95]), so that the certainty value of $\underline{R}$ under I is $\omega_T(\widetilde{A}^I, \underline{B}^I) = \sup\{x \in [0,1] \mid T(\widetilde{A}^I, x) \leq \underline{B}^I\}$ for some t-norm T. Hence, $\omega_T(\widetilde{A}^I, \underline{B}^I) = v$ implies that $\underline{B}^I \geq T(\widetilde{A}^I, v)$.

Definition 4. *Given a fuzzy weighted AND operator* $\widetilde{\wedge}_{[\cdot]}$*, a* t*-norm* T *and an* s*-norm* S*, an extended fuzzy Herbrand interpretation* I *with domain* $V_{\mathcal{P}}$ *is an* extended Herbrand model *of* $\mathcal{P}$ *under* $(\widetilde{\wedge}_{[\cdot]}, T, S)$*, if* $\forall \underline{B} \in B_{\mathcal{P}}(V_{\mathcal{P}})$*:*

$$S\left(\left\{T\left(\widetilde{\wedge}_{[\boldsymbol{w}(R)]}\left(\boldsymbol{a}^I(\underline{R})\right), s(R)\right)\right\}_{\underline{R} \in R_{\underline{B}}}\right) \leq \underline{B}^I$$

Given two interpretations I_1, I_2 of $\mathcal{P}$ with domain $V_{\mathcal{P}}$, I_1 is less or equal to I_2 ($I_1 \preceq I_2$) if $\forall p \in P_{\mathcal{P}}$ and $\forall \boldsymbol{c} \in V_{\mathcal{P}}^{\langle p \rangle}$, $p^{I_1}(\boldsymbol{c}) \leq p^{I_2}(\boldsymbol{c})$. The intersection of I_1, I_2 is the interpretation I with $p^I(\boldsymbol{c}) = \min\{p^{I_1}(\boldsymbol{c}), p^{I_2}(\boldsymbol{c})\}$, $\forall p \in P_{\mathcal{P}}$ and $\forall \boldsymbol{c} \in V_{\mathcal{P}}^{\langle p \rangle}$.

Theorem 1. *Given an extended Herbrand universe* $V_{\mathcal{P}}$*,* $\mathcal{P}$ *has a unique extended minimal Herbrand model* $FM_{\mathcal{P}}$ *under* $(\widetilde{\wedge}_{[\cdot]}, T, S)$*, equal to the intersection of all the extended Herbrand models of* $\mathcal{P}$ *with domain* $V_{\mathcal{P}}$ *under* $(\widetilde{\wedge}_{[\cdot]}, T, S)$*.*

The *fuzzy immediate consequence* under $(\widetilde{\wedge}_{[\cdot]}, T, S)$ of an interpretation I of $\mathcal{P}$ is the interpretation $FT_{\mathcal{P}}(I)$ with the same domain as I, such that $\forall \underline{B} \in B_{\mathcal{P}}(V_{\mathcal{P}})$:

$$\underline{B}^{FT_{\mathcal{P}}(I)} = S\left(\left\{T\left(\widetilde{\wedge}_{[\boldsymbol{w}(R)]}\left(\boldsymbol{a}^I(\underline{R})\right), s(R)\right)\right\}_{\underline{R} \in R_{\underline{B}}}\right)$$

$FT_{\mathcal{P}}$ is the *fuzzy immediate consequence operator* under $(\widetilde{\wedge}_{[\cdot]}, T, S)$ and by fixpoint theory it can be proved that it has a least fixpoint ${FT_{\mathcal{P}}}^{\uparrow\omega}$, which can be determined by a countable number of iterative applications of $FT_{\mathcal{P}}$ (starting from the interpretation that maps all ground fuzzy atoms to 0). It can also be proved that $FM_{\mathcal{P}} = {FT_{\mathcal{P}}}^{\uparrow\omega}$, which defines the intended meaning of $\mathcal{P}$.

3 Connectionist Representation

A connectionist model for the representation of a weighted fuzzy program $\mathcal{P}$ is proposed in [CSSK06]. The proposed neural network has the benefit that its structure reflects the structure of $\mathcal{P}$ and that the link weights are the actual rule weights, in a way that the network has a direct, well-defined symbolic interpretation. Here, we provide a brief overview of the structure of the network.

Because $\mathcal{P}$ may be non-propositional, a property of the network is that its links carry *complex named values*, i.e. multisets of pairs $(\boldsymbol{c}, v)$, where $\boldsymbol{c}$ is a vector of constants (the arguments of a fuzzy ground atom) and v a certainty value.

The network consists of an input, a conjunction and a disjunction layer. The conjunction layer consists of *conjunctive multivalued neurons* that correspond to the rules of $\mathcal{P}$. There is one such neuron for each rule in $\mathcal{P}$. Each neuron has so many inputs as are the atoms of the body of the respective rule, and each link is characterized by the weight of the respective atom. The neuron computes the value of the head of the rule, by appropriately grounding the variables and combining the complex named values that appear in its input links.

The disjunction layer consists of *disjunctive multivalued neurons*, one for each predicate in $P_{\mathcal{P}}$. Their role is to combine into one the possibly many certainty values computed for the same ground fuzzy atom by the conjunction layer. The input links of a disjunctive neuron are connected to the outputs of the conjunctive neurons that compute the rules whose head involves the predicate that the disjunctive neuron represents. The weight of each input link is the strength of the respective rule. The network is recursive, and the outputs of the disjunctive neurons are connected through a unit delay node to the input links of the conjunctive neurons that correspond to the rules that involve in their body the predicates that the disjunctive neurons compute.

The output of the network is considered to be the set of the complex named values that appear at the output of the disjunctive neurons, each characterized by the respective predicate name. Thus the output, or *state*, of the network at time point t is the set $\{(p_1, out_{P_1}(t)), \ldots, (p_k, out_{P_k}(t))\}$, where P_i for $i = 1 \ldots k$ is the disjunctive neuron that represents predicate p_i and out_{P_i} its output.

The network performs a computation in a discrete time setting. As described in [CSSK06], by construction the neural network that corresponds to a weighted fuzzy program $\mathcal{P}$ is a connectionist implementation of $FT_{\mathcal{P}}$, so that at time point t_i its state encodes interpretation ${FT_{\mathcal{P}}}^{\uparrow i}$ (given an initial interpretation provided to the network by the input layer nodes). Thus, the network accomplishes the computation of the minimal model of $\mathcal{P}$ at the same number of time points (iterations) that $FT_{\mathcal{P}}$ needs to reach its least fixpoint. Hence, if $FT_{\mathcal{P}}$ converges to its least fixpoint at a finite number of k iterations, then the state $S(t_k)$, after k time points, of the neural network encodes the minimal Herbrand model $M_{\mathcal{P}}$ of $\mathcal{P}$ under $(\widetilde{\wedge}_{[\cdot]}, T, S)$. In particular, $\forall p \in P_{\mathcal{P}}$, $M_{\mathcal{P}}$ is such that:

$$p^{M_{\mathcal{P}}}(\boldsymbol{c}) = \begin{cases} v & \text{if } (p, W) \in S(t_k) \text{ and } (\boldsymbol{c}, v) \in W \\ 0 & \text{otherwise} \end{cases}$$

Example 1. The neural network that corresponds to the following program (the rules are given on the left and the facts on the right) is illustrated in Fig. 1.

$$w_a : a(x,y) \leftarrow (w_1; a(x,y)), (w_2; b(y,z)), (w_3, c(z)) \qquad v_1 : b(a,b) \leftarrow (1;t)$$
$$w_b : b(x,y) \leftarrow (w_4; c(x)), (w_5; d(y)) \qquad v_2 : d(a) \leftarrow (1;t)$$
$$w_c : c(x) \leftarrow (w_6; d(x)) \qquad v_3 : d(b) \leftarrow (1;t)$$
$$w_e : e(y) \leftarrow (w_7; d(y))$$

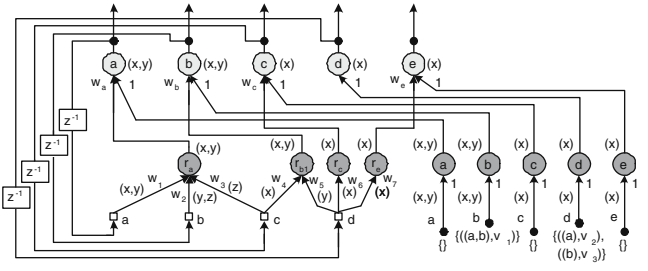


Fig. 1. The neural network of the program of example 1

As it is evident from example 1, the neural network that corresponds to a weighted fuzzy program distinguishes between facts and rules. Its structure reflects the structure of the rules only; the facts are left to be encoded in the input signals. This is useful for the weight adaptation process, described in the next section, because the same rules may be used in combination with different data. Whenever a fact changes the minimal Herbrand model of the program changes as well; thus, if the facts are not encoded in the structure of the network, it is not necessary to construct a new neural network to represent the new program: it suffices only to appropriately change the input signals.

4 Weight Adaptation

In the above discussion, we have assumed that the rules of a weighted fuzzy program have known weights. In the following, we allow also *parametric weighted fuzzy rules* $R(\boldsymbol{w})$, in which the vector of weights $\boldsymbol{w}$ of size q_R is a parameter. Similarly, we allow *parametric weighted fuzzy programs* $\mathcal{P}(\boldsymbol{w})$. In this case, the parameter is the weight vector $\boldsymbol{w}$ of size $q_\mathcal{P}$, obtained from the concatenation (for some ordering) of the weight vectors of the individual rules of $\mathcal{P}$.

Because the network described in Section 3 has a direct symbolic interpretation, it may be used in a machine learning framework and trained in order to learn the rules that characterize a given dataset. If the data consist of database-like tables whose attributes may be seen as fuzzy sets or relations and whose row values as membership values of the individual elements to the sets or relations, the database may be regarded as an interpretation (or model) of an unknown weighted fuzzy program which we would like to learn. Based on some

a priori knowledge on the domain that the database models, we can then set up a parametric weighted fuzzy program $\mathcal{P}(\boldsymbol{w})$ consisting of rule candidates for the description of the data, leaving the rule weights as parameters to be learned (adapted). The available data will be only an arbitrary interpretation of $\mathcal{P}(\boldsymbol{w})$ for some weight values; the aim of the adaptation process should be to adapt the weights, so that the data is a "good" interpretation (or model) of the program.

We define now the weighted fuzzy program adaptation problem as follows: *Given a parametric weighted fuzzy program $\mathcal{P}(\boldsymbol{w})$, an extended fuzzy Herbrand interpretation I for $\mathcal{P}(\boldsymbol{w})$ with domain an extended Herbrand universe $V_{\mathcal{P}}$, and a triple of operators $(\widetilde{\wedge}_{[\cdot]}, T, S)$, find an "optimal" vector of weights $\hat{\boldsymbol{w}} \in [0,1]^{q_{\mathcal{P}}}$ such that I is a model of $\mathcal{P}(\hat{\boldsymbol{w}})$ under $(\widetilde{\wedge}_{[\cdot]}, T, S)$*. We will now investigate how we can learn from I the "optimal" or more in general a "good" vector of weights.

From definition 4, we know that an interpretation I with domain $V_{\mathcal{P}}$ is a model of $\mathcal{P}$ under $(\widetilde{\wedge}_{[\cdot]}, T, S)$, if for all fuzzy ground atoms $\underline{B} \in B_{\mathcal{P}}(V_{\mathcal{P}})$:

$$S\left(\left\{T\left(\widetilde{\wedge}_{[\boldsymbol{w}(R)]}\left(\boldsymbol{a}^I(\underline{R})\right), s(R)\right)\right\}_{\underline{R}\in R_{\underline{B}}}\right) \leq \underline{B}^I \tag{1}$$

This condition must hold $\forall \underline{B} \in B_{\mathcal{P}}(V_{\mathcal{P}})$, thus if we consider a parametric program $\mathcal{P}(\boldsymbol{w})$ and write the corresponding set of inequalities letting the weights be unknown variables, we obtain a system $\Sigma(\boldsymbol{w})$ of $|B_{\mathcal{P}}(V_{\mathcal{P}})|$ inequalities and $q_{\mathcal{P}}$ unknown weights: the vector $\boldsymbol{w}$. Then for some vector $\hat{\boldsymbol{w}} \in [0,1]^{q_{\mathcal{P}}}$, I is a model of $\mathcal{P}(\hat{\boldsymbol{w}})$ if $\hat{\boldsymbol{w}}$ satisfies $\Sigma(\boldsymbol{w})$. However, because the inference bases of any ground atoms of distinct predicates contain instances of distinct rules, and thus involve different weights, $\Sigma(\boldsymbol{w})$ may be split into $|P_{\mathcal{P}}|$ independent systems of $|B_p|$ inequalities, one for each predicate $p \in P_{\mathcal{P}}$. Each one of these systems will have in total $q_p = \sum_{R\in R_p} q_R$ unknown weights, which we denote by $\boldsymbol{w}_p$. In the absence of additional conditions these systems may be solved independently. We denote the partial system corresponding to predicate p by $\Sigma_p(\boldsymbol{w}_p)$, so that $\Sigma(\boldsymbol{w}) \equiv \bigcup_{p\in P_{\mathcal{P}}} \Sigma_p(\boldsymbol{w}_p)$, where now $\boldsymbol{w} = (\boldsymbol{w}_p)_{p\in P_{\mathcal{P}}}$ is the concatenation (for some ordering) of the weight vectors of the $|P_{\mathcal{P}}|$ systems.

It is convenient to define the functions $g_l^p = g_l^p\left(\boldsymbol{w}_p, \left\{\boldsymbol{a}^I(\underline{R})\right\}_{\underline{R}\in R_{\underline{B}}}\right)$, one for each of the left-hand side expressions that make up the $|B_p|$ inequalities of $\Sigma_p(\boldsymbol{w}_p)$ (like inequality 1), where $l = 1 \ldots |B_p|$ and $\underline{B} = p(\boldsymbol{c}_l^p)$, $\boldsymbol{c}_l^p \in V_{\mathcal{P}}^{\langle p\rangle}$. To simplify further notation, we suppress the arguments of g_l^p and write it as $g_i^p(\boldsymbol{w}_p)^I$ where the superscript $\cdot^I$ exists to make clear that its value depends also on $\{\boldsymbol{a}^I(\underline{R})\}_{\underline{R}\in R_{\underline{B}}}$. Using this notation, $\Sigma_p(\boldsymbol{w}_p)$ may eventually be written as:

$$g_l^p(\boldsymbol{w}_p)^I \leq p^I(\boldsymbol{c}_l^p) \text{ for } l = 1 \ldots |B_p|$$

In general, $\Sigma_p(\boldsymbol{w_p})$ (and $\Sigma(\boldsymbol{w})$) will have an infinite number of solutions. It is therefore essential to specify criteria that will allow the selection of some preferred solutions of $\Sigma(\boldsymbol{w})$. A natural choice is to select the maximum solutions of $\Sigma(\boldsymbol{w})$, which have the property that correspond to strong rules, in which the entire body plays the maximal possible role in determining the value of the head. However, it can be proved that even a maximum solution is by itself not "good

enough"; additional criteria are needed in order to obtain rules that express better the relations between the atoms manifested in I. Hence, we recourse to the mean square error minimization criterion and provide the following definitions:

Definition 5. *The* explicit restriction *of the extended fuzzy Herbrand interpretation I with domain $V_{\mathcal{P}}$ on the program $\mathcal{P}$ is the extended fuzzy Herbrand interpretation EI with domain $V_{\mathcal{P}}$ such that:*

$$\underline{B}^{EI} = \begin{cases} \underline{B}^{I} & \text{if } \underline{B} \in EB_{\mathcal{P}}(V_{\mathcal{P}}) \\ 0 & \text{otherwise} \end{cases}$$

Definition 6. *The weight vector $\boldsymbol{w}^*$ is an* optimal fit *under $(\widetilde{\wedge}_{[\cdot]}, T, S)$ of the extended fuzzy Herbrand interpretation I with domain $V_{\mathcal{P}}$ to the parametric weighted fuzzy program $\mathcal{P}(\boldsymbol{w})$, if I is a model of $\mathcal{P}(\boldsymbol{w}^*)$ under $(\widetilde{\wedge}_{[\cdot]}, T, S)$ and:*

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \sum_{\underline{B} \in B_{\mathcal{P}}(V_{\mathcal{P}})} \left(\underline{B}^{I} - \underline{B}^{M(\boldsymbol{w})} \right)^2$$

where $M(\boldsymbol{w}) = FT^{\uparrow\omega}_{\mathcal{P}(\boldsymbol{w})}(EI)$ and EI is the explicit restriction of I on $\mathcal{P}$.

We may now restate the weighted fuzzy program adaptation problem as finding a maximum optimal fit of I under $(\widetilde{\wedge}_{[\cdot]}, T, S)$ to the parametric program $\mathcal{P}(\boldsymbol{w})$, i.e. to finding the maximum optimizers of the following problem $O(\boldsymbol{w})$:

$$\begin{array}{ll} \text{minimize} & \sum_{p \in \mathcal{P}} \sum_{i=1}^{|B_p|} \left(p^I(\boldsymbol{c}_i^p) - g_i^p(\boldsymbol{w}_p)^{M(\boldsymbol{w})} \right)^2 \\ \text{subject to} & g_{i_p}^p(\boldsymbol{w}_p)^I \leq p^I(\boldsymbol{c}_{i_p}^p) \text{ for } p \in \mathcal{P} \text{ and } i_p = 1 \ldots |B_p| \\ & \mathbf{0} \leq \boldsymbol{w}_p \leq \mathbf{1} \end{array}$$

Because of the monotonicity properties of $g_{i_p}^p(\boldsymbol{w}_p)^I$ and the partial ordering of the interpretations, it can be proved that finding an optimizer of $O(\boldsymbol{w})$ is equivalent to independently finding optimizers for the following (partial) problems $O_p(\boldsymbol{w}_p)$, one for each $p \in P_{\mathcal{P}}$:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{|B_p|} \left(p^I(\boldsymbol{c}_i) - g_i^p(\boldsymbol{w}_p)^I \right)^2 \\ \text{subject to} & g_i^p(\boldsymbol{w}_p)^I \leq p^I(\boldsymbol{c}_i) \text{ for } i = 1 \ldots |B_p| \\ & \mathbf{0} \leq \boldsymbol{w}_p \leq \mathbf{1} \end{array}$$

Clearly, the goal function of $O_p(\boldsymbol{w}_p)$ is the sum of the square errors by which the inequalities in $\Sigma_p(\boldsymbol{w}_p)$ are not satisfied as equalities. Thus, it may be regarded as a soft constraint corresponding to the hard constraints $\Sigma_p(\boldsymbol{w}_p)$. In practice, because the data may be noisy, requiring that the hard constraints $\Sigma_p(\boldsymbol{w}_p)$ are satisfied may lead to poor solutions. Thus, we may drop $\Sigma_p(\boldsymbol{w}_p)$ and minimize only the soft constraint, so that we get the problems $O'_p(\boldsymbol{w}_p)$:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{|B_p|} \left(p^I(\boldsymbol{c}_i) - g_i^p(\boldsymbol{w}_p)^I \right)^2 \\ \text{subject to} & \mathbf{0} \leq \boldsymbol{w}_p \leq \mathbf{1} \end{array}$$

Now, I will not be any more a model of the respective program $\mathcal{P}(\boldsymbol{w}^*)$ for an optimizer $\boldsymbol{w}^*$ of the above problems. However, due to the lower sensitivity of $O'_p(\boldsymbol{w}_p)$ to noise, $\mathcal{P}(\boldsymbol{w}^*)$ will eventually capture better the relations manifested in I. For these reason, in the sequel we focus on solving $O'_p(\boldsymbol{w}_p)$.

4.1 Adaptation Algorithm

The problems $O'_p(\boldsymbol{w}_p)$ are non-convex and obtaining a global optimizer for them is computationally hard. Here, we provide a heuristic algorithm for solving them, which relaxes the global optimizer requirement and combines global with local optimization methods in a way that it can be used for the adaptation of the weights of the network of Section 3. The outline of the algorithm has as follows:

1. Use an interval method with a relatively high minimum box width ϵ_D to determine a set D of boxes, subsets of the weight space $[0,1]^{q_p}$, that define the area $\bar{D}$ in which the global minima of the goal function are included.
2. Determine from the elements of D one or more initial points $\boldsymbol{w}^0$.
3. Starting from $\boldsymbol{w}^0$, perform a gradient descent iterative process until a local minimum is reached, ensuring that at each step the new point lies within $\bar{D}$.

Interval methods are deterministic global optimization methods; an elementary discussion may be find in [RV06]. In brief, they split repeatedly the parts of the search space that may contain a global minimum of the goal function f in ever smaller boxes, determine for each such box d a range $r_d = [\underline{r}_d, \overline{r}_d] \supseteq \{f(x) \mid x \in d\}$, and discard any boxes that certainly do not contain a global minimum. The process stops when all the boxes that may contain a global minimum have been split to a width less than ϵ_D. Depending on the properties of f these methods may not behave well; in the worst case it may be needed to split the entire space into boxes of width ϵ_D. In our algorithm, the interval method is used only in order to obtain good starting points for the gradient descent process, thus only a relatively large ϵ_D that leads to an acceptable complexity is required.

From the set D that contains the candidate boxes that may include the global minima of the goal function, in the second step the algorithm chooses some of them, from which some initial points are determined. In our implementation we select the boxes $\{d' \mid \underline{r}_{d'} = \min_{d \in D} \underline{r}_d\}$. The choice is justified because $\underline{r}_d$ is an underestimation of the global minimum of the goal function. Because maximum solutions are preferred, in the case that many boxes achieve the minimum, we keep only the boxes that may contain the maximum solutions. Within the eventually kept boxes, $\boldsymbol{w}_0$ is obtained by evaluating the goal function at k_s random points and keeping the one that produces the lowest value.

For the gradient descent algorithm we note that the goal function of $O'_p(\boldsymbol{w}_p)$ is in general non-smooth, hence subgradient methods (e.g. [Eit04]) are applicable. While performing the subgradient descend, the area defined by $\bar{D}$ acts as a constraint: if we get out of this area we are guaranteed not to reach a global minimum. Thus, the algorithm requires the implementation of a projection

subgradient method in order to enforce this constraint: at step k of the iterative process, the algorithm updates the weight vector according to the rule:

$$\boldsymbol{w}^{k+1} \leftarrow \Pi_{\bar{D}}(\boldsymbol{w}^k - \alpha_k \xi^k) \text{ with } \Pi_{\bar{D}}(z) = \arg\min_{x \in \bar{D}} \|z - x\|$$

where ξ^k is a subgradient [Cla83] of the goal function at $\boldsymbol{w}^k$ (a supergradient if it is concave at $\boldsymbol{w}^k$) and a_k the learning rate. The learning rate initially equals α_0 and decreases according to the rule $a^{k+1} = \frac{a_0}{\sqrt{j+1}}$, where j is the number of times that the learning rule led to an increase of the goal function up to step k.

The above rule is the neural network weight adaptation rule. Because each predicate in $\mathcal{P}$ corresponds to a different and independent problem $O'_p(\boldsymbol{w}_p)$, it follows that each neural network part that models rules for a different predicate has its own adaptation rule: the starting point, the learning rate, the error function to be minimized and the convergence time are different.

4.2 Simulation Results

Table 1 presents the results obtained by running the adaptation algorithm on synthetic data. In particular, we constructed training data for the rule $1.0 : a(x) \leftarrow (0.5; b(x)), (0.8; c(x)), (0.2; d(x)), (0.7; e(x))$. The strength was assumed to be known and the vector of "unknown" weights was $\boldsymbol{w}^* = (0.5, 0.8, 0.2, 0.7)$. For the body we took $b_i, c_i, d_i, e_i \sim U[0, 1]$ for $i = 1 \ldots n$ and 3 datasets were computed for a. In the first two its value was $a_i = \max(0, \min(1, \wedge_{[\boldsymbol{w}^*]}(b_i, c_i, d_i, e_i) + \sigma_N z))$, where $z \sim N(0, 1)$ with $\sigma_N = 0.1$ and $\sigma_N = 0.25$. In the third case a was random: $a_i \sim U[0, 1]$. In all cases $k_s = 100$ and the size of the dataset was $n = 250$. The interval method was applied for ϵ_D equal to 1, 0.5, 0.25 and 0.125.

As we can see from the results presented in Table 1, in the first two cases the algorithm converges to a weight vector $\hat{\boldsymbol{w}}$ close to $\boldsymbol{w}^*$. As the variance of the added noise increases, the approximation becomes worse. It is worth noting that although reducing ϵ_D leads to better results, even for $\epsilon_D = 1$ (when the interval method is essentially skipped) the result is good enough. This may be attributed to the high value of k_s. An important issue is the evaluation of the quality of the results, in the view of the fact that the algorithm always locates a local minimum of the goal function, regardless of the existence of a relation between the data. E.g. in the third dataset such a relation does not exist as all the data were random. As a measure of evaluation, table 1 provides the mean square error μ_1 (the goal of the minimization) and the mean square error μ_2 of those data for which the computed rule $R(\hat{\boldsymbol{w}})$ is not a model:

$$\mu_1 = \frac{1}{n} \sum_{i=1}^{n} \epsilon_i^2 \text{ and } \mu_2 = \frac{1}{|G|} \sum_{i \in G} \epsilon_i^2 \text{ with } \epsilon_i = \hat{a}_i - a_i$$

where $\hat{a}_i = \wedge_{[\hat{\boldsymbol{w}}]}(b_i, c_i, d_i, e_i)$ and $G = \{i \mid \hat{a}_i - a_i > 0\}$, and the variances:

$$\sigma_1 = \frac{1}{n} \sum_{i=1}^{n} (\epsilon_i - \mu_1)^2 \text{ and } \sigma_2 = \frac{1}{|G|} \sum_{i \in G} (\epsilon_i - \mu_1)^2$$

Table 1. Results on synthetic data

σ_N	ϵ_D	$\hat{\boldsymbol{w}}$	μ_1	σ_1	μ_2	σ_2
0.10	0.125	(0.497, 0.859, 0.250, 0.695)	0.0093	0.0093	0.0095	0.0090
0.10	0.250	(0.497, 0.801, 0.250, 0.859)	0.0109	0.0108	0.0088	0.0100
0.10	0.500	(0.497, 0.801, 0.250, 0.859)	0.0109	0.0108	0.0088	0.0100
0.10	1.000	(0.517, 0.869, 0.260, 0.709)	0.0093	0.0093	0.0095	0.0093
0.25	0.125	(0.316, 0.668, 0.000, 0.456)	0.0404	0.0404	0.0434	0.0452
0.25	0.250	(0.316, 0.668, 0.000, 0.456)	0.0404	0.0404	0.0434	0.0452
0.25	0.500	(0.677, 0.617, 0.051, 0.481)	0.0455	0.0444	0.0380	0.0496
0.25	1.000	(0.607, 0.918, 0.293, 0.706)	0.0411	0.0410	0.0440	0.0463
	0.250	(0.057, 0.232, 0.686, 0.000)	0.1064	0.0954	0.0715	0.1292
	0.500	(0.100, 0.276, 0.729, 0.000)	0.1063	0.0954	0.0707	0.1276
	1.000	(0.268, 0.445, 0.896, 0.128)	0.1063	0.0953	0.0695	0.1261

5 Conclusions

Given the increased expressiveness of weighted fuzzy programs in modelling fuzzy datasets, using neural networks in order to learn logic programs is a significant step in the field of integrating symbolic and connectionist models and performing connectionist-based reasoning. Nevertheless, the presented results are only an initial attempt and further research is required. An important issue is that weighted fuzzy programs currently allow the use of definite fuzzy atoms only. The introduction of negation with the non-monotonicity properties that this implies, is a major step that will increase the ability of weighted fuzzy programs and of the corresponding neural networks to be used with real-life applications.

References

[Cla83] F. H. Clarke. *Optimization and Nonsmooth Analysis, Classics in Applied Mathematics*. Wiley and Sons, 1983.

[CSSK06] A. Chortaras, G. Stamou, A. Stafylopatis, and S. Kollias. A connectionist model for weighted fuzzy programs. In *IJCNN '06: Proceedings of the International Joint Conference on Neural Networks*, 2006.

[Eit04] C. Eitzinger. Nonsmooth training of fuzzy neural networks. *Soft Computing*, 8:443–448, 2004.

[HHS04] P. Hitzler, S. Holldobler, and A. Karel Seda. Logic programs and connectionist networks. *Journal of Applied Logic*, 2(3):245–272, 2004.

[KY95] G. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, Upper Saddle River, NJ, 1995.

[LS01] L. Lakshmanan and N. Shiri. A parametric approach to deductive databases with uncertainty. *IEEE Transactions on Knowledge and Data Engineering*, 13(4):554–570, 2001.

[MCA04] J. Medina, E. Mérida Casermeiro, and M. Ojeda Aciego. A neural implementation of multiadjoint programming. *Journal of Applied Logic*, 2(3), 2004.

[RV06] H. Ratschek and R. L. Voller. What can interval analysis do for global optimization? *Journal of Global Optimization*, 1:111–130, 2006.

Classified Ranking of Semantic Content Filtered Output Using Self-organizing Neural Networks

Marios Angelides, Anastasis Sofokleous, and Minaz Parmar

Brunel University, Uxbridge, London, UK
{marios.angelides, anastasis.sofokleous,
minaz.parmar}@brunel.ac.uk

Abstract. Cosmos-7 is an application that can create and filter MPEG-7 semantic content models with regards to objects and events, both spatially and temporally. The results are presented as numerous video segments that are all relevant to the user's consumption criteria. These results are not ranked to the user's ranking of relevancy, which means the user must now laboriously sift through them. Using self organizing networks we rank the segments to the user's preferences by applying the knowledge gained from similar users' experience and use content similarity for new segments to derive a relative ranking.

1 Introduction

Filtering multimedia content is complex because the medium is transient both spatially and temporally. Therefore the content itself has different semantic meaning both spatially and temporally in relation to objects and events, respectively. In order to be able to filter multimedia content we require; 1) A content model that describes the content in terms of spatial and temporal semantic relationships of object and events, 2) A filter that sifts relevant information from the content model based on the user's information requirements.

COSMOS-7 [1] is an MPEG-7 compliant application that reduces the complexity of creating such a content model and filter. It exclusively uses part 5 of the MPEG-7 [2] standard (Multimedia Description Schemes) that semantically describes objects and events and there relationships both temporally and spatially. Unlike other multimedia content modeling systems [3] it does not use low level (syntactic) features, only high level (semantic features) that are meaningful to the user. Using the COSMOS-7 filtering manager a filter is created that can exploit the rich detail captured in the content model by allowing a user to filter out undesirable content.

On examination of the results after filtering it was found there were numerous entries returned that fitted the filter criteria. These results were not ranked by the relevancy to the context of the user's information requirements. This is achievable by understanding the importance a user attaches to high level features. Using collaborative ranking, which is similar to filtering but without exclusion of items, we can predict the user's preference for content by extracting similar users and using

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 55 – 64, 2006.

their preference ranking for content. In this paper we examine the use of two self organizing neural networks to 1) collaboratively filter users into similar clusters and then rank the segments for the user using the previous experience of the peer group 2) Use content based video similarity measures to rank segments outside the peer groups experience in order to find a relative ranking based on experience of similar content by the peer group.

In section 2 an overview is provided of what semantic aspects COSMOS-7 models and what MPEG-7 tools it uses to encapsulate these concepts. Section 3 describes the filtering process of COSMOS-7 to extract a video summary of user preferred content. Section 4 describes the two self organizing neural networks used to personalize the results to the user's taste. The final section is our conclusion and future research.

2 Overview of COSMOS-7

In this section we begin by specifying the attributes that need to be modeled to provide a fully inclusive description of the semantic content which is both concise and flexible. We then describe the COSMOS-7 System, from two particular angles; 1) How COSMOS-7 is modeled using specific MPEG-7 tools to produce such a rich and multi faceted content model. 2) The COSMOS-7 filtering manager that creates and manages filters for extracting content to the user's consumption requirement.

2.1 Modeling Semantic Content Attributes

The semantic content model has to be tightly integrated with the video streams using referencing strategies. In this way, the filtering process may determine the meaning conveyed by the media within the archive so as to compare against the specified filtering criteria. Previously [1] we have identified four key aspects;

Events - Events within the semantic content model represent the context for objects that are present within the video stream at various structural levels. Events are therefore occurrences within the video stream that divide it into shorter semantically continuous content segments involving specific objects, and thus can frequently serve to represent object behaviour.

Temporal Relationships - Temporal relationships between events enable the semantic content model to express the dynamism in content that is apparent at these higher levels, thereby enabling filtering of non-static semantic content which is more in line with "What will or did happen here?" Again, this may occur on both a general and a specific level.

Objects - The expression of objects and object properties within the semantic content model provides for filtering with regards to objects that are readily identifiable within the video content stream. The term 'object' refers to any visible or hidden object depicted within a video frame at any necessary level of detail, from entire objects and groups of objects to the bottom-most component objects. Objects may themselves exist within a structural hierarchy thereby enabling inheritance of properties from higher level objects.

Spatial relationships - Representations of spatial relationships between objects within the semantic content model enable filtering concerning the relative location of objects (rather than the absolute location that comes from a co-ordinate based representation). This enables reference to be made to the relationships between objects within the content and can provide for three-dimensional spatial representations, including those concerning hidden objects, which are difficult to derive from co-ordinate representations. Spatial relationships have a duration due to their occurrence across multiple frames. Because of object motion, spatial relationships between two objects may differ over time within the same segment.

2.2 The COSMOS-7 Content Modeling Tools

The above semantic content aspects can be seen to have generic applicability since virtually all domains require some representation of objects and/or events, including relationships between them, For instance, entertainment-on-demand [4], multimedia

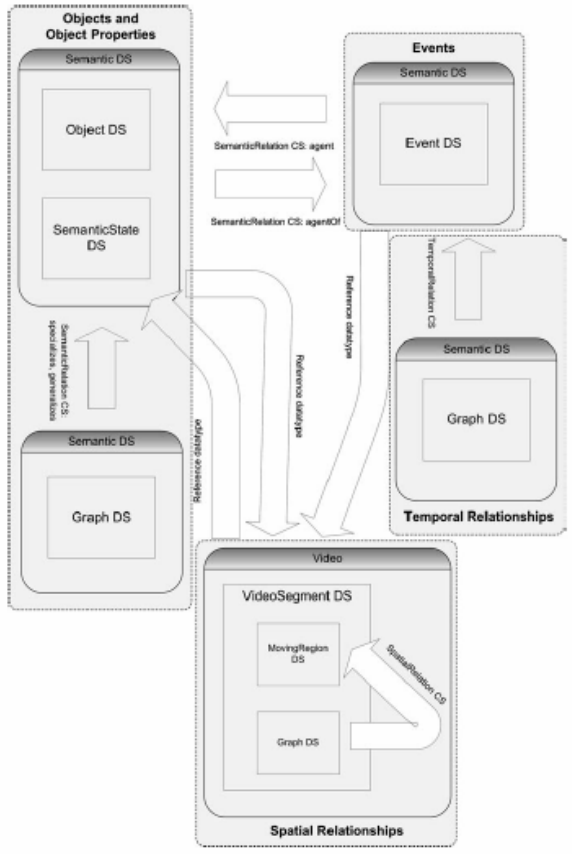


Fig. 1. Key MPEG-7 description tools used in COSMOS-7 [1]

news [5], and organizational content [6]. Hence, when these semantic aspects are accommodated by a content modeling scheme the resultant model can be used to model semantic content for most mainstream domains and user groups and, consequently, facilitate filtering for those domains and user groups.

Figure 1 shows the key MPEG-7 description tools that are used within COSMOS-7 for each semantic aspect and illustrates how they are interrelated.

3 Filtering Events

A user will very often only be interested in certain video content, e.g. when watching a soccer game the user may only be interested in goals and free kicks. Identifying and retrieving subsets of video content in this way requires user preferences for content to be stated, such that content within a digital video resource may then be filtered against those preferences. While new approaches, such as those based on agents [7], are merging, filtering in video streams usually uses content-based filtering methods, which analyze the features of the material so that these may then be filtered against the user's content requirements [8–10]

Content filters are specified using the **FILTER** keyword together with one or more of the following: **EVENT**, **TMPREL**, **OBJ**, **SPLREL**, and **VIDEO**. These may be joined together using the logical operator AND. This specifies what the filter is to return. Consequently, the output of the filtering process may be either semantic content information and/or video segments containing the semantic content information. The criteria are specified using the **WHERE** keyword together with one or more of the following clauses: **EVENT**, **TMPREL**, **OBJ**, and **SPLREL** clauses. These clauses enable event, temporal relationship, object and their properties and

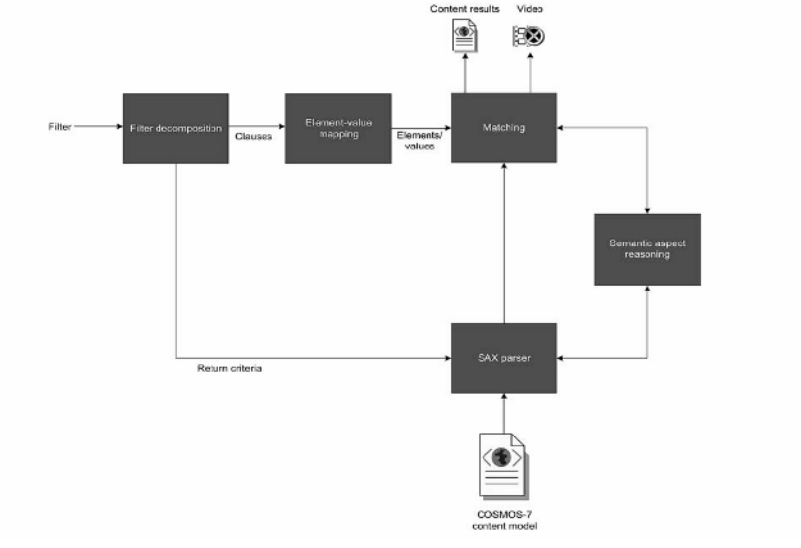


Fig. 2. Filtering process [1]

spatial relationships, respectively, to be specified on the COSMOS-7 representation. Clauses may be joined together using the logical operators AND, NOT and OR and are terminated with semi-colons and grouped with braces.

Figure 2 illustrates the process used by the filter manager. The specific clauses relating to the different aspects are extracted and then mapped to the COSMOS-7 terms that represent these aspects, which in turn becomes a homogenous filter for the COSMOS-7 content model. The filtering process begins with specific reasoning logic that matches the filter criteria for each type of semantic content aspect to corresponding elements found in the content model. In this way, the returned content can be considered to be a subset of the full content model and thus the filtering process reduces the content model to those segments relevant to the filtering criteria.

4 Cosmos-7 Results Ranking Using SONN

Cosmos-7 filter results are presented as a series of video segments allowing a user to selectively interact with them. The filter reduces the content to the user's specific information needs. The number of returned segments can be high as all the information contained in the content model that matches the filter criteria is returned. The segments are presented in chronologic order which is of no real significance to the user. Certainly, a user could browse and play every video segment, but this is not always convenient. Furthermore, the filter does not reflect the user's changing requirements as there is no feedback element to the filter manager. This results in the user having to explicitly redefine their filter for even small requirement changes. What is required is a process of dynamically altering the filter criteria to match the user's changing information preferences as they interact with the system. This is important in information filtering as the information space changes over time and the filter needs to remain current. Results' sorting is an important factor to the user's ability to access the content in a meaningful and effective manner.

We have proposed COSMOS-7 ranking module that is utilized after the filter for sorting the results using prior knowledge. For doing so, it uses two self-organizing neural networks to order the segments in terms of the user's requirements. In related works neural network have been applied to collaborative filtering to cluster users into groups based on similar tastes [11]. Their neural network uses the user demographic and preference attributes for similarity clustering. Neural networks can be used to collaboratively structure user domains, with the clustering based on implicit knowledge of individuals and groups of users [12], this technique uses self-organized neural networks for clustering which combines methods for supervised learning and collaborative filtering. One application [13] uses a predictive self-organizing network that performs synchronized clustering of information and creates preference vectors which are encoded by a user, thus allowing a user to influence the clustering of information vectors.

In figure 2 we present the basic idea of the ranking module based on the collaborative filtering and recommendation principles, which define that a user should be recommended items that are preferable by other people with similar tastes and preferences. Thus, by investigating the items previously rated by other similar users, a

utility value can be given to these items. Although there are many approaches, using various mathematical formulas, we have specified that the rating value of an item I_k and user U_j is computed as an aggregate of the ratings of some other users U_n belonging to the same user group G for the same item I_k. Group G is the set of users as derived by the first neural network, which clusters the users based on their similarities. Naturally, self organizing networks are able to determine as many optimal classes as their internal neurons. We use neural networks for the user clustering since many studies have showed that self organizing neural Networks provide different result compared to statistical clustering. In [14] the authors demonstrate that the self organizing neural networks were able to recognize clusters in a dataset where other statistical algorithms failed to produce meaningful clusters. Furthermore, neural networks clustering results indicate clustering stability [15], which is an important factor for high quality clusters. Our system uses as aggregate function the average of ratings (see eq. 1).

$$R_{collaborative}(I_k, U_i) = \frac{\sum_{U_n \in G(U_i)} r(I_k, U_n)}{\sum_{U_n \in G(U_i)} 1} \tag{1}$$

A common problem that appears in a collaborative system is the need to calculate the ratings of a small number of examples (sparsity problem). Usually, the number of ratings that need to be calculated is bigger compared to the number of ratings already obtained. Some techniques exist for avoiding the problem of sparsity, such as the inclusion of demographics attributes in the group clustering. Therefore the base vector consists of the user attributes B={A_1, A_2, A_3,… A_n}, where A_i is an element of demographics or user preferences. Each user represents a neural network input vector of base vector B. Based on the number of neurons, the neural network will learn to categorize the input vectors it sees.

While the above framework is ideal for formulating the ranking of Cosmos-7 filter results, it is also being borne with some of collaborative filtering drawbacks. New items, not yet rated by anyone, cannot be evaluated. Therefore until the new item is rated by a substantial number of users, the system would not be able to use its rating value appropriately. Cosmos-7 content ranking algorithm is responsible for evaluating new items; the basic idea is that a user will be recommended items similar to the ones the user preferred in the past. The algorithm uses the same principles as the content recommendation and filtering principles. A self organizing neural network clusters the video segments into a number of groups based on their similarities.

The content ranking algorithm is utilized when the collaborative ranking cannot be used though. The item I_k and user U_j is computed as an aggregate of the ratings of some other items I_n belonging to the same item group G for the same user U_i. Equation 2 uses the average ratings of items for a user. The second neural network clusters the items into classes according to their similarities. The base vector includes attributes from COSMOS-7 model such as objects, events etc. Therefore each item

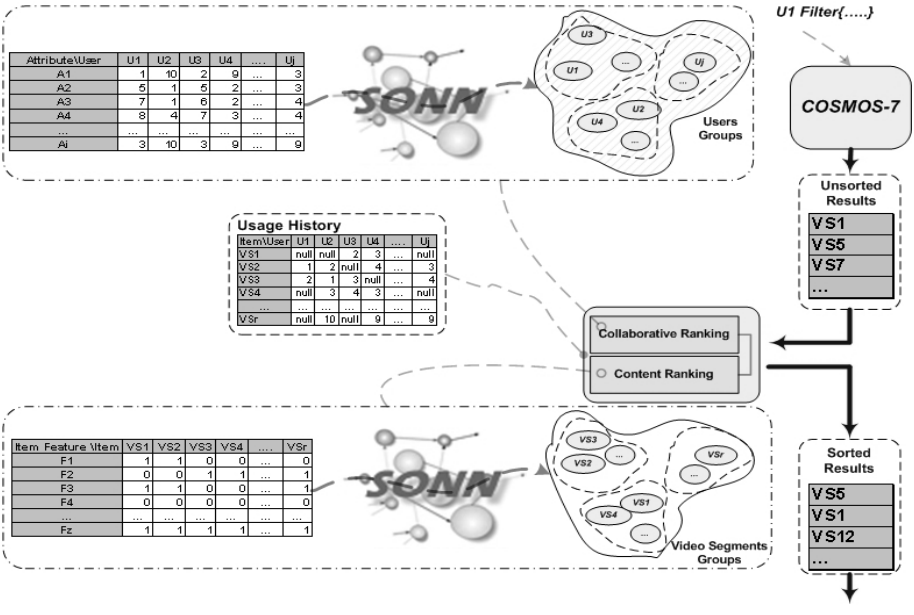


Fig. 3. COSMOS-7 hybrid ranking module for producing personalized ranking

(visual segment) is a neural network input vector with attribute values taken from its content model.

$$R_{content}(I_k, U_i) = \frac{\sum_{I_n \in G(I_k)} r(I_n, U_i)}{\sum_{I_n \in G(I_k)} 1} \tag{2}$$

The advantage of the system is that the SONN adapts the user's peer group as their preferences change over time. As the user interacts with the content the usage history log is updated. The usage history is reduced dimensionally into attribute pairs that are used to update the user profile. The changed user profile is then used to recalculate the user's peer group. This allows the user's changing requirements to be tracked automatically and reflected instantly in the ranking of the content. This process is iterative and makes peer matching more accurate over time.

Table 1 shows how COSMOS-7 ranking module manages to utilize the hybrid algorithm, which uses the strengths of both content based and collaborative methods to combat the disadvantages of both. Basically, the algorithm uses either the collaborative or content ranking technique depending on the conditions of the information available from the resultant segments. Method *getItemRank(I_k, U_i)* searches the usage history and returns the rating of item I_k, and user U_i. If the item hasn't been ranked by the current user before, collaborative ranking is used as explained above. However, if none of the peer group have ranked the particular item before, the content ranking is used.

Table 1. Pseudocode

```
Method rankItems(I [],Ui){
  For each Iz in I[]{
      Ik.Rank= getItemRank(Ik, Ui)
      If Ik.Rank!=null
         break
      else if ((∋ Un ∈ G(Ui)) ^ r( Ik Un )!=null)
         Ik.Rank =Rcollaborative(Ik,Ui)
      else If (∀Un ∈ G(Ui)=>r( Ik Un )=null)
         Ik.Rank =Rcontent(Ik,Ui)
      else
         Ik.Rank = defaultValue
  }
}
```

(3)

Figure 4 shows some preliminary results showing the user satisfaction rating of the COSMOS-7 system. They rank the first 10 segments that are represented to them in order, using a 1 -10 scale (1 = not satisfied at all < 10 = totally satisfied). The evaluation criterion in this case is based on the user's ranking of satisfaction to the

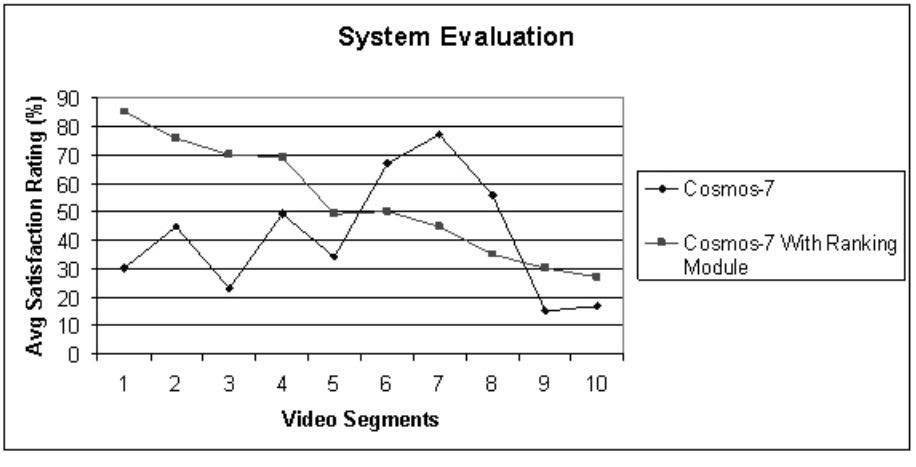


Fig. 4. Graph of user evaluation of COSMOS-7 user experience (ranked and unranked)

content presented to them. This experiment was carried out with a small number of users on two version of COSMOS-7; 1) Using the personalized ranking method and 2) The original chronologically ordered method. The results show that the user satisfaction is high initially with the personalized ranking method with the score decreasing proportionally as the user goes down the ranking as the segments become less relevant to the users. The unranked system shows a non-uniform satisfaction rating graph with a peak around the centre. This is due to the more relevant content

usually being found for most users around the middle/beginning of the end of the content model as a whole. This making the middle segments most relevant. Though the result have been derived with a small pool of content models using a few users, which makes it not fully representative, it still shows that our approach is very promising. A full featured user-evaluation will provide more precise results.

5 Conclusion

In this paper we propose a method that personalizes the order that video segments are presented to the user. The experimental data shows that the user experience is enhanced by presenting the video segments that are more aligned to a user's taste. This benefits the user by increasing the confidence they have in the systems ability to provide relevant data that fits the user's requirement for knowledge. The system improves over time as the user provides more feedback to the system. This enables the system to more accurately define the ranking process by attaining a higher degree of user preference specification. This also allows the user's change in preferences over time to be mapped over time. This enables the ranking process to evolve as the user's knowledge requirements change. Possible future research includes further experiments with more users and items, which will simulate a practical real environment. We'll also investigate how content based profiles, which are being updated during users interaction, can be utilized to calculate the similarity between users.

References

1. Angelides, M.C. and Agius, H.W., An MPEG-7 Scheme for Semantic Content Modeling and Filtering of Digital Video, ACM Multimedia Systems (2006)
2. ISO/IEC: Information Technology –Multimedia Content Description Interface – Part 5: Multimedia Description Schemes. Geneva, Switzerland, International Organisation for Standardisation (2002)
3. Koprinska, I. and Carrato,S., Temporal video segmentation: A survey, Signal Processing: Image Communication, 2001/1, 16, 5, 477-500
4. J. Assfalg, M. Bertini, C. Colombo and A.D. Bimbo, "Semantic annotation of sports videos," *Multimedia, IEEE,* vol. 9, pp. 52-60, Apr-Jun. 2002. Available: http://doi.ieeecomputersociety.org/10.1109/93.998060
5. R. Troncy, "Integrating Structure and Semantics into Audio-visual Documents", In Proc. of 2nd International Semantic Web Conference (ISWC), 20-23 October 2003, Sanibel Island, Florida, USA, pp 566-581 (2003)
6. T. Tran-Thuong and C. Roisin, "A multimedia model based on structured media and sub-elements for complex multimedia authoring and presentation," *IJSEKE,* vol. 12, pp. 473-500, 2002.
7. Wenyin, L., Chen, Z., Lin, F., Zhang, H., Ma, W.-Y.: Ubiquitous media agents: a framework or managing personally accumulated multimedia files. Multimedia Syst. 9(2), 144–156 (2003)

8. Ferman, A.M., Beek, J.H.E.P.V., Sezan,M.I.: Content-based filtering and personalization using structured metadata. In: Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries, p. 393. Portland, Oregon (2002)
9. Wallace, M.S.G.: Towards a context aware mining of user interests for consumption of multimedia documents. In: Proceedings of the 2002 IEEE International Conference on Multimedia and Expo, vol. 1, pp. 733–736 (2002)
10. Eirinaki, M., Vazirgiannis, M.: Web mining for web personalization. ACM Trans. Internet Technol. **3**(1), 1–27 (2003)
11. M. Lee, P. Choi and Y.E.-. Woo, A Hybrid Recommender System Combining Collaborative Filtering with Neural Network, In Adaptive Hypermedia and Adaptive Web-Based Systems: Second International Conference (AH 2002), Malaga, Spain, vol. 2347, pp 531-534 (2002).
12. Novak, J., Wurst, M., Fleischmann, M., Strauss, W.: Discovering, Visualizing and Sharing Knowledge through Personalized Learning Knowledge Maps, 2003 Spring Symposium on Agent-Mediated Knowledge Management, Technical Report SS-03-01, Stanford University, (2003), 101-108
13. Tan A-H, Pan H (2002) Adding personality to information clustering. Proceedings of the Sixth Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp 251–256.
14. Ultsch A. Self Organizing Neural Networks perform different from statistical k-means clustering. In Proccedings of GfKl,(1995), Basel, Swiss
15. Santosh K. Rangarajan, Vir V. Phoha, Kiran S. Balagani, Rastko R.Selmic, S.S. Iyengar, Adaptive Neural Network Clustering of Web Users. *Computer*, (2004) vol. 37, no.4, pp. 34-40

Classifier Fusion: Combination Methods For Semantic Indexing in Video Content

Rachid Benmokhtar and Benoit Huet

Département Communications Multimédias
Institut Eurécom
2229, route des crêtes
06904 Sophia-Antipolis - France
{Rachid.Benmokhtar, Benoit.Huet}@eurecom.fr

Abstract. Classifier combination has been investigated as a new research field to improve recognition reliability by taking into account the complementarity between classifiers, in particular for automatic semantic-based video content indexing and retrieval. Many combination schemes have been proposed in the literature according to the type of information provided by each classifier as well as their training and adaptation abilities. This paper presents an overview of current research in classifier combination and a comparative study of a number of combination methods. A novel training technique called Weighted Ten Folding based on Ten Folding principle is proposed for combining classifier. Experiments are conducted in the framework of the TRECVID 2005 features extraction task that consists in ordering shots with respect to their relevance to a given class. Finally, we show the efficiency of different combination methods.

1 Introduction

With the development of multimedia devices, more and more videos are generated every day. Despite the fact that no tools are yet available to search and index multimedia data, many individual approaches have been proposed by the research community. Video content interpretation is a highly complex task which requires many features to be fused. However, it is not obvious how to fuse them. The fusion mechanism can be done at different levels of the classification. The fusion process may be applied either directly on signatures (feature fusion) or on classifier outputs (classifier fusion). The work presented in this paper focuses on the fusion of classifier outputs for semantic-based video content indexing.

Combination of multiple classifier decisions is a powerful method for increasing classification rates in difficult pattern recognition problems. To achieve better recognition rates, it has been found that in many applications, it is better to fuse multiple relatively simple classifiers than to build a single sophisticated classifier.

There are generally two types of classifier combination: classifier selection and classifier fusion [1]. The classifier *selection* considers that each classifier is an expert in some local area of the feature space. The final decision can be taken only by one classifier, as in [2], or more than one "local expert", as in [3]. Classifier *fusion* [4] assumes that all classifiers are trained over the whole feature space, and are considered as competitive as well as complementary. [5] has distinguished the combination methods of

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 65–74, 2006.

different classifiers and the combination methods of weak classifiers. Another kind of grouping using only the type of classifiers outputs (class, measure) is proposed in [4].

Jain [6] built a dichotomy according to two criteria of equal importance: the type of classifiers outputs and their capacity of learning. This last criteria is used by [1,7] for grouping the combination methods. The trainable combiners search and adapt the parameters in the combination. The non trainable combiners use the classifiers outputs without integrating another *a priori* information of each classifiers performances.

As shown in figure 1, information coming from the various classifiers are fused to obtain the final classification score. Gaussian mixture models, neural network and decision templates are implemented for this purpose and evaluated in the context of information retrieval.

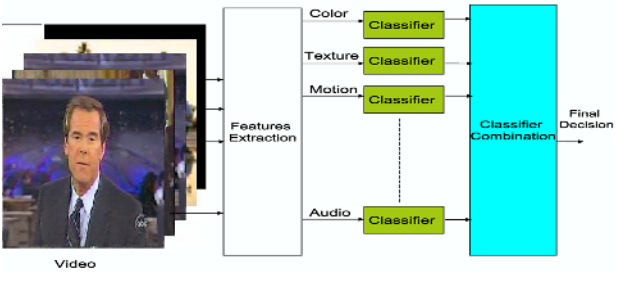


Fig. 1. General framework of the application

The paper presents the research we conducted toward a semantic video content indexing and retrieval system. It starts from a brief state of the art of existing combination methods and involved classifiers, including mixture of Gaussian, neural network and decision templates. All three methods are employed in turn to fuse and compared on the difficult task of semantic contents of video shots estimation . Then, we describe the visual and motion features that were selected. The results of our experiments in the framework of TRECVID 2005 are then presented and commented. Finally, we conclude with a summary of the most important results provided by this study along with some possible extension of work.

2 Combination of Different Classifiers

The classifiers may be of different nature, e.g. the combination of a neural network, a nearest neighbour classifier and a parametric decision rule, using the same feature space. This section starts by describing non-trainable combiners and continues with trainable ones.

2.1 Non Trainable Combiners

Here, we detail the combiners that are ready to operate as soon as the classifiers are trained, i.e., they do not require any further training. The only methods to be applied to combine these results without learning are based on the principle of vote. They are

commonly used in the context of handwritten text recognition [8]. All the methods of votes can be derived from the majority rule E with threshold expressed by:

$$E = \begin{cases} C_i \text{ if } \max\left(\sum_i^K e_i\right) \geq \alpha K \\ \\ \text{Rejection else} \end{cases} \tag{1}$$

where C_i is the i^{th} class, K is the number of classifiers to be combined and e_i is the classifier output.

For $\alpha = 1$, the final class is assigned to the class label most represented among the classifier outputs else the final decision is rejected, this method is called **Majority Voting**. For $\alpha = 0.5$, it means that the final class is decided if more half of the classifiers proposed it, we are in **Absolute Majority**. For $\alpha = 0$, it is a **Simple Majority**, where the final decision is the class of the most proposed among K classifiers. In **Weighted Majority Voting**, the answer of every classifiers is weighted by a coefficient indicating there importance in the combination [9].

The classifiers of type soft label outputs combine measures which represent the confidence degree on the membership. In that case, the decision rule is given by the **Linear Methods** which consist simply in applying to the outputs classifiers a linear Combination [10]:

$$E = \sum_{k=1}^{K} \beta_k m_i^k \tag{2}$$

where β_k is the coefficient which determines the attributed importance to k^{th} classifier in the combination and m_i^k is the answer for the class i.

2.2 Trainable Combiners

Contrary to the vote methods, many methods use a learning step to combine results. The training set can be used to adapt the combining classifiers to the classification problem. Now, we present four of the most effective methods of combination.

Neural Network (NN). Multilayer perceptron (MLP) networks trained by back propagation are among the most popular and versatile forms of neural network classifiers. In the work presented here, a multilayer perceptron networks with a single hidden layer and sigmoid activation function [11] is employed. The number of neurons contained in the hidden layer is calculated by heuristic. A description of the feature vectors given to the input layer is given in section 4.

Gaussian Mixture Models (GMM). The question with Gaussian Mixture Models is how to estimate the model parameter M. For a mixture of N components and a D dimensional random variable. In literature there exists two principal approaches for estimating the parameters: *Maximum Likelihood Estimation* and *Bayesian Estimation*. While there are strong theoretical and methodological arguments supporting Bayesian estimation, in this study the maximum likelihood estimation is selected for practical reasons.

For each class, we trained a GMM with N components, using Expectation Maximization (EM) algorithm [12]. The number of components N corresponds to the model

that best matches the training data. The likelihood function of conditional density models is:

$$p(x; M) = \sum_{i=1}^{N} \alpha_i \mathcal{N}(\mu_i, \Sigma_i)(x) \tag{3}$$

where α_i is the weight of the i^{th} component and $\mathcal{N}(.)$ is the Gaussian probability density function with mean μ_i and covariance Σ_i.

$$\mathcal{N}(\mu_i, \Sigma_i)(x) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_i|}} \exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right) \tag{4}$$

During the test, the class corresponding to the GMM that best fit the test data (according to the maximum likelihood criterion) is selected.

Decision Templates (DT). The concepts of decision templates as a trainable aggregation rule was introduced by [1,7]. Decision Template DT_k for each class $k \in \Omega$ (where Ω is the number of classes) can be calculated by the average of the local classifier outputs $P_m^n(x)$.

$$DT_k(m,n) = \frac{\sum_{x \in T_k} P_m^n(x)}{Card(T_k)} \tag{5}$$

where T_k is a validation set different from the classifier training set. Decision Templates is a matrix of size $[S, K]$ with S classifiers and K classes. To make the information fusion by arranging of K Decision Profiles (DP), it remains to determine which Decision Template is the most similar to the profile of the individual classification.

Several similarity measures can be used, e.g., the Mahalanobis norm (equ.6) and Swain & Ballard (equ.7) or the Euclidian distance (equ.8).

$$Sim(DP(x_i), DT^k) = \left(\sum_{m,n=1}^{S,K} (DP(x_i)_{m,n} - DT_{m,n}^k)\right)^T \Sigma^{-1} \left(\sum_{m,n=1}^{S,K} (DP(x_i)_{m,n} - DT_{m,n}^k)\right) \tag{6}$$

where: $m = 1, ..., S$, $n = 1, ..., K$ and Σ is the Covariance matrix.

$$Sim(DP(x_i), DT^k) = \frac{\sum_{m,n=1}^{S,K} \min(DP(x_i)_{m,n}, DT_{m,n}^k)}{\sum_{m,n=1}^{S,K} (DT_{m,n}^k)} \tag{7}$$

$$Sim(DP(x_i), DT^k) = 1 - \frac{1}{SK} \sum_{m=1}^{S} \sum_{n=1}^{K} (DP(x_i)_{m,n} - DT_{m,n}^k) \tag{8}$$

Finally, the decision is taken by the maximum of the similarity difference.

Genetic Algorithm (GA). Genetic algorithm have been widely applied in many fields involving optimization problems. It is built on the principles of evolution via natural selection: an initial population of individuals (chromosomes encoding the possible solutions) is created and by iterative application of the genetic operators (selection, crossover, mutation) an optimal solution is reached, according to the defined fitness function [13].

3 Combination of Weak Classifiers

In this case, large sets of simple classifiers are trained on modified versions of the original dataset. The three most heavily studied approaches are outlined here: reweighting the data (boosting-Adaboost), bootstrapping (bagging) and using random subspaces. Then, we introduce a new training method inspired from Ten Folding.

3.1 Adaboost

The intuitive idea behind AdaBoost is to train a series of classifiers and to iteratively focus on the hard training examples. The algorithm relies on continuously changing the weights of its training examples so that those that are frequently misclassified get higher and higher weights: this way, new classifiers that are added to the set are more likely to classify those hard examples correctly. In the end, AdaBoost predicts one of the classes based on the sign of a linear combination of the weak classifiers trained at each step. The algorithm generates the coefficients that need to be used in this linear combination. The iteration number can be increased if we have time and with the overfiting risk [14].

3.2 Bagging

Bagging builds upon bootstrapping and add the idea of aggregating concepts [15]. Bootstrapping is based on random sampling with replacement. Consequently, a classifier constructed on such a training set may have a better performance. Aggregating actually means combining classifiers. Often a combined classifier gives better results than individual base classifiers in the set, combining the advantages of the individual classifiers in the final classifier.

3.3 Random Subspace (RS)

The Random Subspace method consists to modify the learning data as in Bagging and Boosting. However, this modifications are realized on the features space. [15] showed that *RS* method allows to maintain a weak learning error and to improve the generalization error for the linear classifiers. It noticed that this method can outperform than the bagging and boosting if the number of features is big.

3.4 Ten Folding Training Approaches

Ten Folding (TF). In front of the limitation (number of samples) of TrecVid'05 test set, *N-Fold Cross Validation* can be used to solve this problem.

The principle of Ten Folding is to divide the data in $N = 10$ sets, where $N - 1$ sets are used for training data and the remaining to test data. Then, the next single set is chosen for test data and the remaining sets as training data, this selection process is repeated until all possible combination have been computed as shown in figure 2. The final decision is given by averaging the output of each model.

Weighted Ten Folding (WTF). With TrecVid'05 test set limitation in mind, the well-known Bagging instability [15] (i.e. a small change in the training data produces a big change in the behavior of classifier) and the overfitting risk for Adaboost (i.e. when the iteration number is big [14]), we propose a new training method based on Ten Folding that we call *Weighted Ten Folding*.

We use the Ten Folding principle to train and obtain N models weighted by a coefficient indicating the importance in the combination. The weight of each model is computed using the single set. The final decision combines measures which represent the confidence degree of each model.

The weighted average decision in WTF improves the precision of Ten Folding by giving more importance for models with weak training error, contrary to the Ten Folding who takes the output average of each model with the same weight.

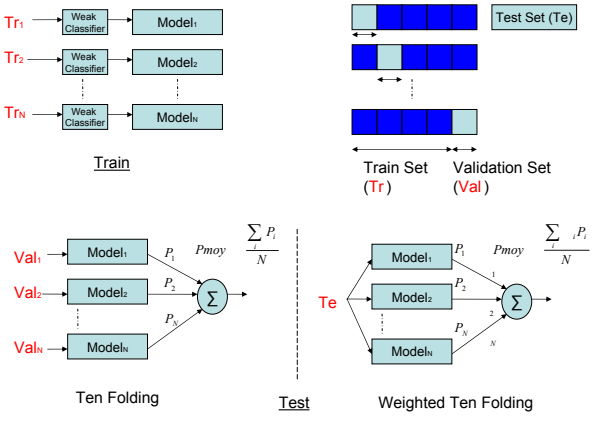


Fig. 2. The standard Ten Folding and Weighted Ten Folding combination classifier

4 Video Features

As far as this paper is concerned, we distinguish two types of modalities, visual and motion features, to represent video content.

4.1 Visual Features

To describe the visual content of a shot, features are extracted from key-frames. Two visual features are selected for this purpose: Hue-Saturation-Value color histograms and energies of Gabor's filters [16]. In order to capture the local information in a way that reflects the human perception of the content, visual features are extracted on regions of segmented key-frames [17]. Then, to have reasonable computation complexity and storage requirements, region features are quantized and key-frames are represented by a count vector of quantization vectors. At this stage, we introduce latent semantic indexing to obtain an efficient region based signature of shots. Finally, we combine the signature of the key-frame with the signatures of two extra frames in the shot, as it is described in [18], to get a more robust signature.

4.2 Motion Features

For some concepts like people walking/running, sport, it is useful to have an information about the motion activity present in the shot. Two features are selected for this purpose: the camera motion and the motion histogram of the shot. For sake of fastness, these features are extracted from MPEG motion vectors. The algorithm presented in [19] is used to estimate the camera motion. The average camera motion over the shot is computed and subtracted from macro-block motion vectors to compute the 64 bins motion histogram of moving objects in a frame. Then, the average histogram is computed over frames of the shot.

5 Experiments and Discussion

Experiments are conducted on the TRECVID 2005 databases [20]. It represents a total of over 85 hours of broadcast news videos from US, Chinese, and Arabic sources. About 60 hours are used to train the feature extraction system and the remaining for the evaluation purpose. The training set is divided into two subsets in order to train classifiers and subsequently the fusion parameters. The evaluation is realized in the context of TRECVID and we use the common evaluation measure from the information retrieval community: the mean precision.

The feature extraction task consists in retrieving shots expressing one of the following semantic concepts: *1:Building, 2:Car, 3:Explosion or Fire, 4:US flag, 5:Map, 6:Mountain, 7:Prisoner, 8:Sports, 9:People walking/running, 10:Waterscape*.

Figure 3 shows Mean Precision results for the trainable combiners presented in section (2.2), the NN improves the precision result for all semantic concept when compared with results obtained by Genetic Algorithm [18]. This improvement is clearly visible on the semantic concept (5, 10, 11: Mean Average Precision), where the GA approach had an overfitting problem which damaged the average precision.

Figure 4 shows the variation of Mean Average Precision results for Decision Templates using different norms (Swain & Ballard, Euclidean and Mahalonibis) for similarity

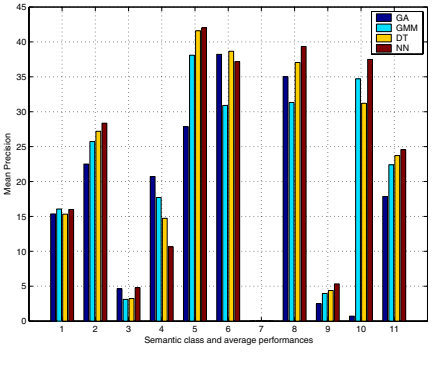


Fig. 3. Comparaison of Genetic Algorithm, Decision Templates method, GMM fusion method and Neural Network fusion method

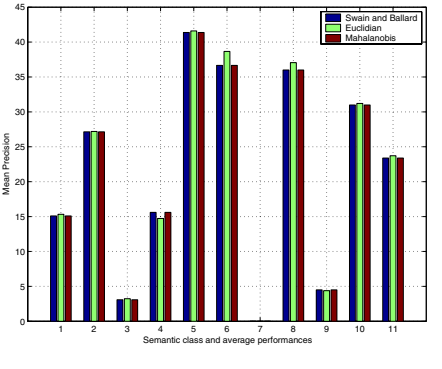


Fig. 4. Comparaison of Decision Templates performance with different norms (Swain & Ballard , Euclidean, Mahalanobis norm)

computation. Similar results are obtained for all three norms, which indicates that the Decision Templates method is more sensitive to data than to the chosen norm.

In the next experiment, Adaboost and Bagging principles are employed to increase the performances of GMM and Neural Network methods, considering them as weak classifier. As seen in figure 5, on average for all semantic concept the *WTF* approach outperforms in turn boosting, bagging and Ten Folding technique in spite of the lack of datum. Significant improvement have been noticed for the following semantic concepts (4, 5, 6, 8,11:Mean Average Precision). This can be explained by the weight computation, which is computed on a validation set independently to training set. This allows to have more representative weights in the test for the whole classifier. So, we have best level-handedness of whole classifier contrary to boosting, where the weights computation is made by the training set.

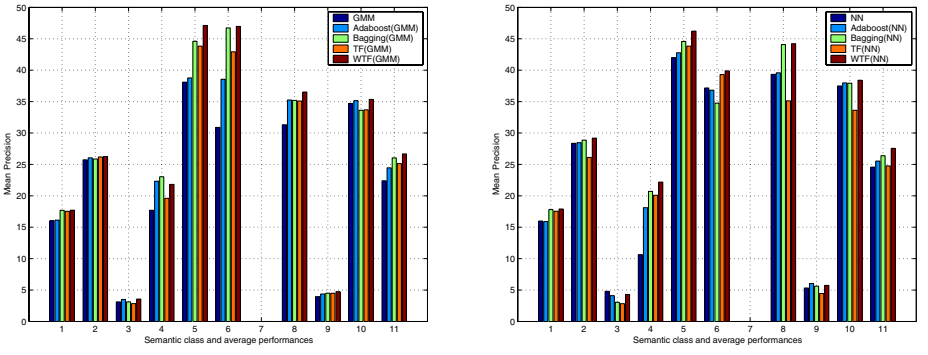


Fig. 5. Comparaison of performance using Adaboost, Bagging, Ten Folding and Weighted Ten Folding for GMM and NN

Fig. 6. Examples of first retrieved shots for waterscape, car and map classes

To conclude this section, figure 6 gives examples of first retrieved shots on TRECVID 2005 dataset for the classes waterscape, car and map to illustrate the efficiency of our classification method.

6 Conclusion

Fusion of classifiers is a promising research area, which allows the overall improvement of the system recognition performance. The work made on the combination also shows the multitude of combination methods which are different by their learning capacity and outputs classifier type.

Our experiments based on the TRECVID 2005 video database, show that Multilayer Neural Network and GMM approaches can improve the combination performance in comparison to the combination of multiple classifiers with averaging [21] and Genetic algorithm [13]. The results are very promising on the difficult problem of video shot content detection, using color, texture and motion features.

AdaBoost and Bagging as they were originally proposed did not show a significant improvement, despite their special base model requirements for dynamic loss and prohibitive time complexity. It is due to the TRECVID test set limitation and overfitting risk if the iteration number is big. The WTF resolves this last problem and improves Bagging and Adaboost results.

We have started to investigate the effect of the addition of many other visual features (Dominant Color, RGB, Canny edges features,...) as well as audio features (MFCC, PLP, FFT), to see their influence on the final result, and how the different approaches are able to deal with potentially irrelevant data. In parallel, we have initiated a program of work about descriptor fusion. We believe such an approach, which may be seen as normalization and dimensionality reduction [22], will have considerable effect on the overall performance of multimedia content analysis algorithms.

Acknowledgement

The work presented here is supported by the European Commission under contract FP6-027026-K-SPACE. This work is the view of the authors but not necessarily the view of the community.

References

1. L. Kuncheva, J.C.Bezdek, and R. Duin, “Decision templates for multiple classifier fusion : an experiemental comparaison,” *Pattern Recognition*, vol. 34, pp. 299–314, 2001.
2. L. Rastrigin and R. Erenstein, “Method of collective recognition,” *Energoizdat*, 1982.
3. R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, “Adaptive mixtures of local experts,” *Neural Computation*, vol. 3, pp. 1409–1431, 1991.
4. L. Xu, A. Krzyzak, and C. Suen, “Methods of combining multiple classifiers and their application to hardwriting recognition,” *IEEE Trans.Sys.Man.Cyber*, vol. 22, pp. 418–435, 1992.
5. R. Duin and D. Tax, “Experiements with classifier combining rules,” *Proc. First Int. Workshop MCS 2000*, vol. 1857, pp. 16–29, 2000.
6. A. Jain, R. Duin, and J. Mao, “Combination of weak classifiers,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, 2000.
7. L. Kuncheva, “Fuzzy versus nonfuzzy in combining classifiers designed by bossting,” *IEEE Transactions on fuzzy systems*, vol. 11, no. 6, 2003.

8. K. Chou, L. Tu, and I. Shyu, "Performances analysis of a multiple classifiers system for recognition of totally unconstrained handwritten numerals," *4th International Workshop on Frontiers of Handwritten Recognition*, pp. 480–487, 1994.
9. B. Achermann and H. Bunke, "Combination of classifiers on the decision level for face recognition," *technical repport of Bern University*, 1996.
10. T. Ho, *A theory of multiple classifier systems and its application to visual and word recognition*. PhD thesis, Phd thesis of New-York University, 1992.
11. G. Cybenko, "Approximations by superposition of a sigmoidal function," *Mathematics of Control, Signal and Systems*, vol. 2, pp. 303–314, 1989.
12. P. Paalanen, J. Kamarainen, J. Ilonen, and H. Kalviainen, "Feature representation and discrimination based on gaussian mixture model probability densities," *Research Report, Lappeenranta University of Technology*, 1995.
13. F. Souvannavong, B. Merialdo, and B. Huet, "Multi modal classifier fusion for video shot content retrieval," *Proceedings of WIAMIS*, 2005.
14. Y. Freud and R. Schapire, "Experiments with a new boosting algorithms," *Machine Learning : Proceedings of the 13th International Conference*, 1996.
15. M. Skurichina and R. Duin, "Bagging for linear classifiers," *Pattern Recognition*, vol. 31, no. 7, pp. 909–930, 1998.
16. W. Ma and H. Zhang, "Benchmarking of image features for content-based image retrieval," *Thirtysecond Asilomar Conference on Signals, System and Computers*, pp. 253–257, 1998.
17. C. Carson, M. Thomas, and S. Belongie, "Blobworld: A system for region-based image indexing and retrieval," *Third international conference on visual information systems*, 1999.
18. F. Souvannavong, B. Merialdo, and B. Huet, "Latent semantic analysis for an effective region based video shot retrieval system," *Proceedings of ACM MIR*, 2004.
19. R. Wang and T. Huang, "Fast camera motion analysis from mpeg domain," *Proceedings of IEEE ICIP*, pp. 691–694, 1999.
20. TRECVID, "Digital video retrieval at NIST," *http://www-nlpir.nist.gov/projects/trecvid/*.
21. S. Chang, W. Hsu, L. Kennedy, L. Xie, A. Yanagawa, E. Zavesky, and D. Zhang, "Video seach and high level feature extraction," *Proceedings of Trecvid*, 2005.
22. Y. Y. C. Zheng, "Run time information fusion in speech recognition," *Proc. of ICSLP*, 2002.

Retrieval of Multimedia Objects by Combining Semantic Information from Visual and Textual Descriptors*

Mats Sjöberg, Jorma Laaksonen, Matti Pöllä, and Timo Honkela

Laboratory of Computer and Information Science
Helsinki University of Technology
P.O. Box 5400, 02015 HUT, Finland
{mats, jorma, mpolla, tho}@cis.hut.fi
http://www.cis.hut.fi/picsom/

Abstract. We propose a method of content-based multimedia retrieval of objects with visual, aural and textual properties. In our method, training examples of objects belonging to a specific semantic class are associated with their low-level visual descriptors (such as MPEG-7) and textual features such as frequencies of significant keywords. A fuzzy mapping of a semantic class in the training set to a class of similar objects in the test set is created by using Self-Organizing Maps (SOMs) trained from automatically extracted low-level descriptors. We have performed several experiments with different textual features to evaluate the potential of our approach in bridging the gap from visual features to semantic concepts by the use textual presentations. Our initial results show a promising increase in retrieval performance.

1 Introduction

The amounts of multimedia content available to the public and to researchers has been growing rapidly in the last decades and is expected to increase exponentially in the years to come. This development puts a great emphasis on automated content-based retrieval methods, which retrieve and index multimedia based on its content. Such methods, however, suffer from a serious problem: the *semantic gap*, i.e. the wide gulf between the low-level features used by computer systems and the high-level concepts understood by human beings. In this paper we propose a method of using different textual features to help bridge the semantic gap from visual features to semantic concepts.

We have used our PicSOM [1] content-based information retrieval (CBIR) system with video data and semantic classes from the NIST TRECVID 2005[1] evaluation set. The TRECVID set contains TV broadcasts in different languages

* Supported by the Academy of Finland in the projects *Neural methods in information retrieval based on automatic content analysis and relevance feedback* and *Finnish Centre of Excellence in Adaptive Informatics Research.*

[1] http://www-nlpir.nist.gov/projects/trecvid/

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 75–83, 2006.

and textual data acquired by using automatic speech recognition software and machine translation where appropriate. Both the training and evaluation sets are is accompanied with verified semantic ground truth sets such as videos depicting explosions or fire.

The general idea is to take a set of example videos in the training set belonging to a given semantic class and map these onto the test set by using Self-Organizing Maps that have been trained with visual and textual feature data calculated from the video objects. This mapping generates different relevance values for the objects in the test set which can be interpreted as membership values of a fuzzy set corresponding the given semantic class. I addition to a basic set of visual and aural features, experiments comparing the retrieval accuracy with different textual features were performed. In this paper we discuss experiments using word histogram and keyword frequency features using SOMs and a binary keyword feature using an inverted file.

Section 2 describes the PicSOM CBIR system and Section 3 the TRECVID video data in more detail. Section 4 discusses how textual features can help in bridging the semantic gap between visual features and high-level concepts. The feature extraction methods are explained in Section 5 and the experiment results in Section 6. Finally, conclusions are drawn in Section 7.

2 PicSOM CBIR System

The content-based information retrieval system PicSOM [1] has been used as a framework for the research described in this paper. PicSOM uses several Self-Organizing Maps (SOMs) [2] in parallel to index and determine the similarity and relevance of database objects for retrieval. These parallel SOMs have been trained with different data sets acquired by using different feature extraction algorithms on the objects in the database. This results in each SOM arranging the objects differently, according to the corresponding feature.

Query by example (QBE) is the main interactive operating principle in PicSOM, meaning that the user provides the system a set of example objects of what he or she is looking for, taken from the existing database. This relevance information is used in the PicSOM system which expands the *relevance assessment* to related objects, such as keyframe images and textual data of a video.

For each object type (i.e. video, image, text), all relevant-marked objects in the database of that type get a positive weight inversely proportional to the total number of relevant objects of the given type. Similarly the non-relevant objects get a negative weight inversely proportional to their total number. The grand total of all weights is thus always zero for a specific type of objects. On each SOM, these values are summed into the best-matching units (BMUs) of the objects, which results in sparse value fields on the map surfaces.

After that the value fields on the maps are low-pass filtered or "blurred" to spread the relevance information between neighboring units. This produces to each map unit a *qualification value*, which is given to all objects that are mapped to that unit (i.e. have it as the BMU). Map areas with a mixed distribution of

positive and negative values will even out in the blurring, and get a low average qualification value. Conversely in an area with a high density of mostly positive values, the units will reinforce each other and spread the positive values to their neighbors. This automatically weights the maps according to relevance and coherence with the user's opinion.

The next processing stage is to combine the qualification values gained from each map to the corresponding objects. These values are again shared with related objects. The final stage is to select a specific number of objects of the desired target type with the highest qualification values. These will be returned to the user as retrieval results.

The PicSOM system has typically been used in interactive retrieval where the user can influence the response of the system with relevance feedback and the results will improve in each iteration. In this paper, however, we run only one non-interactive iteration, as we are merely interested in the mapping abilities of the SOMs for semantic concepts using textual and other features.

3 TRECVID Video Data

In 2005 our research group at Helsinki University of Technology took part in the TRECVID 2005 video retrieval evaluations [3]. The TRECVID data contains about 790 videos divided into a total of almost 100 000 video clips. From this set we picked only those that had some associated textual data and semantic classifications, resulting in a set of about 35 000 video clips. These video clips were used for the experiments described in this paper. Each video clip has one or many keyframes, which were representative still images taken from the video. Also the sound of the video was extracted as audio data. TRECVID provided textual data acquired by using automatic speech recognition software and machine translation from Chinese (Mandarin) and Arabic to English.

In the PicSOM system the videos and the parts extracted from these were arranged as hierarchical trees as shown in Fig. 1, with the main video as the parent object and the different extracted media types as child objects. In this way the relevance assessments can be transferred between related objects in the PicSOM algorithm as described in the previous section. From each media type different features were extracted, and Self-Organizing Maps were trained from these as is shown with some examples in the figure.

A large set of semantic sets were provided with the TRECVID data. These are each a set of video clips both in the training and test sets that belong to a given semantic class, for example videos depicting an exterior of a building. Table 1 shows the eight semantic classes that were used in our experiments. The first and second columns in the table give the number of videos in the training set and in the test set respectively. The given description is a shortened version of the one that was used when the classes were selected by hand during the TRECVID evaluations.

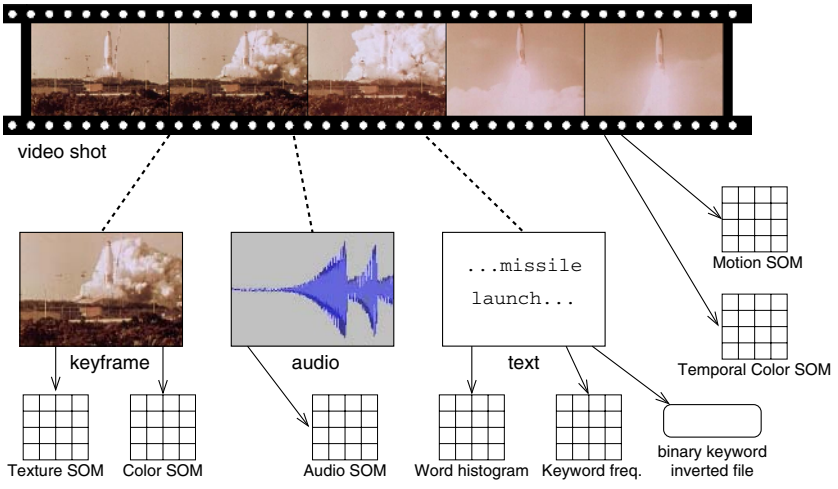


Fig. 1. The hierarchy of videos and examples of multi-modal SOMs

Table 1. Semantic classes from the TRECVID 2005 data set

training set	test set	description
109	265	an explosion or a fire
376	282	depicting regional territory graphically as a map
123	151	depicting a US flag
1578	943	an exterior of a building
375	420	depicting a waterscape or waterfront
23	32	depicting a captive person, e.g., imprisoned, behind bars
460	437	depicting any sport in action
1279	1239	a car

4 Bridging the Semantic Gap with Textual Features

The PicSOM system was initially designed for images, and particularly using visual features only. Such features describe images on a very low abstraction level, for example local color distributions, and do not generally correspond very well with the human perception of an image. In the experiments described in this paper we have also used video and aural features, but the problem remains the same: a very low-level feature description cannot match human understanding.

However, textual features do have a closer relationship to semantic concepts, as they describe the human language which has a much closer relation to the semantic concepts than for example low-level visual features. By including textual features we hope to bring the feature and concept levels closer and thus help to bridge the semantic gap. By using SOM techniques this is done in a fuzzy

manner, providing only semantic class membership values for each video, which is appropriate as such relationships can never be defined exactly, even by human beings.

Different textual features and retrieval methods exists. In this paper we will concentrate on the PicSOM system and try out three different textual features described in more detail in the following section.

5 Feature Extraction

5.1 Non-textual Features

From the videos we calculated the standard MPEG-7 Motion Activity descriptor using the MPEG-7 Experimentation Model (XM) Reference Software [4]. We also calculated our own non-standard temporal features of color and texture data.

A temporal video feature is calculated as follows. Each frame of the video clip is divided into five spatial zones: upper, lower, left, right and center. A still image feature vector is calculated separately for each zone and then concatenated to form frame-wise vectors. The video clip is temporally divided into five non-overlapping video sub-clips or slices of equal length. All the frame-wise feature vectors are then averaged within the slices to form a feature vector for each slice. The final feature vector for the entire video clip is produced by concatenating the feature vectors of the slices. For example using the 3-dimensional average RGB color still image feature we would get a final vector with a dimensionality of $3 \times 5 \times 5 = 75$. The idea is to capture how the averaged still image features change over time in the different spatial zones.

We used average RGB color, texture neighborhood and color moments each separately as a basis for the temporal feature algorithm. Texture neighbourhood is a simple textural feature that examines the luminance values of the 8-neighbourhood of each inner pixel in an image. The values of the feature vector are then the estimated probabilities that the neighbor pixel is brighter than the central pixel (given for each 8-neighborhood position).

If we treat the values in the different color channels of the HSV color space as separate probability distributions we can calculate the three first central moments: mean, variance and skewness. And when we calculate these for each of the five zones mentioned we get our third feature: color moments.

From the still images we calculated the following standardized MPEG-7 descriptors using the MPEG-7 XM software: Edge Histogram, Homogeneous Texture, Color Structure and Color Layout. Additionally we used a Canny edge detection feature which was provided by TRECVID.

From the audio data we calculated the Mel-scaled cepstral coefficient [5], i.e. the discrete cosine transform (DCT) applied to the logarithm of the mel-scaled filter bank energies. This feature is calculated using an external program created by the Speech recognition group at the Laboratory of Computer and Information Science at the Helsinki University of Technology[2].

[2] `http://www.cis.hut.fi/projects/speech/`

5.2 Word Histogram

The word histogram feature is calculated in three stages. First a histogram is calculated for each textual object (document) in the database giving the frequencies of all the words in that text. Then the document-specific histograms are combined into a single histogram or dictionary for the whole database. The final word histogram feature vectors are calculated for each document by comparing its word frequencies to the dictionary, i.e. the words in the database-wide histogram. For each word that does not belong to the list of stop words (i.e. not commonly used words such as "the") in the dictionary we calculate the tf-log-idf weight [6] for the document. The resulting feature vector then gives the tf-log-idf values for all dictionary words in that document.

The tf-idf weight is commonly used in information retrieval and is given as the product of the *term frequency* and the *inverse document frequency*. The term frequency for a word k in one document is calculated as

$$\mathrm{tf}_k = \frac{n_k}{\sum_{j \in K_D} n_j}, \tag{1}$$

where n_k is the number of occurrences of the word k and the denominator gives the number of occurrences of all dictionary words K_D in the document. The corresponding document frequency is calculated as

$$\mathrm{df}_k = \frac{N_k}{N}, \tag{2}$$

where N_k is the number of documents where the word k appears, and N is the total number of documents. The tf-log-idf is then given as the product of Eq. (1) and the log-inverse of Eq. (2):

$$\text{tf-log-idf}_k = \frac{n_k}{\sum_{j \in K_D} n_j} \log \frac{N}{N_k}. \tag{3}$$

The feature vector produced in this manner has a dimensionality of about 27 000. This is finally reduced to 100 by using singular value decomposition.

5.3 Keyword Frequency

Information about word occurrence frequencies were used to extract relevant keywords from each text document. Specifically, the frequency of occurrence for each word was compared to the corresponding frequency in another text corpus which was assumed to be neutral of domain specific terms. In these experiments, the Europarl corpus [7], extracted from European parliament proceedings, was used as the reference corpus. For each word, the ratio of the word's rank in the list of most frequent words in the document to the corresponding rank in the reference corpus was computed as an indicator of a semantically relevant keyword. Using this scheme words such as 'nuclear' would result in a high ratio despite rare occurrence in the document while words such as 'the', 'on' or 'and' would result in a low ratio regardless of frequent occurrence in the document.

5.4 Binary Keyword Features

A recent extension of the PicSOM system allows the usage of an inverted file as an index instead of the SOM [8]. The binary keyword feature is such a feature, where an inverted file contains a mapping from words to the database objects containing them.

The binary keyword features were constructed by gathering concept-dependent lists of most informative terms. Let us denote the number of video clips in the training set associated with semantic class c as N_c and assume that of these videos, $n_{c,t}$ contain the term t in the textual data. Using only non-stop words which have been stemmed using the Porter stemming algorithm [9], the following measure can be calculated for term t regarding the class c:

$$S_c(t) = \frac{n_{c,t}}{N_c} - \frac{n_{all,t}}{N_{all}} \ . \tag{4}$$

For every semantic class, we record the 10 or 100 most informative terms depending on which one gives better retrieval performance.

The inverse file is then created as mapping from these informative words to the database objects (texts) that contain them. In the PicSOM system a measure indicating the closeness of a textual object i to the semantic class c used in generating the inverse file can be calculated as

$$S_{i,c} = \sum_k \frac{\delta_{i,k}}{N_k}, \text{ where } \delta_{i,k} = \begin{cases} 1 & \text{if } k \text{ exists in } i, \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

and where sum is taken over all words k in the inverse file, and where N_k is the total number of textual objects that contain the keyword k. The higher the value of $S_{i,c}$ for a specific textual object is, the closer it is deemed to be to the given class c. The value of this measure is then added to the qualification values of objects produced by the visual and aural features.

6 Experiment Results

Four experiment runs are presented in this paper, each for a different combination of features used: (i) only non-textual features, and non-textual features combined separately with (ii) word histogram, (iii) keyword frequency, and (iv) binary keyword features. The binary keyword feature used different inverted files for each semantic class as explained previously. Each experiment was performed separately for each of the eight semantic classes, and the performance was evaluated using the average precision of retrieval.

The non-interpolated average precision is formed by calculating the precision after each retrieved relevant object. The final per-class measure is obtained by averaging these precisions over the total number of relevant objects, when the precision is defined to be zero for all non-retrieved relevant objects. The per-class average precision was finally averaged over all semantic classes to generate an overall average precision.

The experiment results are summarized in Table 2, with the best results for each class indicated in bold face. The results show how the retrieval performance increases as the textual features are used. Overall the binary keyword features make a substantial improvement, while the keyword frequency and word histogram features give much smaller improvements. If we look at the class-wise results, the binary keywords feature performs best in half of the cases, often with a considerable advantage over the other methods. Keyword frequency seems to do worst overall of the three textual features, but in three cases it is still better than the others, although with a small margin only.

One explanation for the relatively bad results of the keyword frequency and word histogram features is the low quality of the textual data. Speech recognition is never perfect, and machine translation reduces the quality even more. A visual inspection of the texts shows many unintelligible words and sentences. On the other hand, a sufficient number of correct words still seem to get through to make a significant difference. The fact that the binary keyword feature compares keywords with the rest of the database instead of a task-neutral external corpus, as the keyword frequency feature does, can explain the differences as well.

Table 2. Average precision results for experiments

semantic class	non-textual	kw freq.	word hist.	binary kw
an explosion or a fire	0.0567	0.0567	0.0582	**0.0680**
map of regional territory	0.3396	0.3419	0.3418	**0.3423**
depicting a US flag	0.0713	0.0715	0.0716	**0.0808**
an exterior of a building	0.0988	**0.0993**	0.0989	0.0972
waterscape or waterfront	0.2524	**0.2525**	0.2524	0.2500
captive person	0.0054	**0.0059**	0.0058	0.0029
any sport in action	0.2240	0.2242	0.2258	**0.2675**
a car	0.2818	0.2820	**0.2843**	0.2820
overall average	0.1662	0.1667	0.1674	**0.1739**

7 Conclusions

In this paper, we have studied the mapping of semantic classes of videos from a training set to a test set using Self-Organizing Maps. The nature of the resulting presentation of a semantic class can be understood as a fuzzy set where the relevance or qualification values of the retrieved videos can be interpreted as membership values. Furthermore we have studied the effect of using textual features in addition to our original non-textual, mostly visual, features. As textual features have a closer relation to the semantic concepts as expressed in language we hope to narrow the semantic gap and as a result increase the retrieval performance of our PicSOM CBIR system.

Our initial experiments do indeed demonstrate that this arrangement improves the performance of the system somewhat, although not in all cases as much as one might have hoped for. Especially the keyword frequency feature

has some future potential as new improvements are currently being implemented. The choice of reference corpus should be pondered, for example using several corpora would decrease the dependence of a specific choice. Also using the entire TRECVID textual database set itself as a corpora should increase accuracy. Our initial results however show a great potential for this method and inspires us to continue research in this area.

References

1. Laaksonen, J., Koskela, M., Oja, E.: PicSOM—Self-organizing image retrieval with MPEG-7 content descriptions. IEEE Transactions on Neural Networks, Special Issue on Intelligent Multimedia Processing **13**(4) (2002) 841–853
2. Kohonen, T.: Self-Organizing Maps. Third edn. Volume 30 of Springer Series in Information Sciences. Springer-Verlag, Berlin (2001)
3. Koskela, M., Laaksonen, J., Sjöberg, M., Muurinen, H.: PicSOM experiments in TRECVID 2005. In: Proceedings of the TRECVID 2005 Workshop, Gaithersburg, MD, USA (2005) 262–270
4. MPEG: MPEG-7 visual part of the eXperimentation Model (version 9.0) (2001) ISO/IEC JTC1/SC29/WG11 N3914.
5. Davis, S.B., Mermelstein, P.: Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In Waibel, A., Lee, K., eds.: Readings in speech recognition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1990) 65–74
6. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. Computer Science Series. McGraw-Hill, New York (1983)
7. Koehn, P.: Europarl: A multilingual corpus for evaluation of machine translation. (`http://people.csail.mit.edu/~koehn/publications/europarl.ps`) Draft, Unpublished.
8. Koskela, M., Laaksonen, J., Oja, E.: Use of image subset features in image retrieval with self-organizing maps. In: Proceedings of 3rd International Conference on Image and Video Retrieval (CIVR 2004), Dublin, Ireland (2004) 508–516
9. Porter, M.: An algorithm for suffix stripping. Program **14**(3) (1980) 130–137

A Relevance Feedback Approach for Content Based Image Retrieval Using Gaussian Mixture Models

Apostolos Marakakis[1], Nikolaos Galatsanos[2], Aristidis Likas[2], and Andreas Stafylopatis[1]

[1] School of Electrical and Computer Engineering, National Technical University of Athens, 15773 Athens, Greece
[2] Department of Computer Science, University of Ioannina, 45110 Ioannina, Greece
el00077@central.ntua.gr, galatsanos@cs.uoi.gr, arly@cs.uoi.gr, andreas@cs.ntua.gr

Abstract. In this paper a new relevance feedback (RF) methodology for content based image retrieval (CBIR) is presented. This methodology is based on Gaussian Mixture (GM) models for images. According to this methodology, the GM model of the query is updated in a probabilistic manner based on the GM models of the relevant images, whose relevance degree (positive or negative) is provided by the user. This methodology uses a recently proposed distance metric between probability density functions (pdfs) that can be computed in closed form for GM models. The proposed RF methodology takes advantage of the structure of this metric and proposes a method to update it very efficiently based on the GM models of the relevant and irrelevant images characterized by the user. We show with experiments the merits of the proposed methodology.

1 Introduction

The target of content-based image retrieval (CBIR) is to retrieve relevant images from an image database based on their visual content. Users submit one or more example images for query. Then, the CBIR system ranks and displays the retrieved results in order of similarity. Most CBIR systems ([1] – [8]) represent each image as a combination of low-level features, and then define a distance metric that is used to quantify the similarity between images. A lot of effort has been devoted in developing features and strategies that capture human perception of image similarity in order to enable efficient indexing and retrieval for CBIR, see for example [5],[9],[10] and [16]. Nevertheless, low-level image features have a hard time capturing the human perception of image similarity. In other words, it is difficult using only low-level images features to describe the semantic content of an image. This is known in the CBIR community as the *semantic gap* problem and for a number of years it has been considered as the "holy grail" of CBIR [11].

Relevance feedback (RF), has been proposed as a methodology to ameliorate this problem, see for example [1] - [3] and [6] - [8]. RF attempts to insert the subjective human perception of image similarity into a CBIR system. Thus, RF is an interactive process that refines the distance metric of a query interacting with the user and taking into account his/her preferences. To accomplish this, during a round of RF users are

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 84 – 93, 2006.

required to rate the retrieved images according to their preferences. Then, the retrieval system updates the matching criterion based on the user's feedback, see for example [1] – [3], [6] – [8], [15] and [16].

Gaussian mixtures (GM) constitute a well-established methodology to model probability density functions (pdf). The advantages of this methodology such as adaptability to the data, modeling flexibility and robustness have made GM models attractive for a wide range of applications ([17] and [18]). The histogram of the image features is a very succinct description of an image and has been used extensively in CBIR, see for example [4] and [9]. As mentioned previously GM provide a very effective approach to model histograms. Thus, GM models have been used for the CBIR problem ([4], [14] and [17]). The main difficulty when using a GM model in CBIR is to define a distance metric between pdfs that separates well different models, and that can be computed efficiently. The traditionally used distance metric between pdfs the Kullback-Liebler (KL) distance cannot be computed in closed form for GM models. Thus, we have to resort to random sampling Monte-Carlo methods to compute KL for GMs. This makes it impractical for CBIR where implementation time is an important issue. In [14] the earth movers distance (EMD) was proposed as an alternative distance metric for GM models. Although the EMD metric has good separation properties and is much faster to compute than the KL distance (in the GM case) it still requires the solution of a linear program. Thus, it is not computable in closed form and is not fast enough for a CBIR system with RF.

In this paper we propose the use for RF of an alternative distance metric between pdfs which was recently proposed in [21]. This metric can be computed in closed form for GM models. In this paper we propose an efficient methodology to compute this metric in the context of RF. In other words, we propose a methodology to update the GM model of the image query based on the relevant images. Furthermore, we propose an effective strategy that requires very few computations to update this distance metric for RF. The rest of this paper is organized as follows: in section 2 we describe the distance metric. In section 3 we present the proposed RF methodology based on this metric. In section 4 we present experiments of this RF methodology that demonstrate its merits. Finally, in section 5 we present conclusions and directions for future research.

2 Gaussian Mixture Models for Content-Based Image Retrieval

GM models have been used extensively in many data modeling applications. Using them for the CBIR problems allows us to bring to bear all the known advantages and powerful features of the GM modeling methodology, such as adaptability to the data, modeling flexibility, and robustness that make it attractive for a wide range of applications ([18] and [19]). GM models have been used previously for CBIR, see for example [4] and [14], as histograms models of the features that are used to describe images. A GM model is given by

$$p(x_i) = \sum_{j=1}^{K} \pi_j \phi(x_i / \theta_j) \tag{1}$$

where K is the number of components in the model, $0 \le \pi_j \le 1$ the mixing probabilities of the model with $\sum_{j=1}^{K} \pi_j = 1$, and $\phi\left(x_i/\theta_j\right) = N\left(x_i : \theta_j = \left[\mu_j, \Sigma_j\right]\right)$ a Gaussian pdf with mean μ_j and covariance Σ_j.

In order to describe the similarity between images in this context a distance metric must be defined. The Kullback-Leibler (KL) distance metric is the most commonly used distance metric between pdfs, see for example [10]. However, the KL distance cannot be computed in closed form for GMs. Thus, one has to resort to time consuming random sampling Monte Carlo methods. For this purpose a few alternatives have been proposed. In [14] the Earth Movers Distance (EMD) metric between GMs was proposed. This metric is based on considering the probability mass of one GM as piles of earth and of the other GM as holes in the ground and then finding the least work necessary to fill the wholes with the earth in the piles. EMD is an effective metric for CBIR however it cannot be computed in closed form and requires the solution of a linear program each time it has to be computed. This makes it slow and cumbersome to use for RF.

In order to ameliorate this difficulty a new distance metric was proposed in [21]. This metric between two pdfs $p_1(x)$ and $p_2(x)$ is defined as

$$C2\left(p_1, p_2\right) = -\log\left[\frac{2\int p_1(x)\, p_2(x)\, dx}{\int p_1^2(x)\, dx + \int p_2^2(x)\, dx}\right] \tag{2}$$

and can be computed in closed form when $p_1(x)$ and $p_2(x)$ are GMs. In this case it is given by

$$C2\left(p_1, p_2\right) = -\log\left[\frac{2\sum_{i,j} \pi_{1i}\pi_{2j} \sqrt{\frac{\left|V_{12}(i,j)\right|}{e^{k_{12}(i,j)}\left|\Sigma_{1i}\right|\left|\Sigma_{2j}\right|}}}{\sum_{i,j} \pi_{1i}\pi_{1j} \sqrt{\frac{\left|V_{11}(i,j)\right|}{e^{k_{11}(i,j)}\left|\Sigma_{1i}\right|\left|\Sigma_{1j}\right|}} + \sum_{i,j} \pi_{2i}\pi_{2j} \sqrt{\frac{\left|V_{22}(i,j)\right|}{e^{k_{22}(i,j)}\left|\Sigma_{2i}\right|\left|\Sigma_{2j}\right|}}}\right] \tag{3}$$

where

$$V_{ml}(i,j) = \left(\Sigma_{mi}^{-1} + \Sigma_{lj}^{-1}\right)^{-1},$$

$$k_{ml}(i,j) = \left(\mu_{mi} - \mu_{lj}\right)^T \left(\Sigma_{mi} + \Sigma_{lj}\right)^{-1} \left(\mu_{mi} - \mu_{lj}\right),$$

π_{mi} the mixing weight of the i-th Gaussian kernel of p_m, and, finally, μ_{mi}, Σ_{mi} are mean and covariance matrices for the kernels of the Gaussian mixture p_m.

3 Relevance Feedback Based on the *C2* Metric

For a metric to be useful in RF, it is crucial to be easily updated based on the relevant images provided by the user. Thus, assume we have a query modeled as $GMM(q)$, and the database images modeled by $GMM(d_i)$ for $i = 1,...,N$. The search based on this query requires the calculation of a $N \times 1$ table of the distances $C2(q, d_i)$. Also assume that from the retrieved images the user decides that the images with models $GMM(r_m)$ $m = 1,2 \ldots M$ are the most relevant and desires to update his query based on them. One simple and intuitive way to go about it is to generate a new GM model given by

$$GMM(q') = (1-\Lambda)\, GMM(q) + \sum_{m=1}^{M} \lambda_m GMM(r_m) \tag{4}$$

where $0 \le \lambda_m \le 1$, $\sum_{m=1}^{M} \lambda_m = \Lambda$, $0<\Lambda<1$, and is the relevance that is assigned to the image r_m by the user. The attractive feature of the model in Eq. (4) is that relevance λ_m has a physical meaning; it is proportional to the relevance degree assigned by the user and this defines a "composite GM model" that also includes the user preferences.

Furthermore, it is desirable to be able to efficiently compute the distances between the entries $GMM(d_i)$ for $i = 1,2 \ldots N$ and the new query model $GMM(q')$. Based on Eq. (3), the distance C2 is composed by sums of the type $S_{ml} = \sum_{i,j} \pi_{mi} \pi_{lj} \sqrt{\frac{|V_{ml}(i,j)|}{e^{k_{ml}(i,j)} |\Sigma_{mi}| |\Sigma_{lj}|}}$ where m, l indicate the GM models and i, j the Gaussian components. Based on Eq. (4) the update of the distance measure for the new query q' is given by:.

$$C2(q', d_i) = -\log \left[\frac{2(1-\Lambda) S_{qi} + 2\sum_{r} \lambda_r S_{ri}}{(1-\Lambda)^2 S_{qq} + 2(1-\Lambda)\sum_{r} \lambda_r S_{qr} + \sum_{r}\sum_{r'} \lambda_r \lambda_{r'} S_{rr'} + S_{ii}} \right] \tag{5}$$

The relevant images, indicated by r are the database images selected by the user. Since we can a priori compute (and store) the S_{ij} for all the images of the database and since all S_{qi} have already been computed in the previous query, the computation of the distance between $GMM(q')$ and the database image models is very fast since it involves only rescaling operations based on the relevance probabilities λ_r. Another nice property (for relevance feedback) of the model in Eq. (4), is that it can be generalized for any $C2(q)$ which models distance between histograms. In other

words, the pdfs $p_1(x)$ and $p_2(x)$ need not be GMs and could be even simple histograms.

The images retrieved by the system at each retrieval epoch that are not selected by the user as relevant, can be regarded as irrelevant. The determination of the irrelevant images could also be done in a more sophisticated manner that involves explicit selection by the user. Thus, using such images negative feedback can be provided and exploited to update the query. We can thus define in a way similar to Eq. (4) an updated query $GMM(n')$ for the irrelevant images:

$$GMM(n') = (1-\Lambda_n)\,GMM(n) + \sum_m \lambda_m^n GM(r_m^n) \tag{6}$$

where n, n' correspond to the negative query and Λ_n, λ_m^n are analogous to the previously mentioned Λ, λ_m. The best images to retrieve can be found by combining both positive and negative RF. This can be done by minimizing the following distance metric:

$$c(i) = a_{pos} d(q,i) + (1-a_{pos})(1-d(n,i)) \tag{7}$$

where

$$d(q,i) = \frac{C2(q,d_i)}{\max_i C2(q,d_i)} \quad \text{and} \quad d(n,i) = \frac{C2(n,d_i)}{\max_i C2(n,d_i)}$$

with $0 \le a_{pos} \le 1$. After computing the metric $c(i)$ for every database image, we can retrieve the images with the smallest value for this measure. These images will have the property of being near to the user ideal query, which is determined by the initial query and the positive examples, and far away of the user negative examples.

4 Experimental Results

In order to test the validity of this approach we used about 1000 annotated low resolution images from the image database in [22]. These images have been manually separated into 12 semantic categories according to their content (e.g. bears, butterflies, earth pictures etc). The features extracted by the images pixels correspond to the color scheme CIE-Lab ([14]). The GM parameters for each image were estimated with the very popular EM algorithm which for robustness was initialized with multiple runs of k-means algorithm. The number of components for every image was chosen empirically to be 5. In all the experiments, we chose to use full covariance for the GM components. A simple graphical user interface has been developed in order to visualize the results of our relevance feedback scheme. The user can choose the number of images which the system will retrieve at each round, the value of parameters Λ, Λ_n, the positive examples weight a_{pos} and the database image which

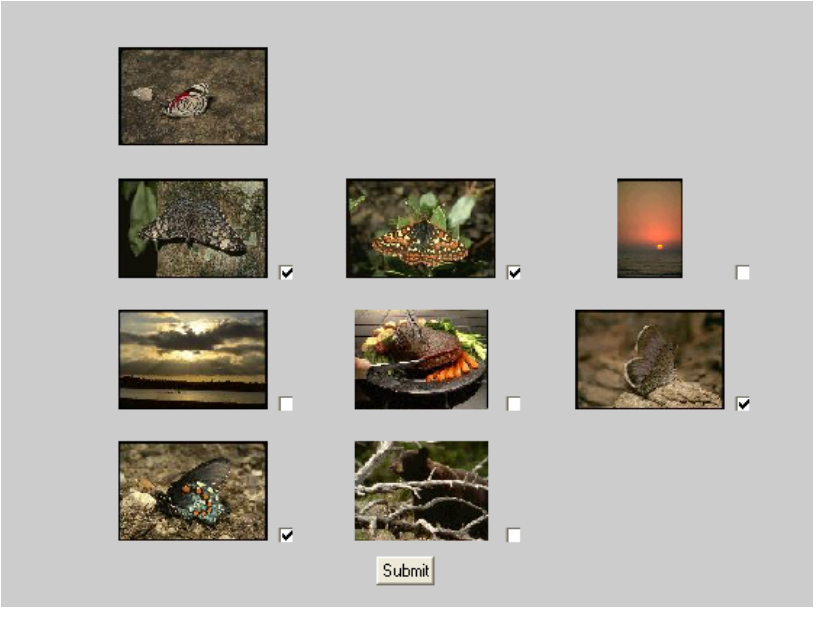


Fig. 1. Initial set of retrieved images by the system and user relevant images selection

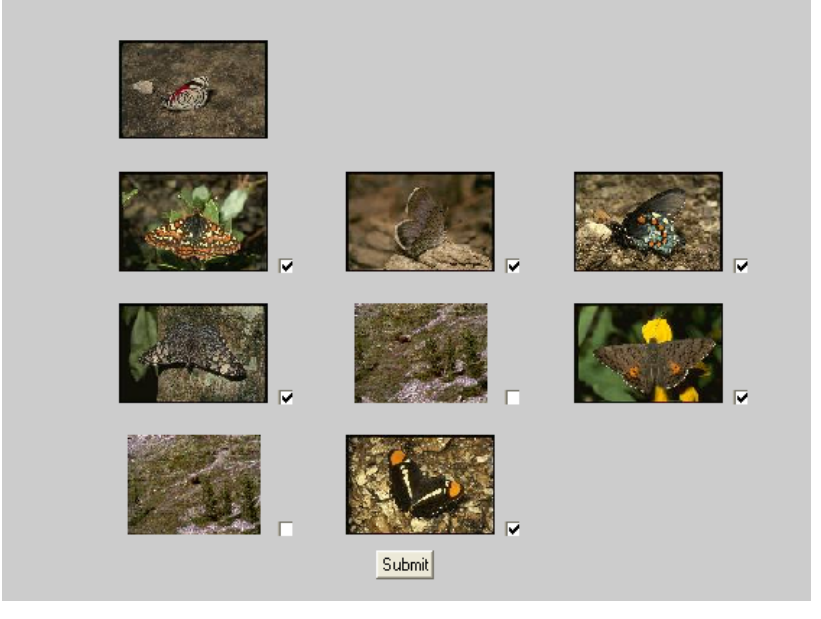


Fig. 2. Retrieved images and user choices after the first RF stage

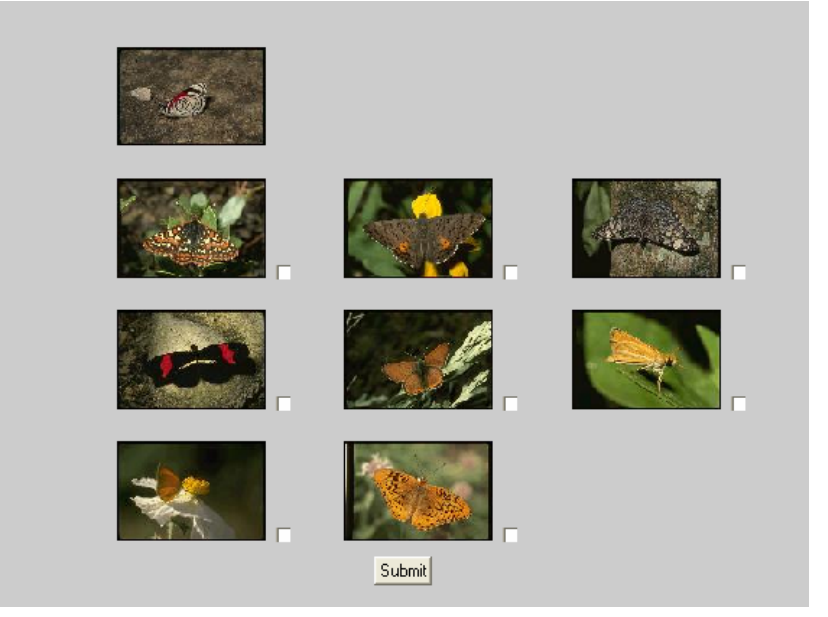


Fig. 3. Retrieved images after the second RF stage

will constitute the initial query. The parameters λ_m, λ_m^n for the positive and negative examples are given equal values regardless of m, because user is not required to specify the relative degree of relevance or irrelevance of the chosen images. In the Figures 1-3, a typical evolution of the RF process is demonstrated for $\Lambda = \Lambda_n = a_{pos} = 0.8$. The image on the top is the initial user query and the images regarded as the most relevant by the system are displayed from the left to the right and from the second row to the last row. The feedback of the user is provided by selecting the relevant images. The retrieval of the images is performed very rapidly due to the efficient way of computing the distances (Eq. 5).

In order to quantify the performace of the system we designated a relevance feedback simulation. In this simulation scheme, each image of the database was used as a query and relevant images were retrieved until three specific Recall Levels [20] (RecLev= 0.05, 0.1 and 0.2) are reached. For these Recall Levels the Precision [20] was specified. The retrieved relevant images were also specified and the percentage of them used for relevance feedback in this simulation is denoted as rprc. Also with nrprc we denote the percentage of non relevant retrieved images used for negative relevance feedback. The relevant and irrelevant images used in RF are selected at random from the sets of retrieved relevant and irrelevant images respectively. Tables 1 and 2 show the progression in Precision for three Recall levels during different rounds of relevance feedback averaged over the entire database. In the first table results we neglect the negative examples ($a_{pos} = 1$) and in the second table we include

Table 1. Average Precision over the entire database at given Recall levels during different rounds of relevance feedback of positive examples

RecLev	Initial	1st RF	2nd RF	3rd RF	4th RF	5th RF
0.05	0.6232	0.76736	0.81253	0.82745	0.83502	0.84404
0.1	0.5572	0.73305	0.77221	0.78665	0.79092	0.80075
0.2	0.46628	0.6438	0.66458	0.66935	0.67545	0.67633

(a) rprc = 0.5

RecLev	Initial	1st RF	2nd RF	3rd RF	4th RF	5th RF
0.05	0.6232	0.84622	0.87761	0.88669	0.88761	0.88848
0.1	0.5572	0.78157	0.81357	0.82486	0.83003	0.83309
0.2	0.46628	0.67363	0.69462	0.70154	0.70801	0.71063

(b) rprc = 1.0

Table 2. Average Precision over the entire database at given Recall levels during different rounds of relevance feedback of positive and negative examples

RecLev	Initial	1st RF	2nd RF	3rd RF	4th RF	5th RF
0.05	0.6232	0.81673	0.85783	0.88485	0.90192	0.90261
0.1	0.5572	0.7863	0.82877	0.84927	0.85469	0.85271
0.2	0.46628	0.72084	0.76122	0.76303	0.77079	0.7687

(a) rprc = 0.5, nrprc = 0.5

RecLev	Initial	1st RF	2nd RF	3rd RF	4th RF	5th RF
0.05	0.6232	0.9036	0.93898	0.94864	0.95506	0.95733
0.1	0.5572	0.84827	0.89067	0.90473	0.9089	0.91897
0.2	0.46628	0.76406	0.80179	0.82141	0.825	0.83261

(b) rprc = 1.0, nrprc = 1.0

them in the feedback using $a_{pos} = 0.8$. The weights given to the previous query models (q, n) and to each of the corresponding feedback examples have been chosen empirically to be equal.

5 Conclusions – Future Work

A probabilistic framework for relevance feedback based on GM models was proposed in this paper. The main advantages of the proposed methodology are accuracy as indicated by our simulation study, speed of implementation and flexibility. The treatment of the positive and the negative feedback examples is performed in a very intuitive way which in combination with the simple form of the distance C2 leads to the possibility of real time evaluation of the image ranking criterion, thus allowing for fast retrieval after user feedback has been specified.

In the future we plan to incorporate in the image models non-color features like texture, shape etc in addition to the color ones. Furthermore, we intend to provide the user with the possibility to determine explicitly the degree of relevance of his feedback examples. In addition, we aim to generalize our RF scheme to support region-based similarity and retrieval. Finally, we are aiming at testing the system to larger image databases.

Acknowledgement. This work was supported by the Greek General Secreteriat for Research and Technology under the PENED 2003 program.

References

1. Y. Ishikawa, R. Subramanya, and C. Faloutsos, "MindReader: Querying databases through multiple examples," presented at the 24th VLDB Conf., 1998.
2. I. J. Cox, M. L. Miller, T. P.Minka, T. V. Papathomas, and P. N. Yianilos, "The Bayesian image retrieval system, PicHunter: Theory, implementation, and psychophysical experiments," *IEEE Trans. Image Process*., vol. 9, no. 1, pp. 20–37, Jan. 2000
3. Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback:A power tool for interactive content-based image retrieval," *IEEE Trans.Circuits Syst. Video Technol*., vol. 8, no. 5, pp. 644–655, Sep. 1998.
4. C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image segmentation using expectation-maximization and its application to image querying," *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 24, no. 8, pp. 1026–1038, Aug. 2002
5. Y. Chen and J. Z. Wang, "A region-based fuzzy feature matching approach to content-based image retrieval," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 9, pp. 1252–1267, Sep. 2002.
6. G. D. Guo, A. K. Jain, W. Y. Ma, and H. J. Zhang, "Learning similarity measure for natural image retrieval with relevance feedback," *IEEE Trans. Neural Netw*., vol. 13, no. 4, pp. 811–820, Jul. 2002.
7. G. Aggarwal, T. V. Ashwin, and S. Ghosal, "An image retrieval system with automatic query modification," *IEEE Trans. Multimedia*, vol. 4, pp. 201–214, Jun. 2002
8. F. Jing, M. Li, H. J. Zhang, and B. Zhang, "An efficient and effective region-based image retrieval framework," *IEEE Trans. Image Process*.,vol. 13, no. 5, pp. 699–709, May 2004.
9. B. S. Manjunath, J. R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Trans. Circuits Syst. Video Technol*., vol. 11, no. 6, pp. 703–715, Jun. 2001.
10. M. N. Do and M. Vetterli, "Wavelet-based texture retrieval using generalized Gaussian density and Kullback- Leibler distance," IEEE Trans.Image Process., vol. 11, no. 2, pp. 146–158, Feb. 2002.
11. Y. Rui, T. S. Huang, and S. F. Chang, "Image retrieval: Current techniques, promising directions, and open issues," *J. Vis. Commun. Image Represen*., vol. 10, pp. 39–62, Mar. 1999.
12. Z. Su, H. Zhang, S. Li, and S. Ma, "Relevance feedback in content-based image retrieval: Bayesian framework, feature subspaces, and progressive learning," *IEEE Trans. Image Process*., vol. 12, no. 8, pp. 924–937, Aug. 2003.
13. X. He, O. King,W. Y. Ma,M. Li, and H. J. Zhang, "Learning a semantic space from user's relevance feedback for image retrieval," *IEEE Trans.Circuits Syst. Video Technol*., vol. 13, no. 1, pp. 39–48, Jan. 2003.

14. H. Greenspan,a, G. Dvir,a and Y. Rubnerb, "Context-dependent segmentation and matching in image databases", *Computer Vision and Image Understanding*, Vol. 93 pp. 86–109, 2004.
15. El. Naqa, Y. Yang, and N. Galatsanos and M. Wernick. "A Similarity Learning Approach to Content Based Image Retrieval: Application to Digital Mammography", *IEEE Transactions on Medical Imaging*, Volume: 23 , Issue: 10 , pp:1233-1244, Oct. 2004.
16. C.T. Hsu, and C. Y. Li, "Relevance Feedback Using Generalized Bayesian Framework With Region-Based Optimization Learning", *IEEE Trans. on Image Proc.*, Vol. 14, No. 10, pp. 1617-1631, October 2005
17. F. Qian, M. Li, L. Zhang, H. J. Zhang, and B. Zhang, "Gaussian mixture model for relevance feedback in image retrieval," presented at the *IEEE ICME*, Aug. 2002.
18. C. M. Bishop, Neural Networks for Pattern Recognition. Oxford Univ. Press Inc., New York, 1995.
19. G. M. McLachlan and D. Peel, Finite Mixture Models. New York: John Wiley & Sons,Inc., 2001.
20. A. D. Bimbo, Visual Information Retrieval. San Mateo, CA: Morgan Kaufmann, 1999.
21. G. Sfikas, C. Constantinopoulos, A. Likas and N. P. Galatsanos, "An Analytic Distance Metric With Application In Image Retrieval For Gaussian Mixture Models", *International Conference Artificial Neural Networks*, 2005.
22. http://wang.ist.psu.edu/docs/related/

Video Representation and Retrieval Using Spatio-temporal Descriptors and Region Relations

Sotirios Chatzis[1], Anastasios Doulamis[1], Dimitrios Kosmopoulos[2], and Theodora Varvarigou[1]

[1] National Technical University of Athens, Electrical and Computer Engineering Department Athens, Greece
{Stchat, Adoulam, Dora}@telecom.ntua.gr
[2] Demokritos National Centre for Scientific Research, Institute of Informatics and Telecommunications, Athens, Greece
dkosmo@iit.demokritos.gr

Abstract. This paper describes a novel methodology for video summarization and representation. The video shots are processed in space-time as 3D volumes of pixels. Pixel regions with consistent color and motion properties are extracted from these 3D volumes by a space-time segmentation technique based on a novel machine learning algorithm. Each region is then described by a high-dimensional point whose components represent the average position, motion velocity and color of the region. Subsequently, the spatio-temporal relations of the regions are deduced and a concise, graph-based description of them is generated. This graph-based description of the video shot's content, along with the region centroids, comprises a concise yet powerful description of the video-shot and is used for retrieval applications. The retrieval problem is formulated as an inexact graph matching problem between the data video shots and the query input which is also a video segment. Experimental results on action recognition and video retrieval are illustrated and discussed.

Keywords: Spatio-Temporal, Graph Matching, Region, Machine Learning, ARVQ.

1 Introduction

Multimedia databases have been the subject of extensive research for the last several years. The major research goal is to progress towards content-based functionalities, such as search and manipulation of objects, semantic description of scenes, detection of unusual events, and recognition of objects. Current video indexing and retrieval methods typically do not analyze video structure at the object level. Low-level visual features, such as color, shape orientation, motion trajectory and texture are commonly used for indexing individual frames, without taking under consideration any temporal information though [2]. These approaches are not satisfactory because video is temporal media, therefore sequencing of individual frames creates new semantics concerning the temporal evolution of the video that are doomed to be lost when processing the video frames individually.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 94 – 103, 2006.

There is supporting evidence [5] that human vision finds salient structures jointly in space and time. The unification of the analysis of spatial and temporal information in video sequences, by constructing a volume of spatio-temporal data in which consecutive frames are stacked to form a third, temporal dimension, was first pioneered by Adelson and Bergen [4. The benefits from analyzing the motion and color features of the whole video sequence as opposed to processing each frame individually are overwhelmingly compelling. Spatio-temporal segmentation is a more direct and robust way of tracking moving regions than the classical tracking approaches that extend inferences from image pairs to multiple frames, since it capacitates us to reason about much longer-term dynamics.

In [3], a threshold-free hierarchical space-time segmentation technique is applied. Subsequently, the produced region information is used for the execution of retrieval operations. However, in [3] and other similar works, the retrieval procedure does not take under consideration the spatio-temporal relations between the video regions, thus it is not able to exploit the information about the overall context within which the regions are located. This context information provides significant extra semantics the use of which could increase dramatically the performance of the retrieval process. Furthermore, this approach cannot provide any information "on the fly" that is, while the video is captured, something that could be a significant deficiency in some applications.

In our perception, the analysis of videos in terms of interaction of consistent color and motion regions comprises a critical enhancement in video retrieval applications. First of all, it capacitates us to detect the occurrence of dynamic events in videos. Such an event could be for example the collision of two cars of a specified color. Furthermore, the proposed representation schema allows for the automatic acquisition of higher level semantics of the video. For instance, we could acquire the time a museum visitor spends in front of a specific antiquity.

In this paper, we propose a novel methodology for the compact representation of a video's spatio-temporal structure and describe a system that utilizes this representation to perform retrieval procedures on the basis of color and motion characteristics of the regions as well as of the spatio-temporal relations between them. The system is capable of detecting human regions on the fly, i.e. as it is captured. We consider video shots of small duration. When the video capturing is finished, the system extracts the consistent motion and color regions and determines the spatio-temporal relations between them. This information can be further used for the execution of image retrieval applications. Our system can be applied in surveillance and data mining applications. For example, in the case of a museum, it could be used for security purposes as well as for the analysis of the visitors' preferences.

2 Approach Overview

2.1 Region Extraction

During the capture of the video, our system detects the (plausibly) existing human regions (faces) on a per frame basis. One way to detect humans is to apply an intelligent, adaptable neural network architecture, as we have proposed in one of our earlier

works [7]**.** When the video capturing is finished the system processes offline the overall video sequence to determine the rest video regions on the basis of color and motion features consistency. Hence, we conceive the video sequence as a 3D stack of frames where the pixels within this stack that correspond to a discrete object form a separate cluster of pixels with high motion and color affinity.

Each pixel is mapped to a 7D feature space with its dimensions representing the pixel position, optical flow motion and color. On the sequel, the system applies a novel machine learning algorithm for the aggregation of the pixels into clusters of coherent motion and color behavior. The basic conception of this algorithm, which shall be extensively presented in chapter 3, is that the pixel clusters representing coherent regions form dense hyper-spheres within the 7D feature space of a maximum radius that can be defined by the radii of the already extracted human regions. The seven dimensions of the found cluster centroids describe the positions of the region trajectories and their velocity vectors, as well as the average colors of these regions. Hence, the cluster centroids comprise spatio-temporal descriptors of the extracted regions, summarizing the location, color and dynamics of independently moving regions with only a small number of bytes. The similarities of sequences are defined using these descriptors. Finally, the regions are classified into two categories: *background regions* and *foreground regions.*

2.2 Graph-Based Video Representation

The recognition and understanding of complex scenes requires not only a detailed description of the objects involved, but also of the spatio-temporal relationships between them. Indeed, the diversity of the forms of the same object in different instantiations of a scene, and also the similarities of different objects in the same scene, make relationships between objects of prime importance in order to disambiguate the recognition of objects with similar appearance. Due to this necessity, after the computation of the spatio-temporal descriptors set, representing the regions the video shot has been segmented into, our system determines the relations between these regions in a spatio-temporal context and creates a concise *graph-based* representation of them.

Graph based representations are often used for scene representation in image processing [2]. Vertices of the graphs usually represent the objects in the scenes, while their edges represent the relationships between the objects. Relevant information for the recognition is extracted from the scene and represented by relational attributed graphs. In model-based recognition, both the model and the scene are represented by graphs.

The presented system introduces a novel algorithm for the *graph-based* representation of the spatio-temporal relations between the extracted objects (regions) of the video shot. As opposed to the conventional algorithms, approaching the video sequence as stack of frames, that firstly generate a representation scheme for each frame and then they correlate these by frame representations on the basis of their temporal relations, our approach adopts a single, unified, time-integral representation scheme for the video sequence on the whole. The proposed algorithm comprises the generation of an undirected, attributed graph *G(V, E)* where the vertices represent the regions the video sequence consists of and the edges represent the spatio-temporal relations between them. The generated graph representation has the following features:

- Its vertices represent the regions the video sequence has been segmented into.
- Each vertex is integrated with the spatio-temporal descriptor (the region centroid's feature vector) that describes the mean features of the region represented, as described above.
- The edges of the graph denote the pairs of regions that are spatio-temporally related. They connect:
  - Pairs of spatially adjacent background objects in a series of frames.
  - Pairs of foreground – background objects, where the foreground object is spatially adjacent to the background one in a series of frames.
  - Pairs of foreground objects, both appearing in some frames, that either have a boundary (thus, they are also spatially adjacent) or they are adjacent with the same background object for overlapping time slots (thus, they have a strong spatio-temporal relation).
- Each edge is embedded with the time slot the described relationship occurred within the video shot.

The generated graph-based spatio-temporal representation, along with the integrated in it spatio-temporal region descriptors, comprise a very concise yet powerful description of a video shot. The high-dimensionality of the used feature space in conjunction with the concrete description of the spatio-temporal relations between the regions minimizes the chances that several of the cluster centers from one sequence would simultaneously fall in the neighborhood of the cluster centers of another sequence. In our approach we obviously assume that the moves of the regions are linear. In real cases, objects might accelerate and make turns, and the camera that tracks them also might introduce scene motion. But, because of the inertia of objects and cameras, space-time regions produced over short durations in video sequences typically have an approximately linear behavior.

3 Offline Region Extraction Algorithm

Each pixel is mapped to a 7D feature space. These dimensions represent the three color coordinates of the pixel, the position of it *(x, y)* and its velocity coordinates (v_x, v_y) that are computed applying an optical flow analysis algorithm. The points of the described feature space that represent pixels of the same color region moving through time tend to be close together and to form a cluster. We exploit this feature by applying a clustering algorithm on the feature space. The centroids of the clusters are used for the region characterization. The seven components of the cluster centroids characterize average values of the motion velocity coordinates, object trajectory position, and colors of the regions through time.

For the region extraction we apply a novel machine learning algorithm, the Adaptive Resonance Learning Vector Quantization Algorithm (ARVQ). This new clustering algorithm proposed here is a modification of the Learning Vector Quantization Algorithm (LVQ) [8]. It extends the LVQ algorithm in the sense that

- During the first phase of the algorithm, a systematic representation of the a priory acquired knowledge is generated (human regions), which though represents only a fraction of the overall information existing.

- During the second phase is performed the fine tuning of the representation generated above, as well as the detection of the rest existing regions. Thus, during the second phase the fraction of the overall existing information that was not known beforehand is acquired and generated in the first phase representation of the a priory known information is refined.

The ARVQ algorithm, as applied in the context of the enhanced region extraction procedures of our system, is the following:

a. (Phase 1). The pixels of the a priory known regions (tracked human objects) are clustered together and the centroids of them are computed. The centroid of a cluster is defined as the mean of the vectors of the pixels belonging to it. Let us define as $\boldsymbol{x}_{ci}$ the i-th pixel of the c-th human face region with centroid $\boldsymbol{w}_c$. The minimum resemblance (Euclidean distance) of a human region (face) pixel to its region's centroid is computed, let it be denoted as

$$RM \equiv \max_{c}\{\max_{ci}\left\|\boldsymbol{w}_c - \boldsymbol{x}_{ci}\right\|\} \tag{1}$$

This quantity shall be used for the fine-tuning of the maximum intra-cluster resemblance threshold that shall be used for the extraction of the non human region clusters, during the second phase of the algorithm. The notion behind this procedure is the following: typically, the variance of the feature vectors of a human face is expected to be low under constant illumination conditions. However, in cases the illumination changes noticeably, the color components of the pixels of a face, present during this phenomenon, will undergo a significant change, something that will be depicted in the magnitude of the *RM* metric. To exploit this fact, the maximum intra-cluster resemblance threshold, is computed as follows:

$$T = T_r \exp\left(1/2|RM - \mathrm{E}\{RM\}|\right) \tag{2}$$

where T_r is the minimum intra-cluster resemblance threshold (determined heuristically) and $\mathrm{E}\{RM\}$ is the expected (mean) value of the *RM* parameter, computed from a big random sample (database) of face images.

b. (Phase 2). Let us consider as $\boldsymbol{x}$ the feature vector of a pixel. Let us also denote as $\boldsymbol{w}_k$ the centroid of the k-th cluster. Then for each pixel in the video stack:

- Find the cluster k for which

$$\left\|\boldsymbol{x} - \boldsymbol{w}_k\right\| = \min_{i}\left\|\boldsymbol{x}_i - \boldsymbol{w}_c\right\| \tag{3}$$

- If pixel $\boldsymbol{x}$ belongs to a human region represented by cluster k', different of k, then update the centroids of the clusters k and k' using the *fine tuning* rule

$$\Delta\boldsymbol{w}_k = -\gamma_{error}(\boldsymbol{x} - \boldsymbol{w}_k) \tag{4a}$$

$$\Delta\boldsymbol{w}_{k'} = \gamma_{error}(\boldsymbol{x} - \boldsymbol{w}_{k'}) \tag{4b}$$

where γ_{error} is the error case learning rate.

- If cluster k is a human region cluster and pixel $\boldsymbol{x}$ does not belong to a human region, try to find a cluster k' not representing human region such that:

$$\left\| \boldsymbol{x} - \boldsymbol{w}_{k'} \right\| = \min_{i} \left\| \boldsymbol{x}_i - \boldsymbol{w}_c \right\| \tag{5a}$$

where i is any cluster not representing a human region

$$\| \boldsymbol{x} - \boldsymbol{w}_{\mathbf{k}'} \| < T \tag{5b}$$

If such a cluster exists, allocate pixel $\boldsymbol{x}$ to this cluster (k') and update its centroid using the rule

$$\boldsymbol{w}_{\mathbf{k'},\mathbf{new}} = ((n\text{-}1)\, \boldsymbol{w}_{\mathbf{k'},\mathbf{old}})/n \tag{6}$$

where n is the number of the pixels within the cluster after the addition of the new pixel

Update also the human region cluster to which this pixel was erroneously allocated initially, by *the fine-tuning* rule (4a).

If a cluster complying with (5) does not exist, create a new one and set $\boldsymbol{x}$ as its centroid.

- If k is a non-human region cluster and pixel $\boldsymbol{x}$ does not belong to a human region, check whether the distance of pixel $\boldsymbol{x}$ from this cluster's centroid is lower that the resemblance threshold T. If such a cluster exists, allocate pixel $\boldsymbol{x}$ to cluster k and update its centroid by the rule

$$\boldsymbol{w}_{\mathbf{k,\ new}} = ((n\text{-}1)\, \mathbf{w}_{\mathbf{k,\ old}})/n \tag{7}$$

where n is the number of the pixels within the cluster after the addition of the new pixel

Else create a new cluster and set $\boldsymbol{x}$ as its centroid.

4 Video Retrieval Using the Spatio-temporal Representation Graph

The graph-based spatio-temporal video representation can be used for model-based video retrieval procedures. The user queries can be images or small video shots depicting information that the user is interested in, e.g. a collision of a blue and a red car. Our system generates the graph representation of the query input, which shall be referred to as the *model,* and conducts a graph matching operation against the graph representations of the data in the video database, which shall be referred to as the *data videos*. The retrieval problem is mathematically formulated as following: Given two undirected graphs with attributed vertices, the *model graph* G_1 (V_1,E_1) and the *data graph* $G_2(V_2,E_2)$, and a vertex resemblance function

$$r : V_1 \times V_2 \rightarrow \Re$$

check whether exists a 1-1 function $f : V_1 \rightarrow V_2$ such that

$$r(u, f(u)) = \max_{w \in V_2} \{ r(u, w) \}\ \forall u \in V_1 \tag{8a}$$

$$(f(u),\ f(v)) \text{ exists iff } (u, v) \text{ exists, where } u, v \in V_1 \tag{8b}$$

Hence, we demand that the information contained in the model video is also contained in the data video, although the data video might also include more information than that (i.e. more regions). The way we formulate the video retrieval problem, we avert affection by the over-segmentation of the data video sequence, since each model vertex is mapped to a single data vertex which, in case of over-segmentation is highly likely to be the vertex of the most representative sub-region of the over-segmented object.

Our video retrieval algorithm, solving problem (8), is applied for each data graph stored in our database. Let us denote as $G_1(V_1, E_1)$ the model graph and as $G_2(V_2, E_2)$ the data graph. The algorithm is the following:

1. For each pair of vertices (u, w), $u \in V_2$, $w \in V_1$, compute their resemblance $r(u,w)$.
2. For each model vertex $w \in V_1$, find the data vertex, $v \in V_2$ that resembles to it most, i.e.

$$r(v, w) = \max_{u \in V_2} \{r(v, u)\}$$

If $n>1$ model vertices $w_1,..,w_n$ are mapped to the same data vertex v, then map v to the model vertex w it resembles most and map the other model vertices to the second more similar to them data vertex. Let us denote $v=f(w)$.

3. For each edge $(w,w') \in E_1$ check whether the edge $(f(w),f(w')) \in E_2$. If it doesn't hold, then the model does not match with the data. Else
4. For each edge $(u,u') \in E_2$ check whether the edge $(f^{-1}(u),f^{-1}(u')) \in E_1$. If it doesn't hold, then the model does not match with the data. Else there is a matching.

The data videos matching with the model are retrieved and ranked using eq. (9) on the basis of the similarity between their regions and the corresponding model regions

$$h(f(G_1, G_2, r) = \sum_{u \in V_1} f(u, f(u)) \tag{9}$$

As resemblance metric of a vertices pair we use the Euclidean distance of their vectors.

In the proposed algorithm, we suppose that the time position and duration of the spatio-temporal relation - 'interaction' between two regions is not a matching criterion taken into account in the retrieval procedure. However, in applications where the time position of the interactions or the duration of them would also be of interest, the proposed algorithm can be extended so as to apply a resemblance metric also on the weights (time slots) of the graph edges, affecting the final ranking of each data item within the results list.

5 Experimental Evaluation

We have exhaustively tested the presented system using the CAVIAR [6] collection. In the following we shall elaborate on the results of two characteristic test cases. One of our evaluation test cases was based on the *Walk1* and *Walk3* test cases of the collection. These video clips were filmed with a wide angle camera lens in the entrance lobby of the *INRIA Labs* at Grenoble, France. In figure 1a and 1b we depict some characteristic frames from the test case scenarios *Walk1* and *Walk3* of the CAVIAR

collection, respectively. In *Walk1* collection, there is one human of interest that walks on a straight line. In *Walk3* the human walks on a B-Line following therefore, a highly different route. In figure 1c we depict the regions extracted from the two videos by providing the generated segmentation result of one of their frames. In order to avert the affection from the different colors of the two regions (clothes of the two humans) we manually edited the frames, altering the colors of the two humans so as to be the same for both of them. We primarily wanted to check whether our system would be able to disambiguate between these two collections, since the rest of the test cases of the collection comprise more than one interacting humans, or different backgrounds, thus these videos would be rejected as not depicting the same interactions (edges of the representation graph). The result was satisfactory. The system is able to disambiguate between the two videos and return the correct one as the first in the results list.

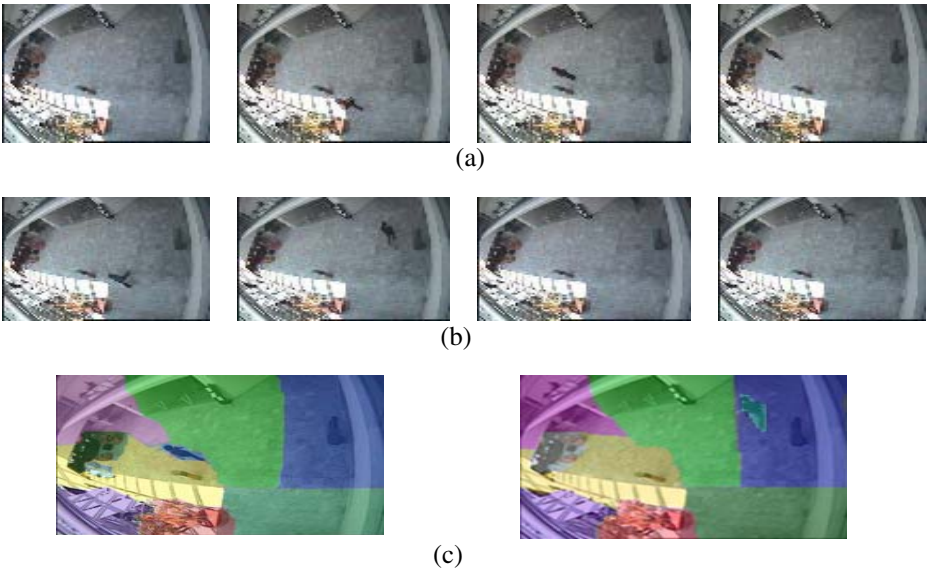


Fig. 1. (a) Characteristic frames from *Walk1* test case video. (b) Characteristic frames from *Walk1* test case video. (c) Segmentation Result on the two videos depicted on one of the frames they consist of.

Another test case aimed to check whether the proposed representation scheme can provide information about the duration of an event. For this purpose, we used the test sets *Browse_WhileWaiting1* and *Browse_WhileWaiting2* and we extended the proposed algorithm so as to apply a resemblance metric on the weights of the corresponding edges of the model and data graphs. This metric is the square of the time-slot's duration. The video-shots of these test cases are of the same background as the previous two and they depict a human browsing a specific region of the video-shot's background for different time durations. Therefore, the segmentation result is similar to the result depicted in figure (1c). The result of the retrieval procedure was

successful again. The system manages to disambiguate between the two videos, since the durations of the event (the human browsing) are of different duration, and returned the correct answer on the top of the results list and with a ranking score 36.1% bigger than the score of the second clip.

On average, we performed 60 tests using this test data collection. The test cases can be divided into the following categories: (a) retrieval of videos depicting interactions between blobs of desired features, (b) retrieval of videos depicting interactions between blobs of desired features and with a specific duration. The 87.2% of our tests were successful, since the correct result was returned at the top of the results list. In all the cases the correct (or "more fitting") result was among the three top results.

6 Conclusions

Video search within large data repositories is a growing research area. In this paper, we have described a novel uniform approach for video representation and space-time segmentation of video data. Unsupervised clustering, using a novel machine learning algorithm, enables the extraction of video segments, which are considered as coherent regions across the video sequence in space-time and are described by means of a spatio-temporal descriptor. An interesting differentiation from existing work in video is that space and time are treated uniformly, and the video is treated as a single entity as opposed to a sequence of separate frames.

The spatio-temporally coherent video regions are further analyzed on the scope of their spatio-temporal adjacencies and their spatio-temporal relations are deduced. A graph-based representation of the video sequence content is generated within which, the spatio-temporal descriptors of the video-regions are integrated and their spatio-temporal relations are depicted. This concise yet powerful video modeling scheme is the major contribution of this paper.

A great variety of retrieval algorithms could be applied to carry out retrieval procedures using the proposed video representation and summarization scheme. In this paper we model the retrieval problem as an inexact graph matching problem and we use a simple best match algorithm to address it. We have tested our system thoroughly using a set of videos taken by static cameras. In the 87.2% of our tests the correct answer to the user's query was on the top of the returned results, while in all our tests it was among the three higher ranked results. Conducted tests with videos where the cameras introduce extra motion in the scene and the backgrounds are not static have shown that our system's performance remains high when processing shots of narrow duration, affirming our theoretical assumptions.

Major future work goals are the introduction of an algorithm eliminating the camera motion effect and the refinement of the video segmentation algorithm, especially in terms of merging the on-line and off-line procedures into one single procedure, efficient enough to be able to provide information on real time.

Acknowledgements. This work has been funded by the POLYMNIA research and development project co-funded by the European Commission under the Sixth Framework Programme, Priority 2 "Information Society Technologies".

References

1. A. Hampapur, R. Jain: Video Data Management Systems: Metadata and Architecture. In: Multimedia Data Management, A. Sheth, W. Klas (eds.), McGraw-Hill (1998)
2. A. Yoshitaka, T. Ichikawa: A Survey on Content-Based Retrieval for Multimedia Databases. In: IEEE Transactions on Knowledge and Data Engineering, 11(1), pp. 81-93 (1999).
3. D. DeMenthon, D. Doermann: Surveillance: Video retrieval using spatio-temporal descriptors. In: Proceedings of the eleventh ACM international conference on Multimedia, ACM Press (2003)
4. E.H. Adelson and J.R. Bergen: Spatiotemporal Energy Models for the Perception of Motion. In: J. Opt. Soc. Am. A., vol. 2 pp. 284 – 299 (1985).
5. S. Gepshtein and M. Kubovy: The Emergence of Visual Objects in Space-Time. In: Proc. National Academy of Sciences, USA, vol. 97, pp. 8186 – 8199 (2000).
6. http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/.
7. Anastasios Doulamis, Klimis Ntalianis, Nikolaos Doulamis and Stefanos Kollias: An Efficient Fully-Unsupervised Video Object Segmentation Scheme Using an Adaptive Neural Network Classifier Architecture. In: *IEEE Trans. on Neural Networks*, pp. 616-630, Vol. 14, No. 3, May 2003.
8. A.S. Sato, K. Yamada, Generalized learning vector quantization: Advances in Neural Information Processing Systems. In: MIT Press, vol. 7, 1995, pp. 423–429.

Bridging the Syntactic and the Semantic Web Search

Georgios Kouzas[1], Ioannis Anagnostopoulos[2], Ilias Maglogiannis[2], and Christos Anagnostopoulos[3]

[1] School of Electrical and Computer Engineering, National Technical University of Athens, Heroon Polytechneiou 9, Zographou, 15773, Athens, Greece
gkouzas@ece.ntua.gr
[2] Department of Information and Communication Systems Engineering, University of the Aegean, Karlovassi 83200, Samos – Greece
{janag, imaglo@aegean.gr}
[3] Department of Cultural Technology and Communication, University of the Aegean, Mytiline 81100, Lesvos – Greece
canag@ct.aegean.gr

Abstract. This paper proposes an information system, which aims to bridge the semantic gap in web search. The system uses multiple domain ontological structures expanding the user's query with semantically related concepts, enhancing in parallel the quality of retrieval to a large extend. Query analyzers broaden the user's information needs from classical term-based to conceptually representations, using knowledge from relevant ontologies and theirs' properties. Besides the use of semantics, the system employs machine learning techniques from the field of swarm intelligence through the Ant Colony algorithm, where ants are considered as web agents capable of collecting and processing relevant information. Furthermore, the effectiveness of the approach is verified experimentally, by observing that the retrieval precision for the enhanced queries is in higher levels, in comparison with the results derived from the classical term-based retrieval procedure.

1 Introduction

A rapid growth of Internet activity can be observed in the last years, especially concerning web applications and information dissemination on many topics [1], [2]. The reason of web success is the fact that there were no special rules or limitations during web enlargement. However, this anarchic expansion of the web resulted to its chaotic structure. The initial design and implementation of the web was focused on information dissemination and accessibility from various sources (in every place in the world) for each interested final user. In other words, the web comprises a huge document collection that can be accessed by everyone. Nevertheless, web information collection consists of completely uncontrolled heterogeneous documents, thus, the search of specific information became difficult from its very first steps.

Automated search engines were developed to provide web users an easier way to get specific information. Various search engines were designed, such as World Wide Web Worm (WWWW), MSN, and the catalogue based Yahoo. The revolution in web search started with Google [3]. But even though the search engines manage to extract

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 104 – 112, 2006.

information for the end user, the web weaknesses, namely its chaotic structure and its lack of semantic context, disable the automated management and process of the web included information. In other words, the web is designed only for people and not for machines. Semantic web attempts to bridge this gap by converting the web structure from syntactic to semantic [4].

The Semantic Web is not a separate web but an extension of the current web, in which information is given a well-defined meaning, enabling computers and people to co-operate better. Specifically, the semantic web includes not only resources relevant to multimedia objects (web pages, images, documents etc) like the current web, but also resources relevant to other kind of objects like persons, organizations, facts and so on. Moreover, the semantic web includes relations between resources that are expressed with more than a simple hyperlink. The general idea is that all data in the semantic web are classified as a directed graph, where each node corresponds to a resource and each link between two nodes refers to a specific property type. The Resource Description Framework (RDF) [5] model is used for the representation of the resource graph. The basic model used in the RDF, consists of three basic object types: resources, properties and statements. Everything that can be described as an expression is defined as a resource. A unique Uniform Resource Identifier-URI is set to each resource. The properties of each resource, like relations to other resources or specific attributes, are described with the triple term "subject – predicate – object" which is called statement. The RDF Schema Specification Language (RDFS) [5] is used to expand the RDF model. In particular, it introduces a prototype ontology lexicon (class, property, Type, subClassOf, domain, range) and it defines ontology classes and class hierarchy. Moreover, the RDFS defines the class properties and relations. In other words, the RDFS constitute a system type for RDF statements.

Although, the semantic web was designed recently and it is still in an early phase, many applications can be based on this initial structure. In our approach, we attempt to introduce and exploit the abilities of the semantic web in the area of web search. Specifically, the proposed system enhances the classic term-based web search, by using semantics. Moreover, the meta-search described in [6], is used instead of a typical search engine, to increase the precision and the coverage of the results. The translation of the term query to an enhanced query with semantics is based on the results ranking of the meta-search engine and on the interaction with ontology web databases like Swoogle. A modification of the ant colony algorithm enhances the results of the proposed query through a real-time local search using the enhanced query with semantics and the results of the meta-search engine.

The paper is organized as follows: Section 2 describes our proposal along with all necessary algorithmic procedures and modules. In particular, the meta-search algorithm used for the initial result collection is presented [6], defining in parallel the transformation of the term based query to an enhanced query with semantics. Finally, the proposed modification over an Ant Colony Optimization (ACO) schema [7] is introduced (Ant Seeker) its functions are analytically explained. In section 3 the proposed system is applied to a specific case study and finally, in Section 4 the conclusions and future work are presented.

2 Description of the System

The proposed algorithmic procedure is based on the following concept. An information source (web page or site) should probably lead to another information source, with a similar content. A meta-search engine collects and ranks the results of more than one search engine in order to present the results in terms of relevance. Each web page that contains relevant information is set as a starting point. Ant Seeker algorithm is used to correlate the starting point with a destination point (another web page), linked in a close depth. In the beginning, starting points are defined as the initial query results derived from the meta-search engine. The algorithm is executed for each starting point. If a web page satisfies the enhanced query with semantics, it is defined as destination point. When a destination point is reached, it is defined as a starting point and the algorithm is repeated. The enhanced query with semantics is based on the initial user's query enhanced with terms of the same semantic meaning. The basic functions of the proposed system are illustrated in figure 1, in an attempt to group together similar information. The meta-search engine, the enhanced query with semantics and the ant seeker algorithm are described in the following.

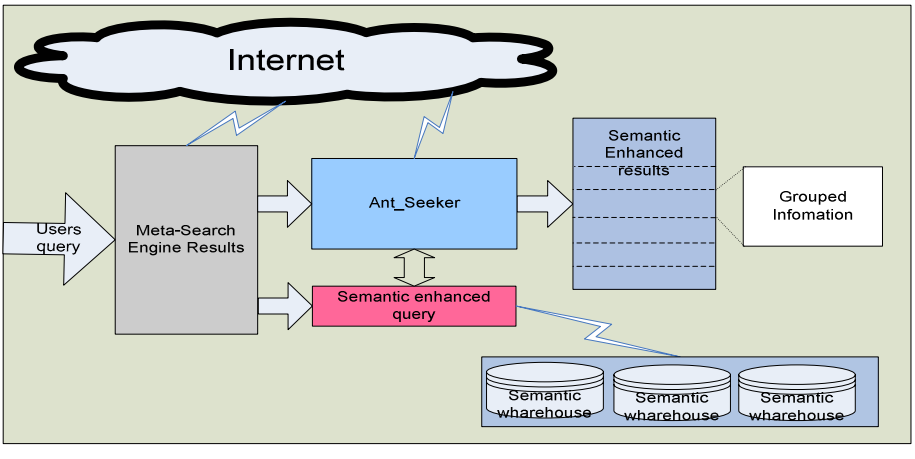


Fig. 1. The architecture of the proposed system

2.1 The Meta-search Algorithm

A meta-search engine is chosen for the query results instead of a typical search engine, like Google, because the meta-search engine utilizes more than one known search engine and the user gets an enhanced amount of information, recording in parallel his search preferences. The meta-search engine chosen for our approach is a user-defined meta search engine (UMSE) and it is described in [6]. UMSE uses a rank-based isolated merging method, since it uses information, which is readily available from search servers, without requiring any other server functionality [8],[9]. In other words the proposed method employs server-assigned ordinal ranks in order to generate the merged list of the meta-results. The UMSE 'extracts' the required

information from all the submitted services combined with the meta-results and the user profile information. Then the duplicate information sources are removed. The problem of UMSE is addressed to have a search engine ranking $S = \langle R, r \rangle$, consisted of a set R of results and an ordering r. Given N ranking from N different search engines, the anticipated outcome is the generation of a single ranking $S_m = \langle R_m, r_m \rangle$, such that $R_m = R_1 \bigcup \ldots \bigcup R_N$ and r_m is the derived meta-results ranking. In other words, the merging algorithm compares whether the information source retrieved in the r^{th} rank position of search engine with priority p, exists until the $(r-1)^{th}$ rank position of the other selected search engines. The duplicate fields in the above sequence are eliminated while the procedure ends with the assignment of the last meta-result. The number of the meta-results is the total returned results from all the involved search engines, having removed the duplicated fields. UMSE allows the user to adjust the number of the returned results from each used search service. This number has a large impact on the total number and the presentation time of the meta-results.

2.2 Bridging the Semantic Gap

After the extraction of the meta-results we want to correlate the query terms with terms, which define the respective ontologies. Even though this seems expected, submitting multiple different queries is a quite time-consuming procedure for the user. Thus, in this step we propose the use of Swoogle, which is a semantic web search and meta-data engine [10]. This search engine reveals and analyses ontologies in semantic web files extracting meta-data. In its current form it uses the Google search engine and the Jena2 [11] parser in order to find and evaluate a large amount of files (semantic web documents - SWDs), which have relevant extensions for the semantic web (*.rdf, *.owl, etc.).

However, in our approach we use Swoogle in order to find relevant data, which identify an ontology or a relation between ontologies, in respect to a query term *q*. In particular, we search in the cached triples (subject, predicate, object) and we extract the *object* values *q′* from all the triples where the *predicate* value is *rdfs:subClassOf* and the value of the *hasLocalname* Swoogle metadata is the term *q*. An example is presented in the following section case study.

2.3 The Ant Seeker Algorithm

The basic concept of ant colony algorithms was inspired by the observation of swarm colonies, specifically ants [12]. Since most species of ants are blind, they deposit a chemical substance called pheromone to find their way to the food source and back to their colony [13], [14]. The pheromone evaporates over time. It has been shown experimentally that the pheromone trail leads to the detection of shortest paths [15]. For example, a set of ants, initially, create a path to the food source. An obstacle with two ends is placed in their way, with one end more distant than the other. In the beginning, equal numbers of ants spread around the two ends of the obstacle. The ants, which choose the path of the nearer end of the obstacle, return before the others. The pheromone deposited to the shortest path increases more rapidly than the pheromone

deposited to the farther one. Finally, as more ants use the shortest path, the pheromone of the longest path evaporates and the path disappears. In artificial life, the Ant Colony Optimization (ACO) uses artificial ants, called agents, to find solutions to difficult combinatorial optimization problems [7], [16]. ACO algorithms are based on the following concept. Each path followed by an ant is associated with a candidate solution to a given problem. The amount of pheromone deposited on a path followed by an ant is proportional to the quality of the corresponding candidate solution for the target problem. Finally, when an ant has to choose between two or more paths, those with the larger amount of pheromone have a greater probability of being chosen by the ant.

```
Initialize system
Define Starting points: Start_List = [UMSE_Results]
Destination_List = []
Total num of Ants = NoAnts
Total number of iterations for algorithm = NoIterations
Initial pheromone value for each node added in search  = IPV
Maximum number of nodes should visit each ant = N_max
For i=0 to NoIterations
        For j=0 to NoAnts
                Init_ant
                Repeat
                        Select_Next_Node(j)
                        Query_Visited_Node(j,  q′ )
                        visited nodes(j)++
                Until ((visited nodes(j) = N_max) or Query_Visited_Node(j,  q′ )=True)
                Calculate route(j)
                Set Visited_node to Destination_List
        End for
        Short Destination_List
        Update pheromone
End for
Set Start_List =Destination_List
```

Fig. 2. Ant Seeker algorithm's pseudo-code

In our approach, we propose a modification of the ACO algorithm [7], which we call Ant_Seeker. In this algorithm each artificial ant employs the following properties:

- Each ant is capable of carrying memory (pheromone based)
- The node selection is based on pheromone level deposited in each node.
- Each ant has a maximum number of nodes that it can visit before discovering a destination node.
- All ants start from a starting node
- Each ant uses the enhanced query (q') described above to identify the nodes

The following paragraph describes how the ant seeker algorithm is applied to the web search. Figure 2 illustrates the pseudo-code of the ant seeker algorithm. In order to initialize our model we introduce the following parameters:

- The parameter NoA establishes the number of ants.
- An initial pheromone value equal to IPV, is set in every new linked page which is introduced in our search area
- Each ant can visit a maximum number of nodes N_{max}

Let's suppose that a starting node is given by a meta-search engine. All ants are initially set to the starting point. Each time, every ant must move from a node *i* to node *j* which should be directly linked to the node *i*. The directly movement between node *i* and *j* is called accessibility and described by h_{ij} parameter. If node *j* is directly linked to node *i*, the parameter h_{ij} is set to 1 otherwise is set to zero. Let $\tau_i(t)$ be the pheromone amount on node *i* at time *t*. Each ant at time *t* chooses the next node until visit a maximum number N_{max} of nodes. Therefore, we call an iteration of the Ant_Seeker algorithm the completion of route for each ant. At this point the pheromone is updated according to Equation 1, where ρ is a coefficient such that $(1 - \rho)$ represents the evaporation of trail between time *t* and *t+1*, while $\Delta\tau_i$ is given according to Equation 2. In Equation 2, $\Delta\tau_i^k$ is the quantity per unit of level of pheromone is laid on node *i* by the k_{th} ant between time *t* and *t+1* and is expressed by Equation 3.

$$\tau_i(t+1) = \rho \cdot \tau_i(t) + \Delta\tau_i \tag{1}$$

$$\Delta\tau_i = \sum_{k=1}^{m} \Delta\tau_i^k \tag{2}$$

$$\Delta\tau_i^k = \begin{cases} \dfrac{Q}{N^k} & \text{if } k \text{ ant visits node } i \text{ in its tour} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

In Equation 4, Q is a constant and N^k is the number of visited nodes for ant *k*. The coefficient ρ must be set to a value lower than 1 for avoiding unlimited accumulation of trail pheromone. An initial pheromone value equal to IPV is set in every new node is added to the search area. In order to satisfy the constraint that an ant doesn't visit a visited node, each ant is associated with a data structure called the *vlist*, that saves the nodes already visited and forbids the ant to visit them again before a tour have been completed. When a tour is completed, the *vlist* is used to extract the nodes are satisfying the enhanced query q'. The *vlist* is then emptied and the ant is free to choose again.

The transition probability from node i to node j for the k^{th} ant is defined at Equation 4, where allowedk = {Nodes can be visited - vlist}. Therefore the transition probability is a trade-off between accessibility (which states that only directly linked nodes should be chosen) and pheromone level at time t (which states that if this node was previously selected then this node is highly desirable, thus implementing the autocatalytic process).

$$P_{ij} = \frac{\tau_j \cdot h_{ij}}{\sum_{k \in allowed_k} \tau_k \cdot h_{kj}} \tag{4}$$

Where h_{ij} is the accessibility of node j from node i and is given by Equation 5.

$$h_{ij} = \begin{cases} 1 & \text{if j node is directly linked from node i} \\ 0 & otherwise \end{cases} \tag{5}$$

As mentioned above, each ant has a specific number of nodes that can visit equal to N_{max}. This number defines the depth search of each ant. When an ant discovers a node which satisfies the enhanced query q' the search stops and the node is set as destination node.

3 Case Study

In this section we present an example of how we bridge the semantic gap between the syntactic and the semantic web by using the syntactic meta-search engine UMSE and the semantic web search engine Swoogle.

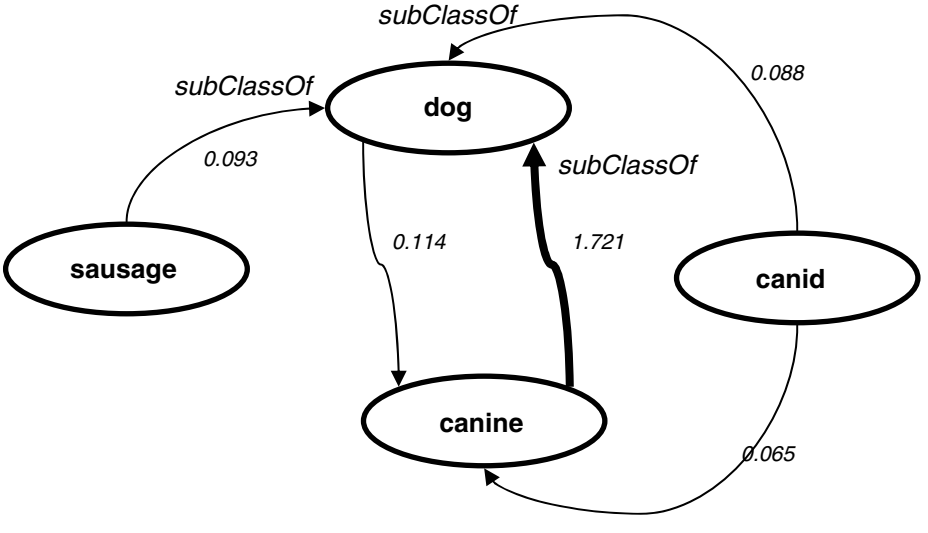


Fig. 3. Relation between query terms and their ontologies (property: subClassOf)

Let us suppose that a user wants to get some information in respect to the term '*dog*' (domestic animal). This term reflects to the meta description "a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times" as it is stated as literal value in the *object* field where the *predicate* value is *rdfs:description* while the value of the *hasLocalname* Swoogle

metadata is the term *'dog'*. However, there are three different semantic approaches where this term corresponds to three different ontologies, namely *Sausage, Canine* and *canid.* Table 1 presents the results extracted by UMSE using Google, Yahoo and MSN as search engine services (top 100 results per search engine), in respect to the four query terms (dog, canine, canid and sausage). The third column presents the relevant results in respect to the amount of the returned meta-results of the second column, while the fourth column presents the precision value over the recall level of the returned meta-results.

By finding the frequency of co-occurrences between these four terms (e.g. frequency of appearance of one term in respect to the meta-results of the others) we derived in figure 3, which illustrates the weighted relations between these terms. Then, a weight threshold value has been arbitrary chosen (0.8 in this case) in order to enhance our query term set. Only term '*canine*' satisfied that threshold being in parallel a defined sub-class for the term. Then, we tested the accuracy of the enhanced Boolean queries 'dog AND canine' as well as 'dog OR canine', which were measured at higher levels compared to the initial query (86.5% and 77.5% respectively).

Table 1. Extracted results, relevancy and accuracy levels (per tested queries)

Query term(s)	UMSE meta-results	Relevant results	Precision
Dog	202	141	69.8%
Canine	213	155	72.7%
Canid	173	24	13.8%
Sausage	164	7	4.3%
dog AND canine	193	167	86.5%
dog OR canine	222	172	77.5%

4 Conclusions

In this paper, a new approach of web search is introduced. Specifically, our aim is the conjunction of the classic term based web search and the semantic web. Semantic definitions like RDF schema and OWL language provide a powerful framework enabling computers and people to co-operate better. The enhanced search can be improved by using the well known optimization algorithms such as ant colony algorithms. The final assessment of the proposed method will be evaluated on a large set of web pages.

Despite the fact that the proposed system is in evaluation phase, the results depict an improvement of precision in respect to the initial query. However, the proposed system should be evaluated in a larger scale. Some parameters like the weight threshold value for the enhancement of the query term set should be better defined. Another crucial point is the Ant_seeker definitions. The optimization of parameters like search depth or pheromone update function would possible improves the precision and the functionality of our system.

References

1. I. Anagnostopoulos, C. Anagnostopoulos, G. Kouzas and D. Vergados, “A Generalised Regression algorithm for web page categorisation”, Neural Computing & Applications journal, Springer-Verlag, Vol. 13, no. 3, pp. 229 – 236, 2004.
2. I. Anagnostopoulos, C. Anagnostopoulos, Vassili Loumos, Eleftherios Kayafas, “Classifying Web Pages employing a Probabilistic Neural Network Classifier”, IEE Proceedings – Software, vol. 151, no. 03, pp. 139-150, March 2004.
3. S. Brin and L. Page. “The anatomy of a large-scale hypertextual web search engine”. In WWW7: Proceedings of the seventh international conference on World Wide Web 7. Elsevier Science Publishers B. V., 1998.
4. Landauer, T. K., Foltz, P. W., & Laham, D. (1998). Introduction to Latent Semantic Analysis.Discourse Processes, 25, 259-284.
5. D. Brickley and R.V.Guha. Rdf schema. http://www.w3.org/TR/rdf-schema/.
6. Anagnostopoulos I., Psoroulas I., Loumos V. and Kayafas E., “Implementing a customized meta-search interface for user query personalization”, , 24th International Conference on In-formation Technology Interfaces, ITI 2002, pp. 79-84, June 24-27, 2002, Cavtat/Dubrovnik, CROATIA.
7. Dorigo M., and Maniezzo V., 1996, “The ant system: optimization by a colony of cooperating agents”. IEEE Transactions on Systems, Man and Cybernetics, 26(1), 1-13.
8. Craswell, Nick, Hawking, David and Thistlewaite, Paul. Merging Results from Isolated Search Engines. 10th Australasian Database Conference, Auckland, New Zealand, January 1999, Springer-Verlag, Singapore.
9. Yuwono, Budi and Lee, Dik L. Server ranking for distributed text retrieval systems on the internet. In Topor, Rodney and Tanaka, Katsumi, editors, DASFAA '97, pages 41-49, Melbourne. World Scientific, Singapore.
10. L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. C. Doshi, and J. Sachs. Swoogle: A search and metadata engine for the semantic web. In in Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management, Washington, DC, Nov. 2004.
11. Jena Semantic Web Framework, http://jena.sourceforge.net
12. Bonabeau E., Dorigo M., & Theraulaz G. “Intelligence: From Natural to Artificial Systems”, 1999, Oxford University Press.
13. Dorigo M. and Caro G.D., 1999, “The Ant Colony Optimization Meta-heuristic,” in New Ideas in Optimization, D. Corne, M. Dorigo, and F. Glover, Eds. London: McGraw-Hill, pp. 11-32
14. Dorigo M., and Caro G.D., 1999, “Ant Algorithms Optimization. Artificial Life”, 5(3), 137-172.
15. Chen S., Smith. S., 1996, ”Commonality and genetic algorithms”. Technical Report CMU-RITR-96-27, The Robotic Institute, Carnegie Mellon University, Pittsburgh, PA, USA.
16. Bianchi, L., Gambardella L.M., Dorigo M., 2002, „An ant colony optimization approach to the probabilistic travelling salesman problem”. In Proceedings of PPSN-VII, Seventh Inter17 national Conference on Parallel Problem Solving from Nature, Lecture Notes in Computer Science. Springer Verlag, Berlin, Germany.

Content-Based Coin Retrieval Using Invariant Features and Self-organizing Maps

Nikolaos Vassilas and Christos Skourlas

Department of Informatics, Technological Educational Institute of Athens
{nvas, cskourlas}@teiath.gr

Abstract. During the last years, Content-Based Image Retrieval (CBIR) has developed to an important research domain within the context of multimodal information retrieval. In the coin retrieval application dealt in this paper, the goal is to retrieve images of coins that are similar to a query coin based on features extracted from color or grayscale images. To assure improved performance at various scales, orientations or in the presence of noise, a set of global and local invariant features is proposed. Experimental results using a Euro coin database show that color moments as well as edge gradient shape features, computed at five concentric equal-area rings, compare favorably to wavelet features. Moreover, combinations of the above features using L1 or L2 similarity measures lead to excellent retrieval capabilities. Finally, color quantization of the database images using self-organizing maps not only leads to memory savings but also it is shown to even improve retrieval accuracy.

1 Introduction

With the introduction of the World Wide Web, the digital cameras and the large – and cheap – memory capacities of modern computers, multimedia databases and large image collections can now be found not only in various organizations of the public or private sector but also in many home PCs. Filing and indexing of such content with traditional manual image annotation and keyword-based techniques is a tedious and, in some cases, almost impossible work, considering that some image collections may contain hundreds of thousands or even millions of images. Moreover, the difficulty to annotate images so that to allow later retrieval that is acceptable by the subjective perception of the various future users makes text-based image retrieval an inappropriate approach [1]. It is, therefore, necessary to develop tools for retrieving information based on image content.

In the recent years, Content-Based Image Retrieval (CBIR) evolved to an important research domain within the context of multimodal information retrieval [2] and a number of CBIR systems and tools have already been developed [1,3,4]. CBIR could then be used to assist the document matching stage in complex multimodal information retrieval applications, such as cross language document retrieval [5], when the documents contain images. In such applications, the documents stored in a cross language documents collection are typically represented using the vector space model. A set of N keywords (index terms) is then used to represent each document as an N-dimensional vector with each element representing either the appearance of the

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 113 – 122, 2006.

corresponding keyword in the document or its relative frequency of occurrence. To correctly translate a keyword from one language to another, besides the use of a dictionary and a thesaurus (for the synonyms), word sense disambiguation techniques that assess the appropriate contextual meaning of the word have to be employed [5]. The similarity between a submitted query and each document in the collection, is usually an inner product based vector matching operation. However, due to a poor selection of keywords, keyword sense ambiguities, inaccurate lexicons or incomplete thesauri, the retrieval accuracy is rather low. In the case that the documents also contain images, redesigning the similarity measures to include the contribution from CBIR systems could lead to significant improvements in document retrieval accuracy.

Due to the inherent difficulties of dealing with any kind of image content and in order to improve retrieval accuracy, most CBIR research results reported in the literature have been obtained either for small image collections or for thematic image databases. Such is the case, in this paper, with the proposed CBIR system for coin identification. As a preliminary stage towards the development of a robust modern and ancient coin identification system, this paper aims at the design of a translation-, scale- and orientation-invariant coin identification system with improved retrieval capabilities under the presence of additive noise and changes in illumination conditions. Unlike commercial coin recognizers, such as those found in automatic coin classifiers of vending machines, which extract features that correspond to the physical properties of the coins [6], all features in the proposed CBIR system are extracted exclusively from the coin images themselves.

Previous work on automatic coin recognition through the use of a neural pattern recognition system, used the sum of gray level values within 37 ring segments of the coins to achieve rotation invariant recognition and was demonstrated in the case of four coin faces [7]. Translation and scale invariance was not dealt with since the coin images were obtained at a constant size and position by an automatic coin classifying machine. In a genetic programming application [8], concentric circular pixel statistics are shown to be more effective than square features for coin detection problems. In [9], following edge detection and numeral subimage extraction from three coins, a rotation invariant recognition is achieved using Gabor filters and a neural classifier. In [10], the coin recognition system was based on simple texture features and probability histograms, did not provide for rotation invariance and was demonstrated for four textural coin faces. Finally, [11] presents a hierarchical coin classification system that combines features from the physical properties of coins with intensity and edge eigenspaces extracted from the coin images. This system achieves rotation invariance by cross-correlation between the polar representations of the query and the hierarchically selected database coins and is demonstrated to have a good classification accuracy on a large coin database.

Section 2 of this paper presents the experimental methodology for the original coin collection as well as for the coin collection after vector quantization with Kohonen's self-organizing maps. In Section 3, the features extracted for content-based coin indexing are presented. The experimental results for three test datasets are presented in Section 4. Finally, conclusions and future work are discussed in the last section.

2 Experimental Methodology

As it is typical in database design applications, the experimental methodology consists of two distinct phases: a) the design phase, whereby we create the database and decide upon its indexing scheme, and b) the retrieval phase, in which, following the presentation of query-images of coins, similar images are retrieved from the database. Moreover, in order to examine the effect of indexed image representation through vector quantization on the quality of the whole retrieval process, we used Kohonen's self-organizing maps algorithm on the RGB coin images. The most evident benefit of an indexed image representation with a relatively small colormap is the memory savings especially for large image collections.

2.1 Database Design

In the first phase, a coin database is created using the following stages:

- Specification of a digital coin image collection
- Preprocessing of each image to detect the position of the coin
- Feature extraction from coin pixels
- Feature-based indexing of the database

At the first stage, we downloaded coin images from the Internet and compiled a collection of 115 Euro coin faces. Among these images, several coin designs, not yet in circulation, were included. All images were in the RGB color space and had a size of approximately 240x240 pixels. Fig. 1 shows a coin sample from the "original" collection.

Fig. 1. A sample from the coin collection

Next, each image is preprocessed in order to detect the position of the coin in the image and isolate it from the background. To this end, each image is first converted to gray-scale, then the edges are found using the Sobel operators and finally the Hough transform [12] is applied on the black-and-white image of edges in order to detect the outer circle of the coin. The circular disk mask, thus created, is then used to isolate the coin from the background (see Fig. 2). Due to the computational complexity of the

Hough transform and the memory demand of the accumulator array, when the image sizes are large, one could first sub-sample the gray-scale image to reduce its size, then apply the Sobel operators and Hough transform to the smaller image and, finally, upscale the so found circle parameters. Alternatively, other methods, such as those based on image thresholding [13], can be used to determine the circular mask.

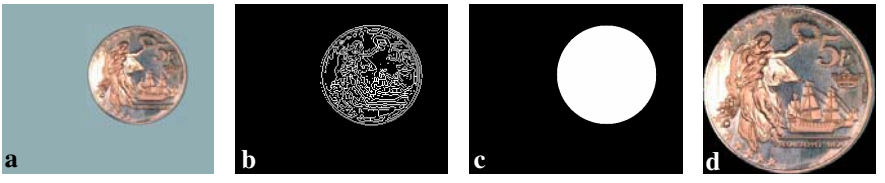


Fig. 2. Preprocessing steps: a) original color image, b) Sobel edge detection on gray-scale image, c) mask from Hough transform, and d) isolated coin

Once the coin images are free from background interference, features are extracted from those pixels that belong exclusively to the coin area. A set of color and shape (edge magnitudes and orientations) features, invariant to translation, scale and orientation, along with features extracted from the wavelet transform are presented in Section 3. The extracted feature vector from each image is then stored in memory in order to form the index to the database.

2.2 Database Retrieval

Retrieval from the coin database is performed by:

- Query-image presentation to the database system
- Preprocessing to detect the position of the coin in the query-image
- Feature extraction and query encoding with its feature vector
- Specification of similarity measure
- Computation of query features similarity with those of the database index
- Retrieval of the most similar database coins

The query coin image presented to the system undergoes the same preprocessing, as that for the database coins, to isolate the coin from the background and to extract its corresponding feature vector. The similarity measures employed in this work between two feature vectors are the L_1 and L_2 distances defined by

$$L_p = \left(\sum_i | f_i(I) - f_i(J) |^p \right)^{1/p} \qquad \text{for} \quad p = 1, 2 \tag{1}$$

where $f(\cdot)$ represents the feature vector of an image and I, J are the database and query images respectively. In fact, the similarity measures that correspond to the groups of color (S_c), edge magnitude (S_m), edge orientation (S_o) and wavelet (S_w) features are combined through a linearly weighted function to produce the overall similarity:

$$S = w_1 S_c + w_2 S_m + w_3 S_o + w_4 S_w \tag{2}$$

with w_i been the weights of the linear combination determined by the user. Hence, the user can tailor the similarity measure according to the needs of the particular application by specifying the weight values. Finally, the database coins are ranked according to their similarities to the query image and the *N* most similar coins are retrieved from the database and shown to the user.

2.3 Quantization of Coin Collection

One of the goals of this work is to examine possibilities of memory savings, especially useful when the image collections are large, using an indexed image representation. Through vector-quantization, an RGB image will be represented with an index table and a colormap. The smaller the colormap the bigger the memory savings since fewer bits per index are needed. Additional advantages of quantizing the image database are found in [14, 15]. The restriction, of course, is to not significantly compromise the overall system's retrieval performance.

Kohonen's self-organizing maps [16] was the vector quantization method used in this work. The coin collection was quantized in two ways. First, Kohonen's algorithm was applied to each coin image separately resulting to indexed images with distinct

Fig. 3. Original (up) and quantized (bottom) coins using a common colormap (right)

colormaps. Second, the same algorithm was applied to the whole coin collection simultaneously, resulting to indexed images with a common colormap. The size of the map used in both cases was of 16x16 neurons and, hence, all images were represented using 256 colors, with one byte associated to each index to the colormap. Fig. 3 shows some original coins (upper row) along with their indexed representations (lower row) for the second quantization case. Also shown to the right of this figure is the coins' common colormap.

The feature extraction and indexing phases for the database coins are the same as for the previously described methodology. During retrieval, the queries are also quantized with Kohonen's algorithm resulting in an indexed representation with their own associated colormap. The remaining processing steps are the same as before.

3 Feature Extraction

In order to allow for robust retrieval, the extracted features should be invariant to translation, scale and rotation and, to some degree, to changes in illumination conditions

and to the presence of noise. However, translation invariance is achieved at the pre-processing stage through the Hough transform detection of the coin's position and is of no concern in the sequel.

Three different kind of features are used in this work, namely, color, shape and wavelet features. Color is known to be invariant to scale and orientation but is sensitive to illumination changes. Four moments (mean, standard deviation, skewness and kyrtosis) from each of the hue, saturation and value components of the HSV color space were extracted, for a total of twelve color features.

The shape features were extracted from the gradient image which was obtained by first transforming the color image to gray-scale and then using the Sobel operators. To achieve some invariance to illumination conditions, the [min, max] range of the gradient image was normalized to [0, 1]. Assuming that scale and orientation changes preserve most of the edges, the normalized polar histogram, i.e. the probability distribution of edge orientations (at 90^o with respect to the edge gradients), not only should it be, adequately, scale and rotation invariant but also can give an estimate of the rotation angle. The latter results from the normalized polar histogram circular correlation between the query image and each one of the database coins. Fig. 4 shows the normalized polar histograms of a database image and a query that is of a different size and orientation with respect to the original image.

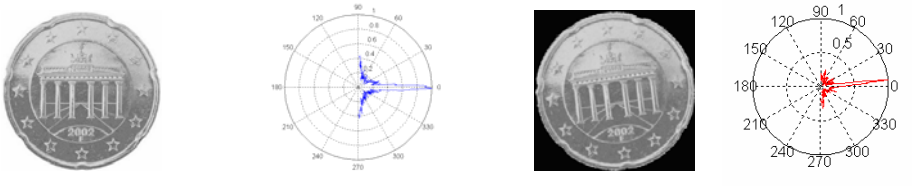


Fig. 4. A database (left) and a query (right) coin along with their polar histograms

Actually, the procedure followed in this work uses the L_1 or L_2 distance measures instead of the typical inner product correlation in order to determine the angle of rotation and the corresponding polar similarity score. In addition, in order to increase the robustness of the system, we extracted both, edge magnitudes and orientations from five equal-area concentric circular rings. The shape features extracted were: a) the mean, standard deviation, skewness and kyrtosis of the ring edge magnitudes, for a total of 5x4 = 20 features, and b) the normalized polar histograms of each ring for 5 degrees angular bins (36 features/ring), for a total of 180 polar features.

Finally, for comparison purposes, we also extracted features from a three level wavelet analysis with the 'db3' mother wavelet of the Daubechies family. In particular, we extracted the mean and standard deviation of the wavelet coefficients for each detail image of the wavelet decomposition as well as for the level-3 approximation image, for a total of 10x2 = 20 wavelet features.

4 Experimental Results

The experiments on coin retrieval have been performed using three sets of queries. The first data set (DS1) contains 123 queries at a lower resolution (approximately of 130x130 pixels) in order to test the effect of scale on the retrieval accuracy. The second data set (DS2) consists of 95 queries at a different scale (70% the size of the original images), random rotations in the [-180^o , 180^o] range and corrupted with additive zero mean Gaussian noise with 0.1 standard deviation. Finally, the third data set (DS3) consists of 115 queries at random scales (between 50% and 100% of the original scale), random orientations (in the [-180^o , 180^o] range) and with a random change of color in order to test scale, rotation and illumination invariance.

After each presentation of a query coin, the similarity scores were computed and sorted in decreasing order. Then the position of the correct database coin was recorded and the retrieval accuracy was measured as the average correct image position with respect to all queries from a particular data set. To give a better picture of the retrieval process, the standard deviation of the correct coin positions is also included in the results. Because in all experiments the mean (μ) and standard deviation (σ) were, as expected, more reliable features than skewness and kyrtosis, the latter features are not used in the following experiments. Also, since the L_1 distance measure gave slightly better results than L_2 in most of the experiments, all results shown in the tables assume the L_1 similarity measure.

Tables 1 and 2 show the retrieval accuracy based on color and edge (ring) magnitude features respectively, for the three data sets and the original coin collection. The feature vectors consisted of the means, standard deviations or their combination. The

Table 1. Retrieval based on color

	Feature vector composed of:		
	μ	σ	$\mu+\sigma$
DS1	4.6 (7.0)	5.7 (5.1)	3.3 (4.9)
DS2	6.5 (6.6)	25.6 (29.7)	7.2 (7.8)
DS3	19.1 (19.4)	30.7 (27.3)	18.3 (22.2)

Table 2. Retrieval using edge strengths

	Feature vector composed of:		
	μ	σ	$\mu+\sigma$
DS1	10.1 (14.6)	23.9 (28.9)	13.8 (20.7)
DS2	17.6 (19.6)	35.2 (30.6)	22.4 (23.0)
DS3	32.3 (28.8)	35.6 (30.8)	30.9 (29.4)

Table 3. Retrieval using edge angles

	Feature vector:
	Normalized Polar Histogram
DS1	12.1 (17.1)
DS2	22.7 (27.6)
DS3	29.4 (31.9)

Table 4. Retrieval using wavelet features

	Feature vector composed of:		
	μ	σ	$\mu+\sigma$
DS1	36.3 (28.2)	32.7 (26.7)	36.7 (28.9)
DS2	34.1 (27.9)	38.8 (30.3)	32.8 (28.2)
DS3	36.3 (29.3)	31.7 (27.0)	31.3 (27.9)

corresponding retrieval accuracy for the edge (ring) orientations and wavelet features, is shown in Tables 3 and 4 respectively. As it is evident from these tables, the best results were obtained for DS1 and, in particular, for the color features since a change of scale does not significantly affect the HSV color histograms. The presence of noise on the queries of DS2 had a less severe effect on the retrieval accuracy than the change in color on the queries of DS3. Regarding the color features, this is explained

by the fact that color mean is not significantly affected by the zero mean noise as by a change in color. However, the color standard deviation is not a reliable feature in the presence of noise or color change. The retrieval accuracy is more affected in the case of edge features since the edges are sensitive to noise and color changes. Finally, although wavelet features are widely used in texture recognition, they do not perform well in the case of textureless retrieval applications. The average coin position seems to not be affected by the presence of noise or color changes on the query images. However, they can improve retrieval accuracy when combined with other features.

Better retrieval results can be obtained by combining the above features through the weighted similarity measure of Section 2.2. Table 5 shows retrieval accuracy for five feature combinations with S_c , S_m and S_w been computed on a combination of mean and standard deviation features. For queries from DS1, by combining color, edge strength and polar features, the correct coin is retrieved (on the average) in the second position, gaining 2.5 positions with respect to the color features alone. Similarly, for queries from DS2 and DS3, the combined color, edge strength and wavelet features, give an improved accuracy of 6 and 14.6 respectively. The latter shows the usefulness of the wavelet features in weighted similarity measures.

Table 5. Retrieval accuracy from original coin collection using combined features

	Similarity measure:				
	$S_c + S_m + S_o$	$S_c + S_m$	$S_c + S_m + S_w$	$S_c + S_o$	$S_c + S_m + S_o + S_w$
DS1	**2.1** (2.2)	2.3 (2.5)	2.9 (3.7)	2.4 (3.1)	2.8 (3.3)
DS2	7.8 (12.4)	6.3 (8.6)	**6.0** (7.1)	6.0 (8.6)	10.1 (15.0)
DS3	15.4 (20.6)	16.7 (21.1)	**14.6** (18.1)	15.4 (20.3)	17.3 (24.8)

Finally, Tables 6 and 7 show retrieval results for five features combinations when the coin collection is quantized, using Kohonen's algorithm, with independent or one common colormap, respectively. For queries from DS1, the effect of either quantization technique on retrieval accuracy is negligible. However, for queries from DS2 or DS3, there is a significant improvement, especially when all coins are quantized with

Table 6. Retrieval accuracy from independently quantized coin images

	Similarity measure:				
	$S_c + S_m + S_o$	$S_c + S_m$	$S_c + S_m + S_w$	$S_c + S_o$	$S_c + S_m + S_o + S_w$
DS1	**2.1** (2.2)	2.3 (2.5)	2.9 (3.7)	2.4 (3.1)	2.8 (3.3)
DS2	7.8 (12.4)	6.3 (8.6)	**6.0** (7.1)	6.0 (8.6)	10.1 (15.0)
DS3	15.4 (20.6)	16.7 (21.1)	**14.6** (18.1)	15.4 (20.3)	17.3 (24.8)

the same colormap. In the best case, for DS2 queries, the correct coin appears on the average within the first 5 retrieved coins while for DS3 queries it appears within the first 3 retrieved coins, a position comparable to that of the simply scaled down images of DS1. Fig. 5 shows two queries from DS2 and DS3 respectively, along with the first six retrieved coins. In the first case the correct database coin is in the fourth position and in the second case it is in the first position.

Table 7. Retrieval accuracy from quantized coin images with same colormap

	Similarity measure:				
	$S_c + S_m + S_o$	$S_c + S_m$	$S_c + S_m + S_w$	$S_c + S_o$	$S_c + S_m + S_o + S_w$
DS1	**2.2** (2.4)	2.4 (2.2)	3.6 (5.4)	2.5 (3.4)	3.3 (4.5)
DS2	6.1 (9.3)	**4.8** (6.7)	6.4 (11.5)	5.4 (7.6)	9.5 (16.1)
DS3	**2.6** (3.3)	6.6 (11.6)	6.3 (11.4)	3.1 (3.8)	2.8 (5.0)

Fig. 5. Retrieval of 6 most similar coins with single-colormap quantized database for one DS2 query (upper row) and one DS3 query (lower row)

5 Conclusions

In this work, we developed a content-based coin retrieval system using color, shape and wavelet features with consideration to translation, scale and rotation invariance. The original coin collection consisted of 115 Euro coins and was used during the design phase to create the index to the database. In the retrieval phase, a weighted similarity measure was used to match the query's feature vector to those of the index and retrieve the most similar database coins. Several experiments have been performed showing improved retrieval accuracy when using feature combinations. Moreover, contrary to intuition, vector quantization of the coin database using self-organizing maps, showed a significant improvement of system's performance when the scaled-down and rotated queries were corrupted by additive noise or had color alterations, especially when a common colormap was used.

Our future work will focus in the use of CBIR to assist the document matching stage in cross language document retrieval applications when the documents contain images. As a pilot document collection, we intend to use coin images along with the corresponding coin descriptions and/or related texts (contexts or small sentences) in any of two languages. The experiments will be designed to evaluate various similarity measures that combine keyword-based with content-based matching under particular keywords' selection and word sense disambiguation techniques.

Acknowledgments

This Project is co-funded by the European Social Fund and National Resources – (EPEAEK-II)-ARXIMHDHS.

References

1. Rui, Y., Huang, T.S., Chang, S.-F.: Image Retrieval: Current Techniques, Promising Directions and Open Issues. J. of Visual Communication and Image Representation, Vol. 10 (1999) 1-23
2. Faloutsos, C., Oard, D.: A Survey of Information Retrieval and Filtering Methods. Tech. Rep. CS-TR-3514, Dept. of Computer Science, Univ. of Maryland, College Park, (1995)
3. Veltkamp, R.C.: Content-Based Image Retrieval Systems: A Survey. Revision of Tech. Rep. UU-CS-2000-34, Dept. of Computer Science, Utrecht University, (2002)
4. Eakins, J.P., Graham, M.E.: Content-Based Image Retrieval. Tech. Rep. JTAP-039, JISC Technology Application Program, Newcastle upon Tyne, (2000)
5. Marinagi, C., Alevizos, T.,Kaburlasos, V.G., Skourlas, C.: Fuzzy Interval Number (FIN) Techniques for Cross Language Information Retrieval. 8th ICEIS, May 2006 (accepted for publication)
6. Moreno, J.M., Madrenas, J., Cabestany, J., Launa, J.R.: Practical Design Methodology for Commercial Automatic Coin Recognizers based on Neural Decision Engines. Proc. Int. Conf. Neural Information Processing and Intelligent Information Systems (1997) 662-665
7. Fukumi, M., Omatu, S., Takeda, F., Kosaka, T.: Rotation-Invariant Neural Pattern Recognition System with Application to Coin Recognition. IEEE Tr. on Neural Networks, Vol. 3, No. 2 (1992) 272-279
8. Zhang, M., Bhowan, U.: Program Size and Pixel Statistics in Genetic Programming for Object Detection. Proc. 6th European Workshop on Evolutionary Computation in Image Analysis and Signal Processing (EVOIASP), LNCS 3005, Springer, Berlin, (2004) 379-388
9. Bremananth, R., Balaji, B., Sankar, M., Chitra, A.: A New Approach to Coin Recognition Using Neural Pattern Analysis. Proc. INDICON, Annual IEEE,(2005) 366-370
10. McNeill, S., Schipper, J., Sellers, T.: Coin Recognition Using Vector Quantization and Histogram Modeling. 17th Florida Conf. on Recent Advances in Robotics (FCRAR), www.mil.ufl.edu/publications/fcrar04/fcrar2004_coin.pdf, (2004)
11. Huber, R., Ramoser, H., Mayer, K., Penz, H., Rubik, M.: Classification of Coins Using an Eigenspace Approach. Pattern Recognition Letters (2005) 61-75
12. Ballard, D.H., Brown, C.M.: Computer Vision. Prentice Hall, Englewood Cliffs, (1982)
13. Castleman, K.R.: Digital Image Processing. Prentice Hall, Upper Saddle River, N.J. (1996)
14. Vassilas, N., Charou, E.: A New Methodology for Efficient Classification of Multispectral Satellite Images Using Neural Network Techniques. Neural Processing Letters, Vol. 9, No. 1 (1998) 35-43
15. Vassilas, N.: Efficient Neural Network-Based Methodology for the Design of Multiple Classifiers. In: Jain, L.C., Fanelli, A-M. (eds.): Recent Advances in Artificial Neural Networks – Design and Applications. CRC Press, New York (2000) 95-125
16. Kohonen, T.: Self-Organizing Maps. Springer, Heidelberg (1995)

Learning Time-Series Similarity with a Neural Network by Combining Similarity Measures*

Maria Sagrebin[1] and Nils Goerke[2]

[1] Fakultät für Ingenieurwissenschaften, Universität Duisburg-Essen, Germany
maria.sagrebin@uni-due.de
http://www.uni-due.de/is
[2] Div. of Neural Computation, Dept. of Computer Science,
University of Bonn, Germany
goerke@nero.uni-bonn.de
http://www.nero.uni-bonn.de

Abstract. Within this paper we present the approach of learning the non-linear combination of time-series similarity values through a neural network. A wide variety of time-series comparison methods, coefficients and criteria can be found in the literature that are all very specific, and hence apply only for a small fraction of applications. Instead of designing a new criteria we propose to combine the existing ones in an intelligent way by using a neural network. The approach aims to the goal of making the neural network to learn to compare the similarity between two time-series as a human would do. Therefore, we have implemented a set of comparison methods, the neural network and an extension to the learning rule to include a human as a teacher. First results are promising and show that the approach is valuable for learning human judged time-series similarity with a neural network.

1 Introduction

Time-series similarity have received a growing interest not only in the Music Information Retrieval community but in a number of different disciplines and applications as well. Depending on the problem and the time-series [4] a lot of different approaches, measures and criteria have been developed to compute the similarity of two given time-series. Unfortunately the approaches are often very distinct and therefore only applicable in a very small problem field.

Instead of designing an additional criteria, we propose to combine the existing criteria in a learning way.

2 The Basic Idea

The basic idea of the proposed approach is to combine the existing similarity measures by using a neural network, and develop a learning scheme that applies for a human teacher. The architecture of the developed system is shown in figure 1. The system receives two time-series **X** and **A** to compare. These

* This thesis and the corresponding work were done at the University of Bonn.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 123–132, 2006.

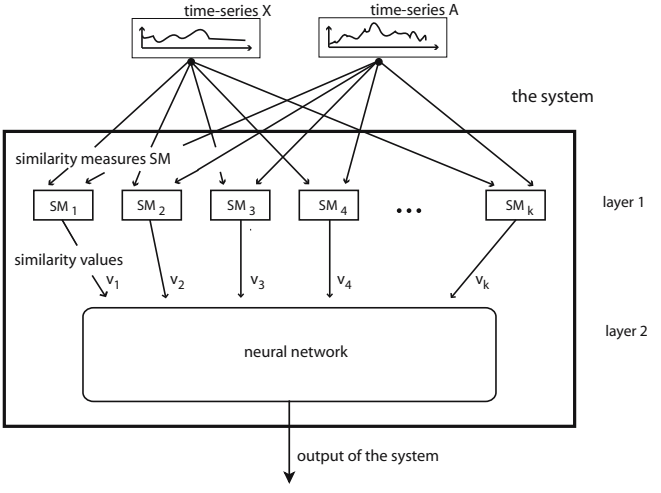


Fig. 1. Architecture of the developed system

two time-series are the input for a set of classical time-series similarity measures ($\mathbf{SM}_{i,i=1,\ldots,k}$). The output from these time-series similarity measures build the input for the neural network, that learns to combine them in a non-linear way. The output of the neural network is the output of the complete system. Of crucial interest are two major aspects for designing this task:

1) What neural network topology, and what neural training parameters are adequate for learning such a task?
2) How can we obtain teacher values from a human observer? We can neither expect that humans give exact similarity values for a set of given time-series, nor can we ask them to judge hundreds or thousands of time-series comparisons. Therefore, a special learning scheme has to be developed and evaluated before we can train the network with human generated teacher values. Directly aligned with the goal to obtain human teacher values the development of the training procedure is subdivided into five phases:

Phase I: Train the neural network with different **SM**s as teacher, to find the best network topology, and to evaluate the influence of weight initialisation.
Phase II: Change the teacher from a continuous valued function into a discretised teacher with 5 classes *(very unalike, unalike, neutral, alike, totally alike)* and train the best network found in Phase I to learn these 5 categories.
Phase III: Improve the discretised teacher by consulting *a second opinion* that is judging the quality of two trials of time-series comparison. Modify the teacher values accordingly and train the network for a while.
Phase IV: Reduce the number of second opinion hints as much as possible with respect to the given data and the application for human teachers.
Phase V: Get a human teacher to rank a subset of all possible time-series comparisons into 5 categories. Ask a human expert to judge some of the neural network comparison proposals. Train the network with these results.

3 Implementation Details

3.1 Time Series Data

The corpus of 18 exemplary audio time-series has been selected from five classes of audio signals. Hereby we considered that any two audio signals of the same class should be more similar to each other than any two audio signals which belong to different classes. This constraint was reasonable to make sure that the similarity values of these time-series can be easily judged by a human teacher.

1. **Superposition of harmonic sounds.** Sounds that are built up from different harmonic partial tones. Exemplary representatives of this class are sounds of the telephone keys. The sampling rate is 22050 *Hz*; the size varies from 2548 to 2956 data points.
2. **Repetitive technical sounds.** Signals which feature by means of repeated irregular clatter. Exemplary representatives of this class are sound recordings of an autonomous robot, driving with the speed of 20cm/s, 40cm/s and 60cm/s respectively. The sampling rate is 22050 *Hz*; the size varies from 16108 to 26248 data points.
3. **Repetitive technical sounds 2.** Signals from group 3 are similar to those of group 2. The major difference results from the characteristic of the clatter. Thus, signals of group 2 and 3 are more similar to each other than to those of other groups. The sounds have been produced by the same robot but with substantial different floor covering. Like the representatives of the second group the audio signals of group 3 have a sample rate of 22050 *Hz*; the size varies from 23837 to 29007 data points.
4. **Non-Rhythmic, long lasting sounds.** Signals which sound somehow "clanking", like banging together large pieces of metal. The signals vary from each other by the number of bangs and the way of echo. These metallic noise signals were recorded with a sampling rate of 44100 *Hz*; the size varies from 21396 to 41988 data points.
5. **Finite sound events.** Noise, and sounds that have a clear start, and a clear ending. Exemplary representatives of this class are sound recordings that were produced by punching paper. The sampling rate is 44100 *Hz*; the size varies from 13665 to 38775 data points.

3.2 Similarity Measures

We have implemented a set of 16 common time-series similarity measures (**SM1-SM16**) based on recent literature [2] [3] [4] [6] about multi-media retrieval and speech and language recognition. These methods have been chosen to represent a variety of different approaches that measure the similarity between time-series.

SM1: Computation of the best possible alignment between the Fourier transforms of the two signals by using dynamic time warping.

SM2: Computation of the best possible alignment between the two signals by using dynamic time warping.

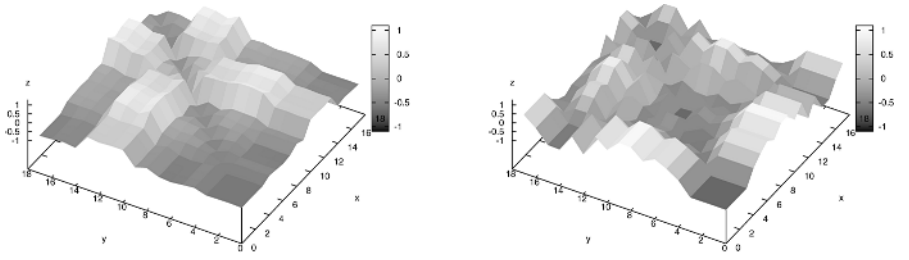
SM3: Computation of the Euclidean distance between the two signals.
SM4: Computation of the Euclidean distance between the two low-energy features [7] of the signals.
SM5: Computation of the Euclidean distance between the two spectral centroid feature [7] vectors of the signals.
SM6: Computation of the Euclidean distance between the two spectral flux feature [7] vectors of the signals.
SM7: Computation of the Euclidean distance between the two spectral rolloff feature [7] vectors of the signals.
SM8: Computation of the Euclidean distance between the two time domain zero crossing [7] feature vectors of the signals.
SM9: Computation of the best possible alignment between the hash-signatures [5] of the signals by using dynamic time warping.
SM10: Computation of the hamming distance between the hash-signatures of the signals.
SM11: Mapping of the Fourier transforms to the Bark scale and computation of the Euclidean distance.
SM12: Computation of the Euclidean distance between the two Fourier transforms of the signals.
SM13: Computation of the maximum correlation coefficient.
SM14: Computation of the best possible alignment between the Mel-frequency cepstral coefficients of the two signals by using dynamic time warping.
SM15: Discretisation of the co-domain in 20 equally large sections; determination of the average of the signal values within such a section; computation of the length of the longest common subsequence.
SM16: Discretisation of the co-domain in 20 equally large sections; determination of the average of the signal values within such a section; computation of the Levenshtein distance between the two modified signals.

Each implemented similarity measure receives two complete time-series with individual length and yields one scalar output value. Some of these measures compute the distances between the time-series instead of a similarity value. This fact does not cause any problems, because the neural network used as the learning unit can interpret these values correctly during the learning process.

For each of the 16 similarity measures we computed a similarity matrix by comparing each of the 18 time-series to the other 18 time-series. Thus, each similarity matrix consists of $18 \times 18 = 324$ similarity values. Two typical similarity matrices (**SM8** and **SM4**) are depicted in figure 2.

3.3 Learning Unit

The learning unit is a neural network of Multi Layer Perceptron (MLP) type. The number of neurons N in the input layer is stated by the number of similarity measures which are built into the system. The number of neurons h in the hidden layer was determined by several series of experiments. The output layer contains only one neuron, since the objective is one similarity value. We decided to use the hyperbolic tangent as transfer function in the hidden and the output layer, because it performed substantially better than the logistic transfer function.



(a) (b)

Fig. 2. Similarity matrices **SM8** (a) and **SM4** (b)

3.4 Learning and Validation

The neural network with an N-h-1 topology was trained using backpropagation of error, with different learning rates for the hidden and the output layer. During the training process the learning rate for the hidden layer was deliberately decreased three times, from 0.5 to 0.25 (after 20% of the epochs were passed), to 0.125 (after 40%), and to 0.05 (after 70%). Accordingly, the learning rate for the output layer was decreased from 0.1 to 0.05, to 0.025, to 0.01. The training process consisted of 30,000 epochs. During learning the following steps were accomplished:

1. Allocation of the 18 audio signals in 18 different tuples of training- and test-sets, following the leave-one-out-strategy [8]. Each test-set contains only one of the 18 audio signals, and each training-set the 17 other signals. Thus, each training-set consists of all remaining combinations, $17 \times 17 = 289$ training patterns. Each test-set consists of the remaining $17 + 18 = 35$ possible combinations.
2. Fix the neural network topology and initialise the weights randomly.
3. Random selection of the first pair of sets, the test- and the related training quantity.
4. Training of the neural network with backpropagation of error by means of the patterns from the training-set and simultaneous validation of the network by means of the patterns from the test-set. The weight combination which yields the best validation result was saved for further usage: early stopping strategy [8].
5. New random selection of the next but different training- and test-set and initialisation of the neural network with the weights saved during step 4.
6. Repeat step 5 until all training- and test-sets are processed.
7. Initialisation of the neural network with the weights saved during the last iteration and computation of the overall error E(h) by presenting all of the available $18 \times 18 = 324$ patterns to the neural network.

4 Phase I: Determine the Network Topology

During phase I two different Similarity Measures served as teachers for the neural network. Taking **SM8** as the teacher means, that the network has to learn the similarity values for **SM8** by combining the 15 other **SM**s in an intelligent way. It showed to be an easy task for the network to learn **SM8**, because some of the other 15 **SM**s results resemble **SM8**. The similarity measure **SM4** was more difficult to learn, because none of the other 15 **SM**s was alike.

To determine a valuable network topology (number h of hidden neurons) we have conducted several training runs with the described leave-one-out-strategy for all 18 test- and training-set combinations with an early-stopping BP learning scheme for 7 different numbers of hidden neurons $h = \{1, 3, 5, 7, 10, 15, 30\}$. Thus the following network configurations were tested: 15-1-1, 15-3-1, 15-5-1, 15-7-1, 15-10-1, 15-15-1 and 15-30-1. Figure 3 shows the overall error E(h) with respect to the number of the hidden neurons. Table 1 contains the related error values with respect to the network topology.

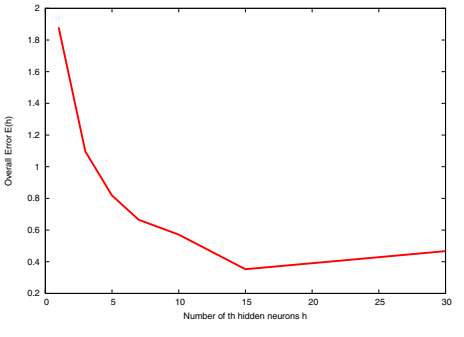


Fig. 3. Overall error E(h) with respect to the number of hidden neurons h

Table 1. overall error with respect to the network topology

network topology	overall error
15-1-1	1.880
15-3-1	1.096
15-5-1	0.818
15-7-1	0.665
15-10-1	0.571
15-15-1	0.353
15-30-1	0.467

At first the increase in the number of hidden neurons causes a decrease in the overall error. The error decreases from 1.880 at 15-1-1 to 0.353 at 15-15-1. Further increase of the hidden layer size revealed a slight rising of the overall error. As a second criteria we have taken into account the distribution of error values over all 324 patterns, and the percentage of results with an error larger than a confidence threshold of $\Theta = 0.07$ see Fig. 4. The found error value for a 15-1-1 networks shows (see Fig. 3 and 4) that a linear combination of the 15 similarity measures (**SM**) is inadequate to approximate one of the other similarity measures. One can conclude from that, that learning a non-linear combination with a neural network is a valuable approach.

Fig. 4 shows that the performance of 15-7-1, 15-10-1 and 15-15-1 networks differ very little from each other. On the basis of these results only the 15-7-1, 15-10-1 and 15-15-1 networks were examined in the following. The 15-30-1

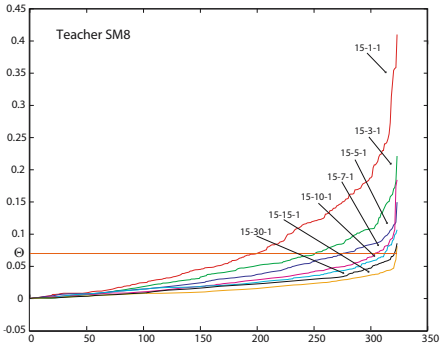


Fig. 4. Error distribution (teacher **SM8**) with respect to network topology, confidence threshold $\Theta = 0.07$

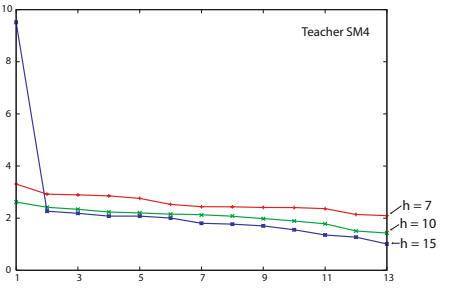


Fig. 5. Sorted error values for 13 initialisations for the three topologies 15-7-1, 15-10-1, 15-15-1 (teacher **SM4**). Only one initialisation shows an unacceptable value.

network was no longer considered because the results are comparable to those of the 15-10-1 network, but computing time is larger.

To investigate the effect of different network initialisations, we have conducted a second series of experiments with **SM4** as teacher, and $h = \{7, 10, 15\}$ respectively.

Each network was initialised 13 times, and the resulting error (BP, leave-one-out, early-stopping) was recorded. Figure 5 shows the resulting errors for all 13 runs, sorted by magnitude. Since only one of these initialisations lead to an unacceptable large error value, we conduct all further experiments without multiple initialisations. Still the larger networks perform obviously better (15-7-1: 2.0968 and 15-15-1: 1.008). Please remember that the teacher **SM4** is harder to learn than teacher **SM8**.

5 Phase II: Learn a Discretised Teacher

To pay respect to the fact, that a human expert can neither give double precision similarity values as result, nor can be persuaded to judge hundreds of pairs of time-series, we have to develop a training scheme for the neural network to bypass this. Thus, we have changed the teacher (**SM3**) from a continuous valued function into a discretised one, with the 5 classes:

Very unalike, Unalike, Neutral, Alike, Totally alike

and trained the best network topology found in Phase I to learn these 5 categories. The center of each class is used as teacher value. As expected, the neural 15-15-1 network performed well with the training details from Phase I; no wrong classifications occured; 100% of correctly learned pattern classes.

6 Phase III: Second Opinion Teaching

To improve the neural network training beyond the 5 classes discretisation, we have developed an extension to the training procedure of Phase II. Consider the

case where the network generated class is the same as the teacher specified class. Caused by the rather rough discretisation of the teacher, we lose the capability to fine tune the result within the correctly learned classes. A human expert will not be able to provide a similarity value with a higher resolution, but he might provide a differential judgement: Based on results from psychophysics we postulate, that a human teacher can provide the information if the similarity $S(\mathbf{X}, \mathbf{A})$ between time-series $\mathbf{X}$ and time-series $\mathbf{A}$ is greater or smaller than between $\mathbf{X}$ and $\mathbf{B}$. With this *extra* information, the learning process can enter a new stage of fine tuning. In Phase III, we simulate a very patient human teacher, by asking the calculated teacher (**SM3**) to judge the relation between two similarity calculations `IF` $S_T(\mathbf{X}, \mathbf{A}) < S_T(\mathbf{X}, \mathbf{B})$.

We compare if the relation of two time-series similarity calculations from the teacher is the same as for the neural network generated similarity. Now again two possible results can occur:

Correct Order: `IF` $S_T(\mathbf{X}, \mathbf{A}) < S_T(\mathbf{X}, \mathbf{B})$ `AND` $S_{\mathrm{MLP}}(\mathbf{X}, \mathbf{A}) < S_{\mathrm{MLP}}(\mathbf{X}, \mathbf{B})$
The neural network has the same relation than the teacher: we are fine, no further learning is necessary; proceed with next pattern.

Wrong Order: `IF` $S_T(\mathbf{X}, \mathbf{A}) < S_T(\mathbf{X}, \mathbf{B})$ `AND` $S_{\mathrm{MLP}}(\mathbf{X}, \mathbf{A}) > S_{\mathrm{MLP}}(\mathbf{X}, \mathbf{B})$
The neural network has produced a different relation than the teacher: we have to train the neural network further. Therefore we have to generate a more sophisticated teacher value.

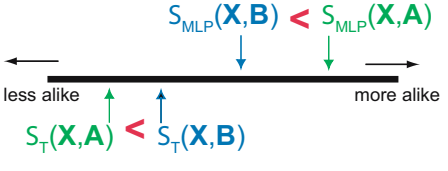


Fig. 6. Wrong order generated by the MLP

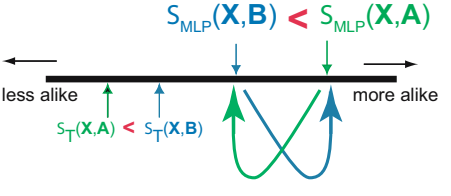


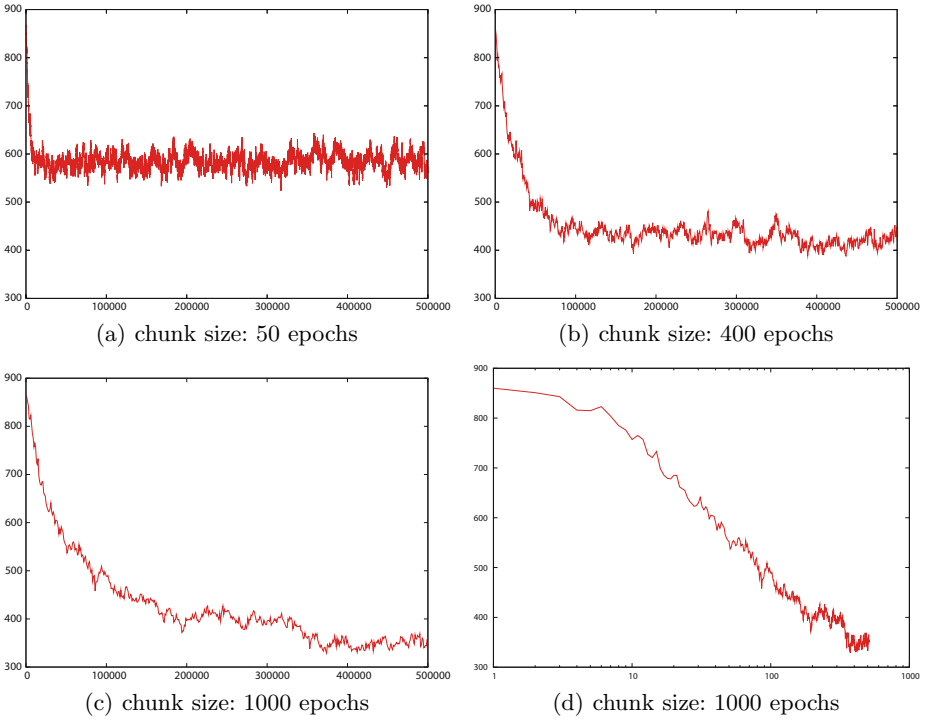
Fig. 7. Principle of the crossover teaching

Since we know, that the order is wrong, we can generate a set of two new teacher values that have the correct order by **crossover teaching**:

Make the output of one trial become the teacher for the other trial. The output from $S_{\mathrm{MLP}}(\mathbf{X}, \mathbf{A})$ will become the **teacher** for $S_{\mathrm{MLP}}(\mathbf{X}, \mathbf{B})$, and vice versa. Please keep in mind that the network has learned the correct classes, crossover teaching is for fine tuning, and will keep this performance.

For each class we generate one pair of crossover teacher values. With this newly generated training-set we train the network using backpropagation of error (learning rate like before) for a distinct number of epochs: a "chunk". As soon as the chunk of learning steps has been processed, a new training-set is generated with a new pair of randomly chosen crossover teacher values per class.

With respect to the chunk size of crossover teaching, we obtained a valuable decrease of wrong ordered pairwise comparisons to one third, see Fig. 8.



(a) chunk size: 50 epochs (b) chunk size: 400 epochs

(c) chunk size: 1000 epochs (d) chunk size: 1000 epochs

Fig. 8. Decrease of wrong ordered pairwise comparisons for different chunk sizes. All classes remain correctly learned.

In the beginning of the crossover teaching process 864 wrong ordered comparisons occur. These are 15% of all possible ordered comparisons ($5526 = 18 \times 17 \times 18$). As one can see, the choice of the chunk size is important. Larger chunks lead to better results. With a chunk size of 1000 epochs the number of wrong ordered pairwise comparisons decreases from 864 to 333 which is only 6% of all possible ordered comparisons.

7 Conclusions

As an alternative approach to time-series comparison we propose to combine existing time-series similarity measures by a neural network in a learning way. Instead of designing a new method we make benefit of the existing ones, and combine them by a neural network. A variety of 16 methods have been implemented to calculate the similarity between any two series from the set of chosen exemplary audio time-series.

Within exhaustive simulations, following the leave-one-out-strategy, and early-stopping, we have determined a network topology (15-15-1 MLP) that is capable of learning to imitate any of the implemented similarity measures as real valued teacher and as discrete classes (eg. 100% correct classifications for **SM3**).

To pay respect to a human teacher who can judge the similarity of two time-series to fall into one of five discrete classes, we have developed a new learning scheme to fine tune the neural network result. Therefore we ask the teacher to judge if the neural network results of two time-series comparisons have the correct order or not. From this judgement, we can generate new teacher values (crossover teaching) to improve the network further. The results are promising and show that the approach is in principle capable for making a neural network to learn the human time-series similarity judgement.

Phase IV and V are future work and include exhaustive tests with human volunteers, which have not been conducted yet. Part of our current work is aiming to realise these psychophysical, psychoacoustic experiments in the near future.

References

1. Delgorge C., Rosenberger C., Poisson G., Vieyres P.: 'Evaluation of the Quality of Ultrasound Image Compression by Fusion of Critaria with a Genetic Algorithm', *Third International Conference on Advances in Pattern Recognition, ICAPR 2005*, UK, August 2005
2. Aucouturier J., Pachet F.: 'Finding songs that sound the same', IEEE Benelux Workshop on Model Based Processing and Coding of Audio, pages 91-98, Leuven, Belgium, 2002
3. Aucouturier J., Pachet F.: 'Improving Timbre Similarity: How high i's the sky?', *Journal of Negative Research Results in Speech and Audio Science*, 1(1), 2004
4. Berenzweig A., Logan B., P.W.Ellis D., Whitman B.: 'A Large-Scale Evaluation of Acoustic and Subjective Music Similarity Measures', *Third International Conference on Music Information Retrieval (ISMIR 03)*, Washington DC, 2003
5. Prof. Dr. Michael Clausen, Lecture: 'Multi Media Retrieval', Summer 2004, Department of Computer Science, University of Bonn, 2004
6. Tzanetakis G., Essl G., Perry C.: 'Automatic musical genre classification of audio signals', *Interna*
7. Tzanetakis G., Perry C.:'Musical Genre Classification of Audio Signals', IEEE Transaction on Speech and Audio Processing, Vol. 10, No. 5, July 2002
8. Haykin, S.: "Neural Networks: A Comprehensive Foundation", Prentice Hall, 1999
9. Zell, A.; "Simulation Neuronaler Netze", Oldenbourg Verlag, 2000.

Prediction Improvement via Smooth Component Analysis and Neural Network Mixing

Ryszard Szupiluk[1,2], Piotr Wojewnik[1,2], and Tomasz Ząbkowski[1,3]

[1] Polska Telefonia Cyfrowa Ltd., Al. Jerozolimskie 181, 02-222 Warsaw, Poland
{rszupiluk, pwojewnik, tzabkowski}@era.pl
[2] Warsaw School of Economics, Al. Niepodleglosci 162, 02-554 Warsaw, Poland
[3] Warsaw Agricultural University, Nowoursynowska 159, 02-787 Warsaw, Poland

Abstract. In this paper we derive a novel smooth component analysis algorithm applied for prediction improvement. When many prediction models are tested we can treat their results as multivariate variable with the latent components having constructive or destructive impact on prediction results. The filtration of those destructive components and proper mixing of those constructive should improve final prediction results. The filtration process can be performed by neural networks with initial weights computed from smooth component analysis. The validity and high performance of our concept is presented on the real problem of energy load prediction.

1 Introduction

The blind signal separation methods have growing range of applications in telecommunications, medicine, economics and engineering. Starting from separation problems, BSS methods are used in flirtation, segmentation and data decomposition tasks [4,5,10,20]. In this paper we apply the BSS method for prediction improvement when many models are tested.

The prediction problem as other regression tasks aims at finding dependency between input data and target [14]. This dependency is represented by a specific model e.g. neural networks [2]. In fact, in many problems we can find different acceptable models where the ensemble methods can be used to improve final results [7]. Usually solutions propose the combination of a few models by mixing their results or parameters [1,8,23]. In this paper we propose an alternative concept based on the assumption that prediction results contain the latent destructive and constructive components common to all the model results. The elimination of the destructive ones should improve the final results. To find those basis components we apply a new algorithm for smooth component analysis. The full methodology will be tested in load prediction framework.

2 Blind Signal Separation and Data Representation

Blind signal separation (BSS) methods aim at identification of the unknown signals mixed in the unknown system [4,10]. The BSS last developments tend to its general

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 133 – 140, 2006.

formulation as the problem of the matrix $\mathbf{X} \in R^{m \times N}$ factorization [5]. The research task is to find the interesting analytical representation of $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m]$ associated with the data model assumption. For the linear case we are looking for

$$\mathbf{X} = \mathbf{AS}, \tag{1}$$

where $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_n]^T$, $\mathbf{S} \in R^{n \times N}$ is the matrix with n latent components ($\mathbf{s}_i$ is typically interpreted as source data), matrix $\mathbf{A} \in R^{m \times n}$ represents the mixing system, N means the number of observations. Typically we assume that rows $\mathbf{x}_i$ of the factorized matrix represent observed variables e.g. physical signals. The estimation of the $\mathbf{S}$ can be obtained by the transformation

$$\mathbf{S} = \mathbf{WX}, \tag{2}$$

where $\mathbf{W}$ matrix represents the separation system inverse to the mixing system. Though, in classic BBS problem estimated matrix $\mathbf{S}$ can be rescaled and permuted comparing to the original one [4,10], it is not a difficulty in our methodology. In practice the BSS separation can be obtained in many ways depending on the real characteristics of source signals like statistical independence, decorrelation, sparsity, nonstationarity, nonnegativity or smoothness. In this way in the BSS area there are many analytical methods exploring different properties of data. The most popular are Independent Component Analysis (ICA) [4,10], Sparse Component Analysis (SCA) [6,13,17,24], Nonnegative Matrix Factorisation (NMF) [12,25], Time Delay Decorrelation [3,4] or Smooth Component Analysis (SmCA) [4]. The choice of particular method depends on the nature of the problem and characteristics of the processed data. We apply the BSS methods for filtration of some unwanted components from the set of prediction results.

3 Prediction Results Improvement

We assume that after the learning process each prediction result includes two types of components: constructive associated with the target and destructive associated with the inaccurate learning data, individual properties of models, missing data, not precise parameter estimation, distribution assumptions etc. Now we collect particular model results together and treat them as multivariate variable $\mathbf{X}$. In similar way we assume that the set of basis components is represented by $\mathbf{S}$. The relation between observed prediction results and latent basis components is expressed by (1) and means matrix $\mathbf{X}$ factorisation by basis components matrix $\mathbf{S}$ and mixing matrix $\mathbf{A}$. Our aim is to find such mixing matrix $\mathbf{A}$ and unknown basis components set that matrix $\mathbf{S}$ (after rows ordering) can be described as

$$\mathbf{S} = \begin{bmatrix} \mathbf{T} \\ \mathbf{V} \end{bmatrix}, \tag{3}$$

where $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_p]^T$ is a $p \times N$ matrix constructive components, $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_q]^T$ is a $q \times N$ matrix destructive components. After basic

components are classified into destructive and constructive ones we can reject the destructive components **V** (replace them with zero) to obtain only constructive basis components matrix

$$\hat{\mathbf{S}} = \begin{bmatrix} \mathbf{T} \\ \mathbf{0} \end{bmatrix}. \tag{4}$$

Now we can mix the cleaned basis results back to obtain improved prediction results

$$\hat{\mathbf{X}} = \mathbf{A}\hat{\mathbf{S}} = \mathbf{A}\begin{bmatrix} \mathbf{T} \\ \mathbf{0} \end{bmatrix}. \tag{5}$$

The replacement of the destructive signal by zero in (5) is equivalent to putting zero in the corresponding column of **A**. If we express the mixing matrix as $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n]$ the purified results can be described as

$$\hat{\mathbf{X}} = \mathbf{A}\hat{\mathbf{S}} = \hat{\mathbf{A}}\mathbf{S}, \tag{6}$$

where $\hat{\mathbf{A}} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_p, \mathbf{0}_{p+1}, \mathbf{0}_{p+2}, ..., \mathbf{0}_n]$.

The effectiveness of the method highly depends on the application of proper transformation providing searched basis components and next it is important to perform proper distinction **T** from **V**. The choice of transformation type can be done in similar way like in other BSS problems, so it is based on some a priori assumptions or on the analysis of data characteristics.

4 Data Variability and Smooth Component Analysis

In our methodology we focus on data with temporal structure, characterised by their variability and therefore Smooth Component Analysis is developed. The analysis of signal smoothness is strongly associated with the definitions and assumptions about such characteristics. When we treat the data as random variable the popular measures of their variability are variance or Hurst exponent [9,18]. However, they are recommended mostly for data without temporal structure or when the data are randomly sampled e.g. noises. Whereas in many cases the data order is important and temporal structure can not be neglected what leads to stochastic processes analysis. On the other hand the data description as a stochastic processes is associated with many restrictive assumptions like ergodicity or nonstationarity often difficult to verify in practise [22]. For this reason we propose a new smoothness measure using random variables with delays (indexed random variables) [15]. Now

$$\mathrm{P}(\mathbf{s}) = \frac{\frac{1}{\mathrm{N}}\sum_{\mathrm{k}=2}^{\mathrm{N}} |\,\mathbf{s}(\mathrm{k}) - \mathbf{s}(\mathrm{k}-1)\,|}{\max(\mathbf{s}) - \min(\mathbf{s}) + \delta(\max(\mathbf{s}) - \min(\mathbf{s}))}, \tag{7}$$

where symbol $\delta(.)$ means Kronecker delta. Measure (7) has simple interpretation: it is maximal when the changes in each step are equal to range (maximal change), and

is minimal when data are constant. The possible values are from 1 to 0. The Kronecker delta term is introduced to avoid dividing by zero.

Smooth Component Analysis (SmCA) is a method of the smooth components finding in a multivariate variable [4]. The components are taken as linear combination of signals $\mathbf{x}_i$ and should be as smooth as possible. Our aim is to find such $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_n]$ that for $\mathbf{S} = \mathbf{WX}$ we obtain $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_n]^T$ where $\mathbf{s}_1$ maximizes $P(\mathbf{s}_1)$ so we can write

$$\mathbf{w}_1 = \arg\max_{\|\mathbf{w}\|=1}(P(\mathbf{w}^T\mathbf{x})) \quad . \tag{8}$$

Having estimated the first $n-1$ smooth components the next one is calculated as most smooth component of the residual obtained in Gram-Schmidt orthogonalization:

$$\mathbf{w}_n = \arg\max_{\|\mathbf{w}\|=1}(P(\mathbf{w}^T(\mathbf{x} - \sum_{i=1}^{n-1}\mathbf{s}_i\mathbf{s}_i{}^T\mathbf{x}))), \tag{9}$$

where $\mathbf{s}_i = \mathbf{w}_i{}^T\mathbf{x}, i = 1 \ldots n$. As the numerical algorithm for finding $\mathbf{w}_n$ we can employ the conjugate gradient method with golden search as a line search routine. The algorithm outline for initial $\mathbf{w}_i(0) = rand$, $\mathbf{p}_i(0) = -\mathbf{g}_i(0)$ is as follows:

1. Identify the indexes l for extreme signal values:

$$l^{\max} = \arg\max_{l \in 1 \ldots N} \mathbf{w}_i{}^T(k)\mathbf{x}(l), \tag{10}$$

$$l^{\min} = \arg\min_{l \in 1 \ldots N} \mathbf{w}_i{}^T(k)\mathbf{x}(l), \tag{11}$$

2. Calculate gradient of $P(\mathbf{w}_i{}^T\mathbf{x})$:

$$\mathbf{g}_i = \frac{\partial\, P(\mathbf{w}_i{}^T\mathbf{x})}{\partial\, \mathbf{w}_i} = \frac{\sum_{l=2}^{N}\Delta\mathbf{x}(l) \cdot sign(\mathbf{w}_i{}^T\Delta\mathbf{x}(l)) - P(\mathbf{w}_i{}^T\mathbf{x}) \cdot (\mathbf{x}(l^{\max}) - \mathbf{x}(l^{\min}))}{\max(\mathbf{w}_i{}^T\mathbf{x}) - \min(\mathbf{w}_i{}^T\mathbf{x}) + \delta(\max(\mathbf{w}_i{}^T\mathbf{x}) - \min(\mathbf{w}_i{}^T\mathbf{x}))}, \tag{12}$$

where $\Delta\mathbf{x}(l) = \mathbf{x}(l) - \mathbf{x}(l-1)$,

3. Identify the search direction (Polak-Ribiere formula[19])

$$\mathbf{p}_i(k) = -\mathbf{g}_i(k) + \frac{\mathbf{g}_i{}^T(k)(\mathbf{g}_i(k) - \mathbf{g}_i(k-1))}{\mathbf{g}_i{}^T(k-1)\mathbf{g}_i(k-1)}\mathbf{p}_i(k-1), \tag{13}$$

4. Calculate the new weights:

$$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + \alpha(k) \cdot \mathbf{p}_i(k), \tag{14}$$

where $\alpha(k)$ is found in golden search.

If we employ the above optimization algorithm as a multistart technique we choose such $\mathbf{w}_i$ that $P(\mathbf{w}_i{}^T\mathbf{x})$ is minimal. The above algorithm can be recommended for separation problem with data including by temporal patterns. In our predictions

improvement problem basis components obtained via algorithm (10)-(14) are an interesting representation of data for further processing.

5 Neural Networks as Generalization Mixing

After basis component are estimated by e.g. SmCA we need to label them as destructive or constructive. The problem with proper signal classification can be difficult task because obtained components might be not pure constructive or destructive due to many reasons like improper linear transformation assumption or other statistic characteristics than explored by chosen BSS method [21]. Therefore particular component can have constructive impact on one model and destructive on the other or there may exist components destructive as a single but constructive in a group. In this way, for all the components' subset we check the impact of elimination them on the final results. The mixing matrix $\hat{\mathbf{A}}$ is the best matrix we can find by simple test with eliminating each combination of the components.

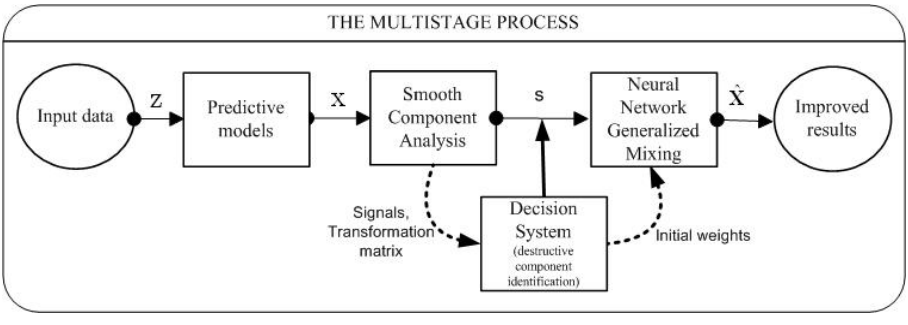


Fig. 1. The concept of filtration stage

However, the basis components can be not pure so their impact should have weight other than 0. It means that we can try to find the better mixing system than described by $\hat{\mathbf{A}}$. The new mixing system can be formulated more general than linear, e.g. we can take MLP neural network as the mixing system

$$\hat{\mathbf{X}} = \mathbf{g}^{(2)}(\mathbf{B}^{(2)}[\mathbf{g}^{(1)}(\mathbf{B}^{(1)}\mathbf{S} + \mathbf{b}^{(1)})] + \mathbf{b}^{(2)}), \tag{15}$$

where $\mathbf{g}^{(i)}(.)$ is a vector of nonlinearities, $\mathbf{B}^{(i)}$ is a weight matrix and $\mathbf{b}^{(i)}$ is a bias vector respectively for *i*-th layer, *i*=1,2. The first weight layer will produce results related to (4) if we take $\mathbf{B}^{(1)} = \hat{\mathbf{A}}$. But we employ also some nonlinearities and the second layer, so comparing to the linear form the mixing system gains some flexibility. If we learn the whole structure starting from system described by $\hat{\mathbf{A}}$ with initial weights of $\mathbf{B}^{(1)}(0) = \hat{\mathbf{A}}$, we can expect the results will be better, see Fig1.

6 Electricity Consumption Forecasting

The tests of proposed concept were performed on real problem of energy load prediction [11,16]. Our task was to forecast the hourly energy consumption in Poland in 24 hours basing on the energy demand from last 24 hours and calendar variables: month, day of the month, day of the week, and holiday indicator. In Fig. 2 you can observe some seasonal patterns of the energy demand. In long term high consumption in the winter while low usage in the summer. There are also shorter seasonalities: lower demand at non working days. Daily effects have two peaks: one in the morning and one in the afternoon.

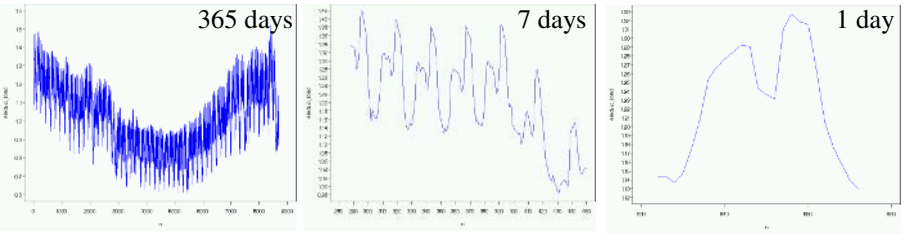


Fig. 2. Energy load in various time periods – you can observe the seasonalities

We trained a hundred MLP neural networks with one hidden layer on the observations from 1988-1997. The quality of the results was measured with MAPE and MSE criteria and six models were chosen for further consideration, see Table 1.

Table 1. Primary models

Criterion	MLP12	MLP18	MLP24	MLP27	MLP30	MLP33	*BEST*
MAPE	2,393	2,356	2,368	2,397	2,402	2,359	*2,356*
MSE [10^{-3}]	1,129	1,111	1,115	1,132	1,146	1,108	*1,108*

In Tables 2-3 you can observe the effects of modelling improvement by SmCA decomposition, negative components identification and then linear or neural mixing.

Table 2. Models after SmCA improvement

Criterion	MLP12	MLP18	MLP24	MLP27	MLP30	MLP33	*BEST*
MAPE	2,408	2,266	2,325	2,329	2,313	2,334	*2,266*
MSE [10^{-3}]	1,143	1,039	1,082	1,086	1,075	1,090	*1,039*

Table 3. The BEST MODELS: primary, and improved by SmCA and NN-SmCA

Criterion	Primary	SmCA	*Improved by %*	NN-SmCA	*Improved by %*
MAPE	2,356	2,266	*-3,83%*	2,225	*-5,56%*
MSE [10^{-3}]	1,108	1,039	*-6,23%*	1,017	*-8,21%*

In Fig. 3 you can see the visualization of the results.

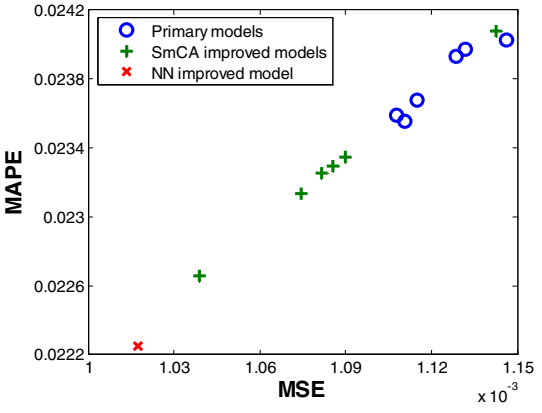


Fig. 3. The concept of the filtration stage

Proposed methods improved the modelling results both in MAPE (4-5%) and MSE (6-8%).

7 Conclusions

The Smooth Component Analysis can be successfully used as a novel methodology for prediction improvement. Presented method performs an efficient integration of the information generated by different models. The practical experiment of energy load prediction confirmed the validity of our method. In experiment we took into the consideration the filtration based on SmCa and the improvement obtained is considered as highly significant in this particular industry. Proposed method can be treated as an alternative for classical ensemble methods like boosting or bagging, but it has an advantage of clear interpretation of the identified components.

References

1. Breiman, L.: Bagging predictors. Machine Learning, Vol.24 (1996) 123-140
2. Bishop, C. M.: Neural networks for pattern recognition. Oxford Univ. Press, UK (1996)
3. Choi, S., Cichocki, A.: Blind separation of nonstationary sources in noisy mixtures, Electronics Letters, Vol. 36(9) (2000) 848-849
4. Cichocki, A., Amari, S.: Adaptive Blind Signal and Image Processing, John Wiley, Chichester (2002)
5. Cichocki, A., Zurada, J.M.: Blind Signal Separation and Extraction: Recent Trends, Future Perspectives, and Applications, ICAISC (2004) 30-37
6. Donoho D.L. and Elad M.: Maximal Sparsity Repre-sentation via l1 Minimization, The Proc. Nat. Aca. Sci., Vol. 100 (2003) 2197-2202
7. Haykin, S., Neural networks: a comprehensive foundation, Macmillan, New York (1994)

8. Hoeting, J., Mdigan, D., Raftery, A., Volinsky, C.: Bayesian model averaging: a tutorial. Statistical Science, Vol.14 (1999) 382-417
9. Hurst H.E.: Long term storage capacity of reservoirs. Trans. Am. Soc. Civil Engineers 116 (1951)
10. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley (2001)
11. Lendasse, A., Cottrell, M., Wertz, V., Verdleysen, M.: Prediction of Electric Load using Kohonen Maps – Application to the Polish Electricity Consumption, Proc. Am. Control Conf., Anchorage AK (2002) 3684-3689
12. Lee, D.D., Seung, H.S.: Learning of the parts of objects by non-negative matrix factorization. Nature, Vol.401 (1999)
13. Li, Y., Cichocki, A., Amari, S.: Sparse component analysis for blind source separation with less sensors than sources, Fourth Int. Symp. on ICA and Blind Signal Separation, Nara, Japan (2003) 89-94
14. Mitchell, T.: Machine Learning. McGraw-Hill, Boston (1997)
15. Molgedey, L., Schuster, H.: Separation of a mixture of independent signals using time delayed correlations, Phisical Review Letters, Vol. 72(23) (1994)
16. Osowski, S., Siwek, K., Regularization of neural networks for improved load forecasting in the power system, IEE Proc. Generation, Transmission and Distribution, Vol. 149(3) (2002) 340-344
17. Parra, L., Mueller, K.R., Spence, C., Ziehe, A., Sajda, P.: Unmixing Hyperspectral Data, Advances in Neural In formation Processing Systems 12, MIT Press (2000) 942-948
18. Samorodnitskij, G., Taqqu, M.:Stable non-Gaussian random processes: stochastic models with infinitive variance. N.York,London, Chapman and Hall (1994)
19. Scales, L. E.: Introduction to Non-Linear Optimization, New York: Springer-Verlag (1985)
20. Stone, J.V.: Blind Source Separation Using Temporal Predictability, Neural Computation, Vol. 13(7) (2001) 1559-1574
21. Szupiluk, R., Wojewnik, P., Zabkowski, T.: Model Improvement by the Statistical Decomposition. Artificial Intelligence and Soft Computing Proceedings. LNCS, Springer-Verlag Heidelberg (2004) 1199-1204
22. Therrien, C.W.: Discrete Random Signals and Statistical Signal Processing. Prentice Hall, New Jersey (1992)
23. Yang, Y.: Adaptive regression by mixing. Journal of American Statistical Association, Vol.96 (2001)
24. Zibulevsky, M., Kisilev, P., Zeevi, Y.Y., Pearlmutter, B.A.: Blind source separation via multinode sparse representation. In Advances in Neural Information Processing Systems, Vol. 14, (2002) 185-191
25. Cichocki, A., Zdunek, R., and Amari, S., "New Algorithms for Non-Negative Matrix Factorization in Applications to Blind Source Separation", 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP2006, Toulouse, France (2006)

Missing Value Estimation for DNA Microarrays with Mutliresolution Schemes

Dimitrios Vogiatzis[1] and Nicolas Tsapatsoulis[2]

[1] Department of Computer Science, University of Cyprus, CY 1678, Cyprus
phone: +30-2289-2749; fax: +30-2289-2701
dimitrv@cs.ucy.ac.cy
[2] Department of Telecommunications Science
and Technology University of Peloponnese Greece
ntsap@uop.gr

Abstract. The expression pattern of a gene across time can be considered as a signal; a microarray experiment is collection of thousands of such signals where due to instrument failure, human errors and technology limitations, values at some time instances are usually missing. Furthermore, in some microarray experiments the gene signals are not sampled at regular time intervals, which renders the direct use of well established frequency-temporal signal analysis approaches such as the wavelet transform problematic. In this work we evaluate a novel multiresolution method, known as the lifting transform to estimate missing values in time series microarray data. Though the lifting transform has been developed to deal with irregularly spaced data its usefulness for the estimation of missing values in microarray data has not been examined in detail yet. In this framework we evaluate the lifting transform against the wavelet transform, a moving average method and a zero imputation on 5 data sets from the cell cycle and the sporulation of the *saccharomyces cerevisiae*.

1 Introduction

Microarray experiments allow the simultaneous study of expression patterns of thousands of genes. As a consequence microarray datasets are characterized by a large number features (gene expression values) and a relatively small number of samples (different experimental conditions) [13]. The multiresolution theory (see [14]) is one of the most popular approaches for analyzing such a datasets. As its name implies, multiresolution theory is concerned with representation and analysis of signals at more than one resolution. The appeal of such an approach is obvious: Characteristics (similarities, regulatory patterns, etc.) that might go undetected at one resolution may be easy to spot at another.

The multiresolution theory is based on the wavelet transforms. Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small waves, called wavelets, of varying frequency and *limited duration*. This allows them to be used for identifying patterns, in a signal, localized both in frequency and time (1-D) or space (2-D). In a typical setting, the wavelet transform can be used to decompose a discrete signal into detail and approximation coefficients. The detail coefficients

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 141–150, 2006.

(approximations) coefficients correspond to high (low) frequencies. This constitutes the first step of the multiresolution analysis, the same process could be repeated for the approximation coefficients, which are further divided into approximation and detail coefficients. The first steps of the multiresolution analysis produced the finer signals components (finer scale) and the latter steps correspond to more coarse components (coarse scale).

Wavelets have been widely used in signal processing for more than 20 years. For an accessible introduction to the application of wavelets in statistics see [1]. Moreover, their usefulness has been proved in the domain of data mining [9] and they have been also been applied in the Biomedical domain [10], as well as in the analysis of microarray experiments [21].

An important problem of gene expression microarray experiments is that they can generate data sets with multiple missing expression values. There are many possible reasons for missing data in microarray experiments, such as technology limitations, human errors, instrument failure. Unfortunately, many algorithms for gene expression analysis, including multiresolution theory, require a complete matrix of gene array values as input. A few missing values in the feature set, and especially in cases where missing values do not correspond to the same genes across all samples, could lead to reduced effectiveness of multiresolution analysis due to the loss of synchronization among the features of the various samples. Methods for imputing missing data are needed, therefore, to increase the effectiveness of multiresolution analysis, as well as classification algorithms such as hierarchical and K-means clustering, on microarray data mining.

In this study, we consider the issue of missing value estimation from the point of view of signal processing. To this end we investigate the rather novel scheme of second order wavelets also known as lifting schemes as a method for estimating missing data in time series cDNA microarray experiments and we contrast it with the older scheme of discrete wavelet transforms.

In section 2 we put our work in the context of missing value estimation methods for microarrays, then in section 3 we refer briefly to the discrete wavelet transform but also to the novel lifting scheme which is central to our evaluation. Subsequently, in section 5 we report on the experimental setup and the results we obtained. Finally conclusions are drawn in section 6.

2 Missing Value Estimation Methods for Microarray Datasets

One solution to the missing data problem is to repeat the microarray experiment. However, this approach is very expensive and can be only used in validation of microarray analysis and data imputing algorithms [5]. In the simplest approaches missing value are either replaced by zeros [4] or, less often, by a moving average over the existing gene values (often called row average). Both methods do not take into consideration the correlation structure of the data and may lead to detection of synthetically generated patterns (artefacts) during the microarray data analysis stage. Thus, methods for more accurate estimation of missing values are required.

Although there are very few methods published in the literature concerning missing value estimation for microarray data, a lot of work has been conducted to similar problems in other fields. Probably the most similar context, with the missing value

estimation for microarray data problem, is the context of identifying missing data in experiments [11]. Common methods, in this framework, include least squares estimates, iterative analysis of variance methods [25], randomized inference methods, likelihood-based approaches [24], nearest neighbours [12], etc. All the above are well established methods that are, explicitly or implicitly, imply a generic probability density distribution for the data population. Unfortunately, this assumption can not easily justified for the microarray data where the identification of regulatory patterns in gene expression values is actually the question at hand.

In [22] the estimation of missing values in DNA microarrays is examined. The authors compare k-NN and SVD (Singular Value Decomposition)-based methods to the 'row average' method. Once again both k-NN and SVD estimate the missing values based on global characteristics of the data population, thus, ignoring the importance of the local neighbourhoods of the missing data. Similarly in [8] the authors handle the problem of missing values with a novel imputation scheme. According to this scheme genes with the missing values are represented as linear combinations of similar genes. In [6] an SVD impute method which is implemented as the Fixed Rank Approximation Algorithm is considered. A bayesian principal component analysis method has been also used as it is reported in [17]. In a very recent approach the a Gene Ontology (GO) is combined with a k-NN algorithm to predict missing values with encouraging results [23].

In our approach, missing values are estimated using a wavelet lifting scheme. As already stated, the main advantage of the wavelets is that they allow the localization of a signal in both the time and frequency domains. Thus, the importance of the local neighbourhood of missing data is preserved. However, classic wavelet regression can not directly applied to missing value estimation in irregularly spaced data sets. Fortunately, second generation wavelets and lifting schemes are appropriate methods to account for this problem.

3 Multiresolution Analysis and Microarrays

From the point of view of mathematics, a function can be represented as an infinite series expansion in terms of a dilated and translated version of a basis function called the *mother wavelet* denoted as $\psi(x)$ and weighted by some coefficient $b_{j,k}$.

$$f(t) = \sum_{j,k} b_{j,k}\psi_{j,k}(t) \tag{1}$$

Normally, a wavelet starts at time $t = 0$ and ends at time N. A shifted wavelet denote as ψ_{jo}, starts at time $t = k$ and ends at time $t = k + N$. A dilated wavelet w_{j0} starts at time $t = 0$ and ends at time $t = N/2^j$. A wavelet w_{jk} that is dilated j times and shifted k times is denotes as:

$$\psi_{j,k}(t) = \psi(2^j t - k) \tag{2}$$

For practical purposes, we can use the discrete wavelet transform, which removes some of the redundancy found in the continuous transform. In this study we rely on

wavelet shrinkage. The shrinkage is based on discarding some of the detail coefficients and then by reconstructing the signal based on the reduced set of coefficients. Moreover, in [2,3] it has been shown that the wavelet shrinkage method outperform other methods for denoising signals.

Wavelet analysis is based on the assumption that that the signal coefficients are observed in a regular grid, that is the data points are equidistant. However, this is not the case with all from the domain of gene expression. A novel scheme known as the lifting transform can deal with this issue.

3.1 Adaptive Lifting

Wavelet analysis can be applied to regularly distributed data, which means that the following hypothesis must hold $t_i = \frac{i}{n}$, n denotes the number of samples, and t_i is the sampling rate. However this might not be the case in some real world measurements including DNA microarray experiments. Moreover, in the discrete wavelet analysis the signal samples must be a power of two and each level of analysis considers 50% of the data of the previous level.

Next, the lifting transform that we describe was conceived in [16], where we refer to for more details. This transform can deal with irregularly spaced data. Similarly to the wavelet transform, in a lifting scheme a function f can be represented as a linear combination of scaling coefficients:

$$f(x) = \sum_{k=1}^{n} c_{n,k}\phi_{n,k}(x) \tag{3}$$

The first step of the transform is step n all the way till step 1. At the first step it holds that $\phi_{n,k}(x_i) = \delta_{i,k}$ and $f(x_{n,i}) = c_{n,i} = \sum_{k=1}^{n} c_{n,k}\delta_{i,k}$

The first phase is to *Lift* one point, which means that the point is removed. Up to $n-2$ points can be lifted for a signal of length n; let j_n be the first point to be lifted. The selection is based on removing first the points which correspond to the highest detail. This is defined as the point which corresponds to the smallest interval, provided we assign intervals to points, which start at midway after the previous point and end at midway before the next point. This is expressed as $\int \phi_{n,j_n}(x)dt = min \int \phi_{n,k}(x)dx$, where $k \in [1, n]$. The rationale is that the denser regions of the function being a result of higher frequency sampling can be seen as the detail coefficients which are removed first. The scaling coefficients correspond to the sparsely sampled regions of the function.

The second phase is to *Predict* the lifted point, which will produce the detail coefficient at the lifted point. Prediction is based on discovering through regression a polynomial curve that passes through the neighbouring points (let as denote them as J_n) of j_n. Prediction is of the following form: $\sum_{i \in J_n} a_i^n c_{n,i}$, where a_i^n are the polynomial coefficients resulting from the regression. The detail coefficient at the lifted point is the difference between the function value at this point and the prediction,

$$d_{j_n} = c_{n,j_n} - \sum_{i \in J_n} a_i^n c_{n,i} \tag{4}$$

The third phase is to *Update* the function samples that were part of the neighbourhood of the lifted point.

$$c_{n-1,i} = c_{n,i} + b_i^n d_{j_n},\ \forall i \in J_n, i \neq j_n \tag{5}$$

where b_i^n are the prediction coefficients and can be computed with a least squares method. After the lift, predict and update phases the function can be represented as:

$$f(x) = d_{j_n}\psi_{j_n}(x) + \sum_{i \in 1 \ldots n, i \neq j_n} c_{n-1,i}\phi_{n-1,i}(x) \tag{6}$$

The first term corresponds to a detail coefficient and the second to the scaling coefficient. In particular $\psi_{j_n}(x)$ is a wavelet function and the weights b result from the requirement that the integral of the wavelet function is zero [7]). The three step process *Lift, Predict* and *Update* can be repeated for more points. Adaptation is performed at the prediction phase, where the regression polynomial is selected with the criterion of producing the smallest detail coefficients. In addition, adaption can also be implemented by having a variable number of neighbours, for every lifted point, again with a view of obtaining the smallest detail coefficients. The power of the lifting transform is that in can be *adapted* to irregularly distributed points by selecting a polynomials (between degrees one to three in the current implementation) that conform to the local structure of the signal as mentioned above. It is also important to note that since, the aforementioned lifting method lifts one coefficient at a time as opposed to the discrete wavelet transform where 50% of the signal samples become detail coefficients, thus it is more "continuous" than the wavelet transform.

3.2 Smoothing in the Lifting Scheme

The problem of signal denoising has been thoroughly addressed in the statistics literature, and it can be stated (for a univariate function) as follows: $y(t_i) = g(t_i) + \epsilon(t_i), \quad i = 1, \ldots n$ Thus it is assumed that the observed values $y(t_i)$ stem from the unobserved values $g(t_i)$ with the addition of noise that follows the $N(0, \sigma^2)$ distribution. In the wavelets field, denoising can be achieved by assuming a wavelet transform of the above equation, which leads to: $d_{jk} = d_{jk}^* + e_{jk}$, where d_{jk} are the observed wavelet detail coefficients, d_{jk}^* are the "true" detail coefficients and e_{jk} is the wavelet transform of the noise. All this is true in the classic framework of wavelets where e_{jk} follows the $N(0, \sigma^2)$ distribution. However, the lifting scheme is not orthogonal and the noise will be correlated and different coefficients have different variances. Basically, it has been suggested to adapt the empirical Bayesian wavelet shrinkage approach to the lifting scheme [16].

In [18] the authors faced the problem of transmembrane protein prediction. The usual approach is through hydrophobicity analysis of the aminoacids that constitute the protein. They have used a multiresolution method (lifting based) to regress the Kyte Doolittle hydrophobicity index over the residues. The novelty they introduced is incorporation of the 3D structure of the proteins in their calculations. However, the residues are not equidistant rather they are irregularly distributed. Thus the direct use of wavelets was not possible, therefore the authors have introduced the lifting transform that we have briefly described.

4 Lifting Methods Employed

The problem we have addressed is that of missing value estimation in time series microarray experiments. The form of the data set is a series of vectors corresponding to genes, and each dimension corresponds to the expression level of a gene at a certain time step $g_k = (x^k_{t_1} \dots x^k_{t_n})$, where g_k is a gene that has been expressed by produced RNA (and hence protein) and $x^k_{t_1}$ denote the expression of that gene at different time steps. Usually, the number of genes is in the order of a few thousands and the number of time steps is two orders of magnitude less. Let us assume that we have a one dimensional signal, where a single value at a specific time step is missing, let us call the index of the missing value as p_M. The lifting based processing methods that we have employed to predict the missing value are described bellow:

In *Lifting I* the missing value is set to zero and all signals samples are lifted (apart from the last two), according to the scheme described in the previous section. Then the lifted coefficients, which correspond to details are set to zero, and the signal is reconstructed with the reverse lifting transform. The *Lifting II* is exactly the same as Lifting I, but instead of zeroing the detail coefficients, they are thresholded with a bayesian threshold. Both lifting I and II implement a signal denoising technique similar to wavelet based denoising, but instead of the wavelet transform we apply the lifting transform [18].

In *Lifting III* we detect two points, one before the missing point p_A and another one after the missing point p_B. Next, the neighbours of p_A will be used to predict p_A. The definition of neighbourhood includes the two immediate neighbours (one on the left of the predicted point and one on the right). The result of the prediction is a linear, quadratic or cubic polynomial that minimises the distance between the real value of p_A and its prediction. More accurately, point p_A is lifted and the polynomial that produces the smallest detail coefficients is chosen. Let us call it $f_A(x)$. Similarly, a polynomial will be produced for p_B, let as call it $f_B(x)$. The the prediction for the missing value is $\frac{f_A(p_M)+f_B(p_M)}{2}$, where p_M is the index of the missing point. The lifting III method differs from lifting I and II in that only two points are lifted (the neighbours of the missing point) and it that it does not involve a reverse transform.

5 Experiments and Results

Five time series data sets were used in our experiments. Four of them concern the cell cycle of the yeast *saccharomyces cerevisiae*. A cell cycle is the sequences of stages a cell passes from one division to the next. Microarray experiments were performed to identify all genes whose mRNA levels are regulated by the cell cycle [20]. We derived the data sets from the public site (`http://genome-www.stanford.edu/cellcycle/data/rawdata/`). We have also used the microarray sporulation data set [19] from (`http://cmgm.stanford.edu/pbrown/sporulation/`). In particular, at the experiments we used the alpha factor, cdc 15, cdc 28, elutriation and the sporulation time courses. We preprocessed all data sets by eliminating vectors with missing values (of course the number of time steps was not affected), but in some cases as much as 25% of the vectors were eliminated. In table 1, we report the characteristics

of the data sets. Two issues are important, first the cdc 15 and the sporulation data sets are *irregularly* distributed in time, and for all data sets the number of time steps is rather small compared to other types of signals.

Table 1. Time series microarray data sets

data set	# vectors	# time steps	Sampling rate (numbers denote minutes)
alpha	4489	18	from 0 to 119, every 7 min
cdc 28	1383	17	from 0 to 160 every 10 min
cdc 15	4382	24	from 10 to 70 every 20 min, from 70 to 240 every 10 min and then every 20 min till 290
elutriation	5766	14	from 0 to 390 every 30 min
sporulation	6117	7	at 0, 0.5, 2, 5, 7, 8, 11.5

For experimentation purposes a randomly selected value (time point) was removed from each one of the vectors (gene expression values across time) in the dataset and was considered as a missing value. We avoided removing the first and last time points; thus missing values correspond to time steps between 2 to $n-1$, where n is the length of the gene signal. Each experiment was performed 10 times to reduce randomness. The performance of each method is computed as:

$$\sum_{i=1}^{m} \frac{\left\| x_{t_l}^i - est_{t_l}^i \right\|}{m} \tag{7}$$

where $x_{t_l}^i$ is the real value, est_{t_l} the estimated value and m is the number of missing values to be estimated which is equal to the number of vectors of the data set we consider; $t_l \in [2, n-1]$ is a randomly chosen point across a time series.

We have compared the lifting I, II and III approaches elaborated at the previous section with a weighted moving average method and a discrete wavelet transform.

In the *weighted moving average*: The missing value of the signal is set to zero, and then we employ a gaussian like weighted filter whose central value is set to zero. Then the signal is convolved with the filter. After the convolution we obtain the missing value. In the *wavelet* method: The missing value of the signal is set to zero, then the discrete wavelet transform is applied with the Daubechy mother wavelet. We have applied one level of decomposition as the time series are very short (see table 1). Then the detail coefficients are set to zero and the signal is reconstructed with the reverse wavelet transform. After that we obtain the missing value.

The results appear in tables 2,3, where under the label zero we report the results obtained by filling the missing value with zero. The numbers denote average errors as computed from eq. 7 and the numbers in parentheses are the standard deviations. The ranges for the datasets are: alpha [-2.7100,4.7600], cdc 28 [-4.1200,3.0900], cdc 15 [-4.6300, 4.1400], elutriation [-6.2200,4.9500] and sporulation [-6.0012,4.4118]. As mentioned above, in the cdc 15 and the sporulation data sets the vectors' components are irregularly distributed but we chose to ignore that when applying the wavelet method, so as to discover the deterioration of the results as compared to the lifting schemes.

Table 2. Average error on the cdc 15 and cdc 28 datasets

	cdc 15	cdc 28
Zero	0.3102 (0.0049)	0.3244 (0.007)
Lifting I	0.3326 (0.0033)	0.3373 (0.004)
Lifting II	0.3178 (0.0036)	0.3222 (0.004)
Lifting III	0.3310 (0.0037)	0.2997 (0.005)
Moving Average	0.3771 (0.0038)	0.3113 (0.006)
Wavelet	0.3410 (0.0036)	0.3072 (0.005)

Table 3. Average error on the Alpha, Elutriation and Sporulation datasets

	Alpha	Elutriation	Sporulation
Zero	0.1939 (0.0022)	0.2140 (0.0016)	0.7166 (0.0035)
Lifting I	0.2039 (0.0018)	0.2124 (0.0018)	0.4693 (0.0058)
Lifting II	0.1963 (0.0019)	0.2122 (0.0014)	0.5744 (0.0063)
Lifting III	0.1938 (0.0021)	0.1917 (0.0012)	0.4224 (0.0034)
Moving Average	0.2067 (0.0020)	0.1780 (0.0015)	0.3576 (0.0050)
Wavelet	0.1955 (0.0022)	0.1893 (0.0015)	0.4361 (0.0032)

The lifting based experiments were performed on the R software package (`http://www.r-project.org/`) with the Adlift package developed by M.Popa and M.Nunes (`http://www.maths.bris.ac.uk/ maman/computerstuff/Adlift.html`). The rest of the experiments were performed on the Matlab 6.1 platform, with the wavelet toolbox (v.2.1) (`http://www.mathworks.com/`)

6 Conclusions and Future Work

We have evaluated various multiresolution schemes along with some classic methods to estimate missing values in signals derived from microarray experiments. We were especially interested in a rather novel multiresolution analysis scheme, the adaptive lifting transform and in particular the "lifting of one coefficient at a time" version. This scheme is adaptive in the sense that it conforms to the local structure of the signal by locally selecting approximation polynomials, and thus it can cope with signals where the samples are not evenly distributed. It is also more continuous that the wavelet transform, in that one coefficient is lifted at a time and the relevant detail coefficient is obtained, instead of obtaining detail coefficients for 50% of the signal components, as in the wavelet transform. We compared the lifting methods against more established ones, such as the discrete wavelet transform, a moving average method and the zero imputation (missing value is filled with zero). In each signal from each data set a random value was chosen to be estimated as if it was missing. The lifting I and II are based on denoising, in particular the lifting I employs a rather crude method: the detail coefficients are zeroed before the signal reconstruction. The lifting II method is based on a bayesian thresholding of the detail coefficients. Finally, the lifting III performs adaptive prediction of the missing value from its neighbourhood.

Two of the data sets, the cdc 15 and the sporulation, have irregularly distributed samples, where the wavelet method was applied without regard to the irregular time grid. On both the sporulation and cdc 15 the lifting III method outperformed the wavelet method, albeit by a short margin. Also on the cdc 15 dataset the lifting I and II methods were better than the wavelet method. On regularly distributed data sets, such as the elutriation set the moving average method is the overall winner. Another remark is that occasionally a trivial method such as the zero imputation is occasionally better than other methods (e.g. on the cdc 15 data set). The moving average method was the overall winner on the elutriation and sporulation data sets and the lifting III on the cdc 28 data set. The overall conclusion is that on irregularly sampled time series the classic wavelet transform is worse than the lifting transform.

It should be pointed out that the methods of missing value prediction that we evaluated were based only on single signals in the sense that we did not take into account the information that could be furnished by similar signals of the same data set as performed by the relevant research we reported in the introduction. For instance in [22] on cell cycle regulated genes, the error was in [0.05, 0.1] depending on the chosen parameters. Clearly lower, than that furnished by the methods we proposed. However, the purpose of our approach was to investigate the extend to which the correlation of the samples in a single signal can predict missing values, whereas in the relevant literature such information has not been investigated.

In future work we will investigate the extend to which the lifting transform can predict more that one missing values per signal, and especially when they are consecutive. This will probably impose a heavy burden on the classic wavelet transform or the moving average method since they do no adapt to the local signal structure. Moreover, another path that we plan to follow is to dynamically determine for the lifting transform the number of the components that should be lifted. In the current research all components were lifted apart from two. From experiments that we conducted the number of lifted coefficients is crucial for the accuracy of the prediction of the missing value. A good place to start the relevant investigation is the work described in [15].

References

1. F. Abramovich, T.C. Bailey, and T. Sapatinas. Wavelet Analysis and its statistical applications. *The Statistician*, 49:1–29, 2000.
2. D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard. Wavelet shrinkage: Asymptopia? *J. R. Statist. Soc. B.*, 57(2):301–337, 1995.
3. David L. Donoho and Iain M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.
4. A. A. Alizadeh et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.
5. A. J. Butte et al. Determining significant fold differences in gene expression analysis. In *Pac. Symp. Biocomput.*, pages 6–17, 2001.
6. S. Friedland, A. Niknejad, and L. Chihara. A simultaneous reconstruction of missing data in DNA microarrays. *Institute for Mathematics and its Applications Preprint Series*, (1948), 2003.
7. M. Jansen, G. P. Nason, and B. W. Silverman. Multivariate nonparametrix regression using lifting. Technical report, Department of Mathematics, University of Bristol, UK, 2004.

8. H. Kim, G.H. Golub, and H. Park. Missing value estimation for DNA microarray gene expression data: local least squares imputation. *Bioinformatics*, 21(2):187–198, 2005.
9. T. Li, Q. Li, S. Zhu, and M. Ogihara. A survey on wavelet applications in data mining. *SIGKDD Explorations*, 4(2):49–68, 2003.
10. P. Liò. Wavelets in bioinformatics and computational biology: state of art and perspectives. *Bioinformatics*, 19(1):2–9, 2003.
11. R. J. A. Little and D. B. Rubin. Statistical analysis with missing data. In *Wiley, New York*, 1987.
12. W. Loh and N. Vanichsetakul. Tree-structured classification via generalized discriminant analysis. *Journal of American Statistics Association*, 83:7157251, 1988.
13. P.F. Macgregor and J.A. Squire. Application of microarrays to the analysis of gene expression in cancer. *Clinical Chemistry*, 48(8):1170–1177, 2002.
14. S. Mallat. A theory of multiresolution signal decomposition: The wavelet model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, 1989.
15. M.A. Nunes and G.P. Nason. Stopping times in adaptive lifting. Technical Report 05:15, 2004.
16. M.A. Nunes, M.I. Popa, and G.P. Nason. Adaptive lifting for nonparametric regression. Technical Report 04:19, Statistics Group, Department of Mathematics, University of Bristol, UK, 2004.
17. S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara, and S. Ishii. A bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19:2088–2096, 2003.
18. M. I. Popa and G. P. Nason. Improving Prediction of Hydrophobic Segments along a Transmembrane Protein Sequence using Adaptive Multiscale Lifting. Technical Report 04:19, Statistics Group, Department of Mathematics, University of Bristol, UK, 2005.
19. S. Chu S, J. DeRisi, M. Eisen, J. Mulholland, D. Botstein, PO Brown, and I. Herskowitz. The transcriptional program of sporulation in budding yeast.
20. P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Botstein, and B. Futcher. *Molecular Biology of the Cell*, 9(12):3273–3297, 1998.
21. T. A. Tokuyasu, D. Albertson, D. Pinkel, and A. Jain. Wavelet transforms for the analysis of microarray experiments. In *IEEE Computer Society Bioinformatics Conference (CSB'03)*, page 429, 2003.
22. O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and RB. Altman. Missing value estimation methods for DNA microarrays.
23. J. Tuikkala, L. Elo, O.S. Nevalainen, and T. Aitkallio. Improving missing value estimation in microarray data with gene ontology. *Bioinformatics*, 22(5):566–572.
24. G. N. Wilkinson. Estimation of missing values for the analysis of incomplete data. *Biometrics*, 14:257286, 1958.
25. Y. Yates. The analysis of replicated experiments when the field results are incomplete. *Emp. J. Exp. Agric*, 1:129142, 1933.

Applying REC Analysis to Ensembles of Sigma-Point Kalman Filters

Aloísio Carlos de Pina and Gerson Zaverucha

Federal University of Rio de Janeiro, COPPE/PESC,
Department of Systems Engineering and Computer Science,
C.P. 68511 - CEP. 21945-970, Rio de Janeiro, RJ, Brazil
{long, gerson}@cos.ufrj.br

Abstract. The Sigma-Point Kalman Filters (SPKF) is a family of filters that achieve very good performance when applied to time series. Currently most researches involving time series forecasting use the Sigma-Point Kalman Filters, however they do not use an ensemble of them, which could achieve a better performance. The REC analysis is a powerful technique for visualization and comparison of regression models. The objective of this work is to advocate the use of REC curves in order to compare the SPKF and ensembles of them and select the best model to be used.

1 Introduction

In the past few years, several methods for time series prediction were developed and compared. However, all these studies based their conclusions on error comparisons.

Results achieved by Provost, Fawcett and Kohavi [15] raise serious concerns about the use of accuracy, both for practical comparisons and for drawing scientific conclusions, even when predictive performance is the only concern. They indicate ROC analysis [14] as a superior methodology than the accuracy comparison in the evaluation of classification learning algorithms. Receiver Operating Characteristic (ROC) curves provide a powerful tool for visualizing and comparing classification results. A ROC graph allows the performance of multiple classification functions to be visualized and compared simultaneously and the area under the ROC curve (AUC) represents the expected performance as a single scalar.

But ROC curves are limited to classification problems. Regression Error Characteristic (REC) curves [1] generalize ROC curves to regression with similar benefits. As in ROC curves, the graph should characterize the quality of the regression model for different levels of error tolerance.

The Sigma-Point Kalman Filters (SPKF) [10] is a family of filters based on derivativeless statistical linearization. It was shown that Sigma-Point Kalman Filters achieve very good performance when applied to time series [10].

Current research on time series forecasting mostly relies on use of Sigma-Point Kalman Filters, achieving high performances. Although most of these works use one of the filters from the SPKF family, they do not use an ensemble [4] of them, which

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 151 – 160, 2006.

could achieve a better performance. Therefore, the main goal of this paper is to advocate the use of REC curves in order to compare ensembles of Sigma-Point Kalman Filters and choose the best model to be used with each time series.

This paper is organized as follows. The next section has a brief review of REC curves. Then, a summary of the main characteristics of the Sigma-Point Kalman Filters is presented in Section 3. An experimental evaluation comparing the REC curves provided by each algorithm and ensembles of them is reported in Section 4. Finally, in Section 5, the conclusions and the plans for future research are presented.

2 Regression Error Characteristic Curves

Results achieved by Provost, Fawcett and Kohavi [15] indicate ROC analysis [14] as a superior methodology to the accuracy comparison in the evaluation of classification learning algorithms. But ROC curves are limited to classification problems. Regression Error Characteristic (REC) curves [1] generalize ROC curves to regression with similar benefits. As in ROC curves, the graph should characterize the quality of the regression model for different levels of error tolerance.

The REC curve is a technique for evaluation and comparison of regression models that facilitates the visualization of the performance of many regression functions simultaneously in a single graph. A REC graph contains one or more monotonically increasing curves (REC curves) each corresponding to a single regression model.

One can easily compare many regression functions by examining the relative position of their REC curves. The shape of the curve reveals additional information that can be used to guide modeling.

REC curves plot the error tolerance on the x-axis and the accuracy of a regression function on the y-axis. Accuracy is defined as the percentage of points predicted within the error tolerance. A good regression function provides a REC curve that climbs rapidly towards the upper-left corner of the graph, in other words, the regression function achieves high accuracy with a low error tolerance.

In regression, the residual is the analogous concept to the classification error in classification. The residual is defined as the difference between the predicted value $f(\mathrm{x})$ and actual value y of response for any point (x, y). It could be the squared error $(y - f(\mathrm{x}))^2$ or absolute deviation $| y - f(\mathrm{x}) |$ depending on the error metric employed. Residuals must be greater than a tolerance e before they are considered as errors.

The area over the REC curve (AOC) is a biased estimate of the expected error for a regression model. It is a biased estimate because it always underestimates the actual expectation. If e is calculated using the absolute deviation (AD), then the AOC is close to the mean absolute deviation (MAD). If e is based on the squared error (SE), the AOC approaches the mean squared error (MSE). The evaluation of regression models using REC curves is qualitatively invariant to the choices of error metrics and scaling of the residual. The smaller the AOC is, better the regression function will be. However, two REC curves can have equal AOC's but have different behaviors. The one who climbs faster towards the upper-left corner of the graph (in other words, the regression function that achieves higher accuracy with a low error tolerance) may be preferable. This kind of information can not be provided by the analysis of an error measure.

In order to adjust the REC curves in the REC graph, a null model is used to scale the REC graph. Reasonable regression approaches produce regression models that are better than the null model. The null model can be, for instance, the mean model: a constant function with the constant equal to the mean of the response of the training data.

An example of REC graph can be seen in Fig. 1. The number between parentheses in the figure is the AOC value for each REC curve. A regression function dominates another one if its REC curve is always above the REC curve corresponding to the other function. In the figure, the regression function dominates the null model, as should be expected.

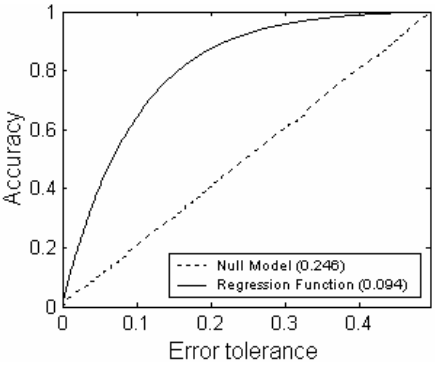


Fig. 1. Example of REC graph

3 Sigma-Point Kalman Filters

It is known that for most real-world problems, the optimal Bayesian recursion is intractable. The Extended Kalman Filter (EKF) [11] is an approximate solution that has become one of the most widely used algorithms with several applications.

The EKF approximates the state distribution by a Gaussian random variable, which is then propagated through the "first-order" linearization of the system. This linearization can introduce large errors which can compromise the accuracy or even lead to divergence of any inference system based on the EKF or that uses the EKF as a component part.

The Sigma-Point Kalman Filters (SPKF) [10], a family of filters based on derivativeless statistical linearization, achieve higher performance than EKF in many problems and are applicable to areas where EKFs can not be used.

Instead of linearizing the nonlinear function through a truncated Taylor-series expansion at a single point (usually the mean value of the random variable), SPKF rather linearize the function through a linear regression between r points, called sigma-points, drawn from the prior distribution of the random variable, and the true nonlinear functional evaluations of those points. Since this statistical approximation technique takes into account the statistical properties of the prior random variable the

resulting expected linearization error tends to be smaller than that of a truncated Taylor-series linearization.

The way that the number and the specific location of the sigma-points are chosen, as well as their corresponding regression weights, differentiate the SPKF variants from each other. The SPKF Family is composed by four algorithms: Unscented Kalman Filter (UKF), Central Difference Kalman Filter (CDKF), Square-root Unscented Kalman Filter (SR-UKF) and Square-root Central Difference Kalman Filter (SR-CDKF).

Now we will present a brief overview of the main characteristics of the Sigma-Point Kalman Filters. See [10] for more details.

3.1 The Unscented Kalman Filter

The Unscented Kalman Filter (UKF) [12] derives the location of the sigma-points as well as their corresponding weights so that the sigma-points capture the most important statistical properties of the prior random variable x. This is achieved by choosing the points according to a constraint equation which is satisfied by minimizing a cost-function, whose purpose is to incorporate statistical features of x which are desirable, but do not necessarily have to be met. The necessary statistical information captured by the UKF is the first and second order moments of $p(x)$.

3.2 The Central Difference Kalman Filter

The Central Difference Kalman Filter (CDKF) [8] is another SPKF implementation, whose formulation was derived by replacing the analytically derived first and second order derivatives in the Taylor series expansion by numerically evaluated central divided differences. The resulting set of sigma-points for the CDKF is once again a set of points deterministically drawn from the prior statistics of x. Studies [8] have shown that in practice, just as UKF, the CDKF generates estimates that are clearly superior to those calculated by an EKF.

3.3 Square-Root Forms of UKF and CDKF

SR-UKF and SR-CDKF [9] are numerically efficient square-root forms derived from UKF and CDKF respectively. Instead of calculating the matrix square-root of the state covariance at each time step (a very costly operation) in order to buid the sigma-point set, these forms propagate and update the square-root of the state covariance directly in Cholesky factored form, using linear algebra techniques. This also provides more numerical stability.

The square-root SPKFs (SR-UKF and SR-CDKF) achieve equal or slightly higher accuracy when compared to the standard SPKFs. Besides, they have lower computational cost and a consistently increased numerical stability.

4 Experimental Evaluation

Since the experiments described in [1] used just one data set and their results were only for REC demonstration, we first did tests with two well-known regression algorithms

using 25 regression problems, in order to better evaluate the REC curves as a tool for visualizing and comparing regression learning algorithms.

Then we present the results of the comparison by using REC curves of SPKFs and EKF applied to time series and finally we investigate the use of an ensemble method (stacking [18]) with the tested models, evaluating it with REC curves, as suggested by Bi and Bennett [1]. In this work, 12 time series with real-world data were used in order to try to establish a general ranking among the models tested. The names and sizes of the used time series are shown in Table 1. All data are differentiated and then the values are rescaled linearly to between 0.1 and 0.9. As null model we choose the mean model, a constant function with the constant equal to the mean of the response of the training data.

4.1 Preliminary Results with Regression

Initial experiments were carried out in order to reinforce the conclusions reached out by Bi and Bennett [1] in favor of the use of REC curves as a mean to compare regression algorithms (similarly to arguments for ROC curves in classification).

Table 1. Time series used in the experimental evaluation

Time series	Data points	Time series	Data points
A[1]	1000	Series 1[2]	96
Burstin[3]	2001	Series 2[2]	96
Darwin[3]	1400	Series 3[2]	96
Earthquake[3]	2097	Soiltemp[3]	2306
Leuven[4]	2000	Speech[3]	1020
Mackey-Glass[5]	300	Ts1[3]	1000

We have used REC curves in order to compare the performance of the Naive Bayes for Regression [7] to the performance of Model Trees [16]. Naive Bayes for Regression (NBR) uses the Naive Bayes methodology for numeric prediction tasks by modeling the probability distribution of the target value with kernel density estimators. Model Tree predictor is a state-of-the-art method for regression. Model trees are the counterpart of decision trees for regression tasks. They have the same structure as decision trees, but employ linear regression at each leaf node to make a prediction. In [7] an accuracy comparison of these two learning algorithms is presented and its results show that Model Trees outperform NBR significantly for almost all data sets tested.

[1] Data from a competition sponsored by the Santa Fe Institute. (http://www-psych.stanford.edu/%7Eandreas/Time-Series/SantaFe)
[2] Data of monthly electric load forecasting from Brazilian utilities [17].
[3] Data from the UCR Time Series Data Mining Archive [13].
[4] Data from the K.U. Leuven competition. (ftp://ftp.esat.kuleuven.ac.be/pub/sista/suykens/workshop/datacomp.dat)
[5] Numerical solution for the Mackey-Glass delay-differential equation.

The 25 regression data sets used in this study were obtained from the UCI Repository of Machine Learning Databases [2]. With 16 of the data sets the Model Tree predictor clearly outperforms NBR, as can be seen, for instance, in Fig. 2. The number between parentheses in the figure is the AOC value for each REC curve. Note that the REC curve for Model Tree covers completely the REC curve for NBR, becoming clear the superiority of the former algorithm when applied to this specific data set.

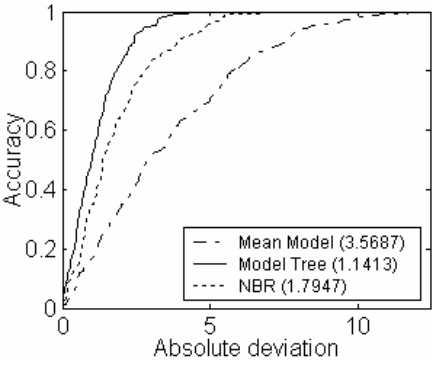


Fig. 2. REC graph used to compare the performances of NBR and Model Tree when applied to data set pwLinear

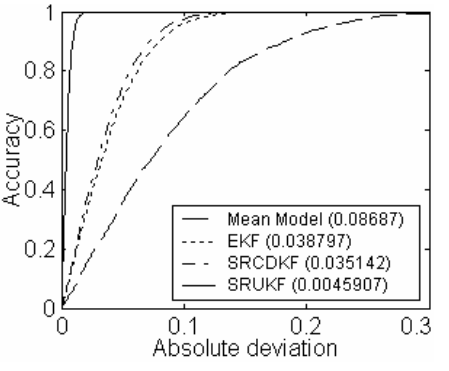


Fig. 3. EKF and SPKFs applied to Burstin time series

4.2 Comparing SPKFs by Means of REC Curves

First, we have compared UKF and CDKF with their square-root forms, SR-UKF and SR-CDKF respectively. As expected, the REC curves for UKF and for SR-UKF are very similar. This means that the difference between the performances of the models provided by UKF and SR-UKF was negligible. The same fact could be verified with the REC curves for CDKF and SR-CDKF. Therefore, because of these results and the other advantages mentioned before in Section 3, we have continued our experiments only with the square-root forms of the SPKF.

By analyzing the generated REC graphs, we could verify that, for most time series, the model provided by SR-UKF dominates the models provided by SR-CDKF and EKF, that is, the REC curve for the SR-UKF model is always above the REC curves for SR-CDKF and EKF. Therefore, the model provided by SR-UKF would be preferable. An example is shown in Fig. 3.

SR-UKF was outperformed by SR-CDKF only for the Mackey-Glass time series (Fig. 4). SR-CDKF and EKF achieved similar performances for almost all time series, as can be seen, for instance, in Fig. 5. However, the analysis of the AOC's gives a small advantage to SR-CDKF. The lower performance of EKF when compared to the others is probably caused by the non-linearity of the series. Therefore, SR-UKF consistently showed to be the best alternative to use with these series, followed by SR-CDKF and EKF, in this order. The Model Tree predictor and NBR were also tested for the prediction of the time series, but both provided poor models.

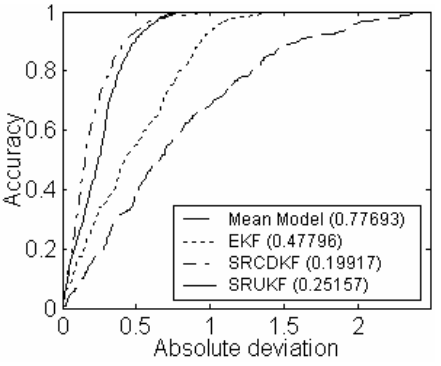


Fig. 4. EKF and SPKFs applied to Mackey-Glass time series

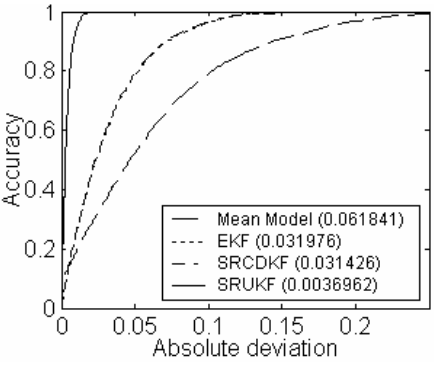


Fig. 5. EKF and SPKFs applied to Earthquake time series

4.3 Stacking of Sigma-Point Kalman Filters

Stacking [18] is an ensemble method [4] used to combine different learning algorithms. It works as follows. Suppose we have a set of different learning algorithms and a set of training examples. Each of these algorithms, called base learners, is applied to the training data in order to produce a set of hypotheses. The results computed by this set of hypotheses are combined into new instances, called meta-instances. Each of the "attributes" in the meta-instance is the output of one of the learning algorithms and the class value is the same of the original instance. Another learning algorithm, called meta-regressor (or meta-classifier, for classification), is trained and tested with the meta-instances and provides the final result of the stacking.

We have used stacking to build ensembles of SPKFs and EKF. A Model Tree predictor was chosen as a meta-regressor not only because it achieved good results in the initial experiments, but also because it is a state-of-the-art regression method and it has already been successfully used as a meta-classifier for stacking [6], outperforming all the other combining methods tested.

Table 2. Stackings built

Stackings	Base learners
Stacking 1	EKF, SR-CDKF
Stacking 2	EKF, SR-UKF
Stacking 3	SR-CDKF, SR-UKF
Stacking 4	EKF, SR-CDKF, SR-UKF

In order to determine which subset of algorithms can provide the best ensemble, we built four models by stacking: one containing the square-root SPKFs and EKF, and the others leaving one of them out. If we were testing several algorithms we could use a method to build the ensembles [3]. Table 2 shows the stackings built: Stacking 1 is composed by EKF and SR-CDKF, Stacking 2 is composed by EKF and SR-UKF, Stacking 3 is composed by SR-CDKF and SR-UKF, and Stacking 4 is composed by EKF, SR-CDKF and SR-UKF.

Table 3. AOC's of the REC curves provided for the stackings with SR-UKF as a base learner

Time series	Stacking 2	Stacking 3	Stacking 4
A	0.001366	0.001497	**0.001310**
Burstin	0.001740	**0.001613**	0.001740
Darwin	**0.013934**	0.014069	0.014052
Earthquake	0.000946	**0.000943**	0.000946
Leuven	0.005172	0.005190	**0.005142**
Mackey-Glass	0.228064	0.133420	**0.128672**
Series 1	0.001167	0.001306	**0.001111**
Series 2	0.013139	**0.012294**	0.012639
Series 3	0.000800	**0.000717**	0.000767
Soiltemp	0.000884	**0.000780**	0.000782
Speech	0.000714	0.000713	**0.000706**
Ts1	0.005010	0.005044	**0.004881**

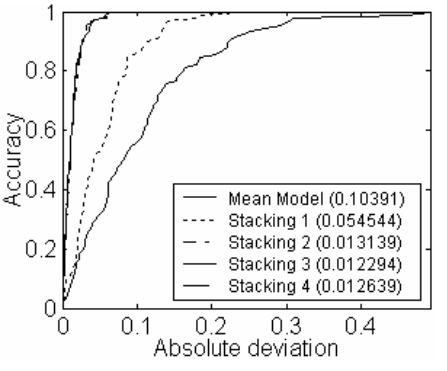


Fig. 6. Stackings applied to Series 2 time series

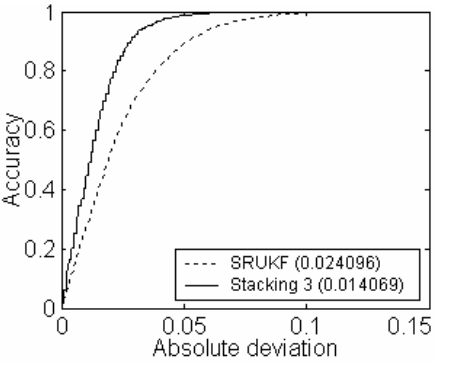


Fig. 7. SR-UKF and Stacking 3 applied to Darwin time series

The REC curves show that all stackings that have the SR-UKF as a base learner achieve similar high performances. This can be seen, for example, in Fig. 6.

Table 3 shows the AOC values of the REC curves provided for the stackings with SR-UKF as a base learner. By analyzing the values we can see that among the three stackings that contain the SR-UKF, those who have SR-CDKF as a base learner achieve a slightly better performance. Since the number of time series for which Stacking 3 achieved the best performance is almost the same number of time series for which Stacking 4 was the best, we have considered that the inclusion of EKF as a base learner does not compensate the overhead in terms of computational cost. Thus, the model chosen as the best is that provided by Stacking 3 (SR-CDKF and SR-UKF as base learners).

By comparing the best stacking model (SR-CDKF and SR-UKF as base learners and Model Tree predictor as meta-regressor) to the best individual algorithm (SR-UKF) we could verify that the stacking achieved a significantly higher performance for all time series tested. This can be clearly noted in Fig. 7.

5 Conclusions and Future Works

We have used REC curves in order to compare the SPKF family of filters (state-of-the-art time series predictors) and ensembles of them, applied to real-world time series.

The results of the experiments pointed SR-UKF as the best SPKF to use for forecasting with the series tested. Further experiments showed that a stacking composed by SR-CDKF and SR-UKF as base learners and a Model Tree predictor as meta-regressor can provide a performance statistically significantly better than that provided by the SR-UKF algorithm working individually. The REC curves showed to be very efficient in the comparison and choice of time series predictors and base learners for ensembles of them.

Currently, we are conducing tests with REC curves in order to compare Particle Filters [5], sequential Monte Carlo based methods that allows for a complete representation of the state distribution using sequential importance sampling and resampling. Since Particle Filters approximate the posterior without making any explicit assumption about its form, they can be used in general nonlinear, non-Gaussian systems. As a future work we intend to investigate further the use of ensembles with SPKFs, as well as with Particle Filters.

Acknowledgments

We would like to thank Eibe Frank for providing the source code for the NBR algorithm. The Sigma-Point Kalman Filters were obtained from the ReBEL Toolkit, under Academic License from OGI School of Science and Engineering, Rudolph van der Merwe and Eric A. Wan. The implementation of Model Trees was obtained from the WEKA System (http://www.cs.waikato.ac.nz/~ml/weka/). The authors are partially financially supported by the Brazilian Research Agency CNPq.

References

1. Bi, J., Bennett, K.P.: Regression Error Characteristic Curves. In: Proceedings of the 20th International Conference on Machine Learning (ICML), Washington, DC (2003) 43–50
2. Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases. Machine-readable data repository, University of California, Department of Information and Computer Science, Irvine, CA (2005) [http://www.ics.uci.edu/~mlearn/MLRepository.html]
3. Caruana, R., Niculescu-Mizil, A.: An Empirical Evaluation of Supervised Learning for ROC Area. In: Proceedings of the First Workshop on ROC Analysis (ROCAI) (2004) 1–8
4. Dietterich, T.G.: Machine Learning Research: Four Current Directions. The AI Magazine, 18 (1998) 97–136
5. Doucet, A., de Freitas, N., Gordon, N.: Sequential Monte-Carlo Methods in Practice. Springer-Verlag (2001)
6. Dzeroski, S., Zenko, B.: Is Combining Classifiers with Stacking Better than Selecting the Best One?. Machine Learning, 54 (2004) 255–273
7. Frank, E., Trigg, L., Holmes, G., Witten, I.H.: Naive Bayes for Regression. Machine Learning, 41 (2000) 5–25

8. Ito, K., Xiong, K.: Gaussian Filters for Nonlinear Filtering Problems. IEEE Transactions on Automatic Control, 45 (2000) 910–927
9. van der Merwe, R., Wan, E.: Efficient Derivative-Free Kalman Filters for Online Learning. In: Proceedings of the 9th European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium (2001)
10. van der Merwe, R., Wan, E.: Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models. In: Proceedings of the Workshop on Advances in Machine Learning, Montreal, Canada (2003)
11. Jazwinsky, A.: Stochastic Processes and Filtering Theory. Academic Press, New York (1970)
12. Julier, S., Uhlmann, J., Durrant-Whyte, H.: A New Approach for Filtering Nonlinear Systems. In: Proceedings of the American Control Conference (1995) 1628–1632
13. Keogh, E., Folias, T.: The UCR Time Series Data Mining Archive. University of California, Computer Science & Engineering Department, Riverside, CA (2002) [http://www.cs.ucr.edu/~eamonn/TSDMA/index.html]
14. Provost, F., Fawcett, T.: Analysis and Visualization of Classifier Performance: Comparison Under Imprecise Class and Cost Distributions. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD), AAAI Press (1997) 43–48
15. Provost, F., Fawcett, T., Kohavi, R.: The Case Against Accuracy Estimation for Comparing Classifiers. In: Proceedings of the 15th International Conference on Machine Learning (ICML), Morgan Kaufmann, San Francisco (1998) 445–453
16. Quinlan, J.R.: Learning with Continuous Classes. In: Proceedings of the 5th Australian Joint Conference on Artificial Intelligence, World Scientific, Singapore (1992) 343–348
17. Teixeira, M., Zaverucha, G.: Fuzzy Bayes and Fuzzy Markov Predictors. Journal of Intelligent and Fuzzy Systems, 13 (2003) 155–165
18. Wolpert, D.: Stacked generalization. Neural Networks, 5 (1992) 241–260

Analysis of Fast Input Selection: Application in Time Series Prediction

Jarkko Tikka, Amaury Lendasse, and Jaakko Hollmén

Helsinki University of Technology, Laboratory of Computer and Information Science, P.O. Box 5400, FI-02015 HUT, Finland
tikka@mail.cis.hut.fi
http://www.cis.hut.fi/tikka

Abstract. In time series prediction, accuracy of predictions is often the primary goal. At the same time, however, it would be very desirable if we could give interpretation to the system under study. For this goal, we have devised a fast input selection algorithm to choose a parsimonious, or sparse set of input variables. The method is an algorithm in the spirit of backward selection used in conjunction with the resampling procedure. In this paper, our strategy is to select a sparse set of inputs using linear models and after that the selected inputs are also used in the non-linear prediction based on multi-layer perceptron networks. We compare the prediction accuracy of our parsimonious non-linear models with the linear models and the regularized non-linear perceptron networks. Furthermore, we quantify the importance of the individual input variables in the non-linear models using the partial derivatives. The experiments in a problem of electricity load prediction demonstrate that the fast input selection method yields accurate and parsimonious prediction models giving insight to the original problem.

1 Introduction

Time series analysis is an important problem in natural and engineering sciences, both from the viewpoint of prediction and understanding of the behavior of the systems under study. There are numerous applications of time series analysis scattered in the published literature of econometrics, system identification, chemistry, statistics, pattern recognition, and neural networks [1]. It would be very appealing to be able to predict the behavior of the time series accurately, and at the same time to give insight to the system itself. Our target is to estimate time series prediction models that are both accurate and interpretable. By interpretability we mean that the models contain only a relatively small subset of input variables for the prediction. This gives emphasis to what is important in the prediction of system behavior. These kind of parsimonious, or sparse time series models are the focus of the paper. Inputs of the sparse models are selected from a large set of autoregressive input variables for a given past horizon. This approach tries to circumvent the problems of the high-dimensional input space, i.e. curse of dimensionality.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 161–170, 2006.

In the estimation of the sparse time series models, we rely on sparse regression techniques [2] and a backward selection strategy. In addition, resampling procedures [3] are used to take into account the inherent uncertainty of the finite data samples used in the estimation procedure. One of the main goals of the proposed method is to offer a fast and reliable method for input selection. In the first phase of the methodology, the linear model that is built is forced to be sparse. That is, we do not select the most accurate model, rather we select a compromise between sparsity and accuracy. In the second phase, the non-linear prediction model is constructed using the selected sparse set of inputs.

In this paper, we present an analysis of our previously published input selection method used for the problem of long-term time series prediction [4]. It is noteworthy, however, that the method is generally applicable to input selection problems. Our interest is to apply the method to input selection within time series prediction.

The rest of the article is organized as follows: Sect. 2 introduces relevant background in the time series prediction. Section 3 reviews our fast input selection procedure in the context of linear prediction models. Section 4 focuses on non-linear prediction models, i.e. multi-layer perceptron (MLP) networks, in which the selected variables are finally used. Also, sensitivity analysis of the MLP networks is presented. Section 5 presents the empirical experiments on which the findings are based on. Summary and Conclusions are presented in Sect. 6.

2 Time Series Prediction

In a time series prediction problem, future values of time series are predicted using the previous values. The previous and future values of time series are referred to inputs and outputs of the prediction model, respectively. One-step-ahead prediction is needed in general and it is called short-term prediction. If multi-step-ahead predictions are needed, it is known as long-term prediction.

Unlike short-term prediction, long-term prediction faces typically growing amount of uncertainties arising from various sources. For instance, an accumulation of errors and lack of information make the prediction more difficult. In the case of long-term prediction, there are several strategies to build prediction models. The direct and the recursive prediction are shortly described next.

2.1 Recursive Prediction Strategy

In the case of multi-step-ahead prediction, the recursive strategy uses the predicted values as known data to predict next ones. First, a one-step-ahead prediction is done $\hat{y}_t = f_1(y_{t-1}, y_{t-2}, ..., y_{t-l})$, where $y_{t-i}, i = 1, \ldots, l$ are the inputs. It is also possible to use external variables as inputs, but they are not considered here in order to simplify the notation. After that, the same model is used to predict two-step-ahead $\hat{y}_{t+1} = f_1(\hat{y}_t, y_{t-1}, y_{t-2}, \ldots, y_{t-l+1})$, where the predicted value of $\hat{y}_t$ is used instead of the true value, which is unknown. Then, the k-step-ahead predictions $y_{t+k-1}, k \geq 3$ are obtained iteratively. In the prediction of kth step, $l - k$ observed values and k predicted values are used as the inputs in the

case of $k < l$. When $k \geq l$, all the inputs are the predicted values. The use of the predicted values as inputs may deteriorate the accuracy of the prediction.

2.2 Direct Prediction Strategy

In the direct strategy, the model $\hat{y}_{t+k-1} = f_k(y_{t-1}, y_{t-2}, \ldots, y_{t-l})$ is used for k-step-ahead prediction. The predicted values are not used as inputs at all in this approach, thus the errors in the predicted values are not accumulated into the next predictions. When all the values from y_t to y_{t+k-1} need to be predicted, k different models must be constructed. This increases the computational complexity, but more accurate results are achieved using the direct than the recursive strategy as shown in [4] and [5]. We only apply the direct strategy in this paper.

3 Fast Input Selection

Consider the situation that there are N measurements available from an output variable y and input variables $x_i, i = i, \ldots, l$. In the regression problems the usual task is to estimate the values of the output y using the inputs x_i. If the dependency is assumed to be linear it can be written mathematically

$$y_j = \sum_{i=1}^{l} \beta_i x_{j,i} + \varepsilon_j, \quad j = 1, \ldots, N \ . \tag{1}$$

The errors ε_j are assumed to be independently normally distributed with zero mean and common variance. All the variables are assumed to have zero mean and unit variance, thus the constant term is dropped out from the model. The ordinary least squares (OLS) estimates of the regression coefficients $\hat{\beta}_i$ are obtained by minimizing the mean squared error (MSE) [2].

The OLS estimates are not typically satisfactory [2]. Firstly, the generalization ability of the model may be improved by shrinking some coefficients toward zero or setting them exactly to zero. Secondly, if the number of inputs is large interpretation of the model might be difficult. Understanding or interpretability of the underlying process can be increased by selecting the subset of inputs which have the strongest effect in the prediction. Many approaches to input selection are presented in [2] and [6].

We propose an efficient input selection procedure in [4]. The algorithm is based on the bootstrap resampling procedure and it requires separate training and validation sets. However, it is straightforward to use other resampling procedures [3], e.g. k-fold cross-validation, instead of bootstrap.

The input selection procedure starts by estimating the linear model using all the available inputs. The sampling distributions of OLS estimates $\hat{\beta}_i$ and the standard deviation s_{tr} of the training MSEs are estimated using M times k-fold cross-validation. We have Mk different training sets and, thus, Mk estimates for the each coefficient β_i, which formulate the sampling distribution. In addition, we have Mk estimates for both training and validation MSE.

The median m_{β_i} is calculated from Mk estimates $\hat{\beta}_i$. The median is used as the location parameter for the distribution, since it is a reasonable estimate for skewed distributions and distributions with outliers. The width of the distribution of $\hat{\beta}_i$ is evaluated using the difference $\Delta_{\beta_i} = \hat{\beta}_i^{high} - \hat{\beta}_i^{low}$, where $\hat{\beta}_i^{high}$ is $Mk(1-q)$th and $\hat{\beta}_i^{low}$ is Mkqth value in the ordered list of the Mk estimates $\hat{\beta}_i$ [3] and q can be set, e.g., $q = 0.165$. With this choice of q, the difference Δ_{β_i} is twice as large as the standard deviation in the case of the normal distribution. The difference Δ_{β_i} describes well the width of both asymmetric and symmetric distributions.

The next step is to delete the least significant input variable. The ratio $|m_{\beta_i}|/\Delta_{\beta_i}$ is used as a measure of significance of the corresponding input variable. The input with the smallest ratio is pruned from the set of inputs. After that, the cross-validation procedure using the remaining inputs and pruning is repeated as long as there are variables left in the set of inputs.

The previous procedure removes inputs sequentially from the set of possible inputs. In the end, we have l different models. The purpose is to select a model which is as sparse as possible, but it still has comparable prediction accuracy. The initial model is selected based on the minimum validation error E_v^{min}. The final model is the least complex model whose validation error is under the threshold $E_v^{min} + s_{tr}^{min}$, where s_{tr}^{min} is the standard deviation of training MSE of the model having the minimum validation error. This means that we also include our uncertainty in the training phase into the selection of final model. The algorithmic details of the proposed method are presented in [4].

Advantage of the described algorithm is the ranking of the inputs according to their explanatory power. The pruning starts from the least significant inputs and the resulting model includes only a few most significant ones. This might be useful information for interpretation of the underlying process. Also, the computational complexity of the proposed algorithm is linear $\mathcal{O}(l)$ with respect to the number of available inputs l. Therefore, it is applicable in the case of large number of inputs.

4 Non-linear Modeling Using MLP Networks

Although the linear models are easy to interpret and fast to calculate they are not accurate enough in some problems. The dependencies between the variables are described better using a non-liner model. However, many non-linear models are black-box models and interpretation is almost impossible. Our proposal is to use the selected inputs also in the non-linear model. Goals of this approach are to avoid the curse of dimensionality, over-parameterization, and overfitting in the non-linear modeling phase. In addition, the interpretability of the non-linear model increases, since only a subset of inputs is included to the model.

MLP networks are used in the non-linear modeling phase

$$\hat{y} = \mu + \sum_{j=1}^{p} \alpha_j \tanh(\sum_{i=1}^{l} w_{ij} x_i + b_j) \ , \tag{2}$$

where p is the number of neurons in the hidden layer, $\hat{y}$ is the estimate of the output y and μ, α_j, and w_{ij} are the weights of the network. It is known that only one hidden layer is required to approximate any continuous function if the number of connection weights is sufficient [7].

The number of connection weights is controlled by the number of neurons in the hidden layer. The selection of number of neurons is based on k-fold cross-validation. The optimal connection weights minimize MSE in the validation sets. Another option is to set the number of neurons to be large enough and to use weight decay (WD) to reduce the effective number of connection weights [8]. When WD is applied the cost function is

$$E = \frac{1}{N}\left(\sum_{j=1}^{N}(y_j - \hat{y}_j)^2 + \lambda \boldsymbol{\theta}^T \boldsymbol{\theta}\right) \quad , \tag{3}$$

where $\boldsymbol{\theta}$ is the vector containing all the parameters of the network and λ is the weight decay parameter. A proper value for λ can be selected by cross-validation. In WD, the values of weights are shrunk toward zero, but they are not set exactly to zero. So, it is very likely that WD does not perform input selection.

4.1 Sensitivity Analysis

It may not be enough that the most relevant inputs are found. In many applications, it is important to evaluate the way inputs contribute to explanation or prediction of the output.

The contribution of each input to the output can be evaluated using partial derivatives of the MLP network [9]. Partial derivatives (PAD) method gives two results. First, a profile of the output variations for a small changes of each input. Second, classification of the inputs in increasing order of relative importance. It is found that the PAD method gives stable results [9].

The partial derivative of MLP network (2) with respect to the input x_i is

$$d_i = \frac{\partial \hat{y}}{\partial x_i} = \sum_{j=1}^{p} \alpha_j (1 - I_j^2) w_{ij} \quad , \tag{4}$$

where $I_j = \tanh(\sum_{i=1}^{l} w_{ij} x_i + b_j)$. A set of graphs of the partial derivatives versus each corresponding input can be plotted. The graphs show the influence of the inputs on the output.

The sensitivity of the MLP output for the data set with respect to an input is calculated

$$\mathrm{SSD}_i = \sum_{j=1}^{N} d_{i,j}^2, \qquad \mathrm{SSD}_i \leftarrow \frac{\mathrm{SSD}_i}{\sum_{i=1}^{l} \mathrm{SSD}_i} \quad . \tag{5}$$

Sum of squared derivatives (SSD) value is achieved for each input. SSD_i is the sum over all the observations. In the end, the SSD_i values are scaled such that $\sum_{i=1}^{l} \mathrm{SSD}_i = 1$. The input having the highest SSD_i value influences most on the output. The ranking based on SSD values can be compared to the ranking obtained using the input selection method presented in Sect. 3.

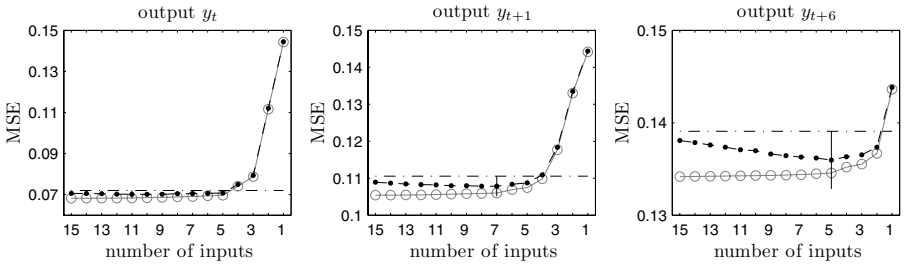


Fig. 1. Illustration of input selection, training error (*gray line*) and validation error (*black line*) as a function of the number of inputs in the linear model. The vertical line marks the minimum validation error and the horizontal dash-dotted line represents the threshold, which is used in the selection of the final model.

5 Experiments

The two-phase modeling strategy described in the previous sections is applied to time series prediction. The data set used is the Poland electricity load time series[1] [5]. It contains daily measurements from the electricity load in Poland in the 1990's. The data set has 1400 observations in the training set and 201 observations in the test set. The training and test sets are not consecutive. The objective is to predict the electricity load one- (y_t), two- (y_{t+1}), and seven-day-ahead (y_{t+6}). We use direct prediction approach, i.e. we have to construct own model for each case. The data is normalized to zero mean and unit variance before the analysis.

5.1 Phase I: Input Selection

The maximum number of inputs is set to be $l = 15$, i.e. the available inputs are $y_{t-l}, l = 1, \ldots, 15$ in each of the three prediction cases. In the input selection, 10-fold cross-validation repeated $M = 100$ times is used. This choice produces 1000 estimates for the coefficients β_i, which is considered to be large enough for reliably estimating the distribution of the parameters in the linear model.

Figure 1 illustrates the input selection procedure. In all the three cases, it is notable that the validation error starts to increase only in the end. Almost all the inputs are pruned from the model then. If the final model had been selected according to the minimum validation error the number of inputs would have been 11, 7, and 5 in the case of one-, two, and seven-day-ahead prediction, respectively. However, even more parsimonious models are achieved when the thresholding is used. The final numbers of inputs are 5, 5, and 2 and the validation errors do not increase significantly.

In Fig. 2, the selected inputs for all the three models are visualized. The smaller the white number is in the selected inputs (in the black rectangles) the more important the corresponding input is in the prediction. In other words, the

[1] http://www.cis.hut.fi/projects/tsp/?page=Timeseries

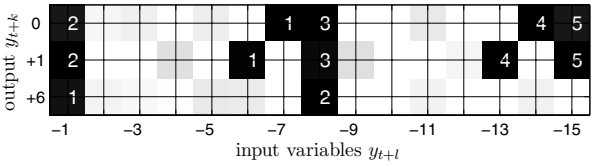


Fig. 2. The final models. The outputs $y_{t+k}, k = 0, +1, +6$ are in the vertical axis and the possible inputs $y_{t-l}, l = 1, \ldots, 15$ are in the horizontal axis. The selected inputs are denoted by black rectangles on each row and the white numbers indicate the ranking of the inputs according to the importance in the prediction.

number 1 indicates that the input was the last to prune from the model. For instance, the upper row shows that the one-day-ahead model has 5 inputs, which are $y_{t-7}, y_{t-1}, y_{t-8}, y_{t-14}$, and y_{t-15} in the decreasing order of importance. The model has nice interpretation, since the inputs correspond to values of 7, 1, 8, 14, and 15 days before the predicted value. It is plausible that the two most important inputs are the values of one week and one day before.

5.2 Phase II: Non-linear Modeling

Based on the results of the input selection, non-linear models are trained. Three MLP networks are constructed for each output: i) MLP using the selected inputs without weight decay, number of neurons (the maximum were 15) in the hidden layer selected by 10-fold cross-validation repeated five times, ii) MLP using the selected inputs with weight decay, number of neurons in the hidden layer was 20, and iii) MLP with all the inputs with weight decay, number of neurons in the hidden layer was 20. In the cases ii) and iii) MLPs are evaluated using 30 values of the regularization parameter λ, which are logarithmically equally spaced in the range $\lambda \in [10^{-4}, 10^{3}]$. The optimal value of λ is selected using 10-fold cross-validation repeated five times to increase the reliability of the results.

All the networks are trained using the Levenberg-Marquardt optimization method by back-propagating the error gradients [8]. Ten different initializations are used in the training of the networks in order to avoid local minima. The training errors were monotonically decreasing as a function of increasing complexity. In Fig. 3, the validation errors are shown as a function of λ for the cases ii) (*left*) and iii) (*right*). It is notable that the minimum errors are roughly on the same level, but the curve is flater in the left figure. Thus, a sparser grid for λ could be used, which would reduce the computational burden. In the case i), the minimum validation error is obtained using $p = 6$, $p = 6$, and $p = 7$ neurons in the hidden layer for the outputs y_t, y_{t+1}, and y_{t+6}, respectively.

The prediction accuracy of the final models were evaluated using the test set, which is not used at all in the training and selection of the final models. Thousand bootstrap replications were drawn from the test set and MSE was calculated for each replication. The means and the standard deviations of MSE for each model are reported in Table 1. The sparse linear models are equally accurate as the full linear models, which indicates that the selected inputs are the most informative.

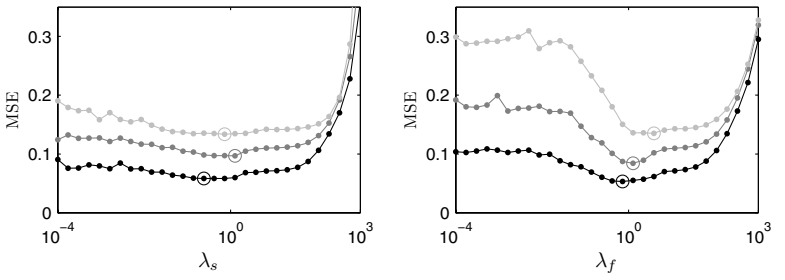


Fig. 3. Validation errors as a function of λ for one-day-ahead (*black line*), two-day-ahead (*dark gray line*), and seven-day-ahead (*light gray line*) prediction in the case of selected inputs (*left*) and all the inputs (*right*). Circles mark the minimums.

Table 1. MSEs and standard deviations of MSEs for the test set calculated using the bootstrap resampling procedure. n is the number of inputs, p is the number of neurons in the hidden layer, and λ is the regularization parameter.

	full linear linear $n = 15$	sparse linear model	MLP n-p-1 no WD	MLP n-20-1 with WD	MLP 15-20-1 with WD
1-day-ahead	0.054 (0.012)	0.055 (0.012) $n = 5$	0.038 (0.010) $n = 5$, $p = 6$	0.040 (0.010) $n = 5$, $\lambda = 0.24$	0.038 (0.010) $\lambda = 0.73$
2-day-ahead	0.085 (0.019)	0.086 (0.018) $n = 5$	0.074 (0.018) $n = 5$, $p = 6$	0.077 (0.017) $n = 5$, $\lambda = 1.27$	0.079 (0.016) $\lambda = 1.27$
7-day-ahead	0.118 (0.023)	0.116 (0.023) $n = 2$	0.116 (0.022) $n = 2$, $p = 7$	0.117 (0.023) $n = 2$, $\lambda = 0.73$	0.114 (0.023) $\lambda = 3.86$

In the cases of one- and two-day-ahead prediction, MLP with selected inputs without WD is the most accurate. It decreases MSE 30% and 13% compared to the full linear model in one- and two-day-ahead predictions, respectively. Also, it is slightly better than MLP with all the inputs. For seven-day-ahead prediction, MLP with all the inputs and WD has the lowest prediction error, although the errors of the other methods are nearly the same.

In Fig. 4, the relative importances of the inputs calculated by (5) are shown. The importances are averages over thousand bootstrap replications of the test set. In the case of one-day-ahead prediction and 5-6-1 MLP, the inputs are ranked in the order of decreasing importance as follows: y_{t-1}, y_{t-7}, y_{t-15}, y_{t-8}, and y_{t-14}. The ranking is nearly the same as with the linear models, see Fig 2. In 15-20-1 MLP, the five most important inputs in the order of decreasing importance are y_{t-1}, y_{t-7}, y_{t-2}, y_{t-8}, y_{t-15}. Four of them are same as obtained with the linear models. Also, in the cases of two- and seven-day-ahead prediction

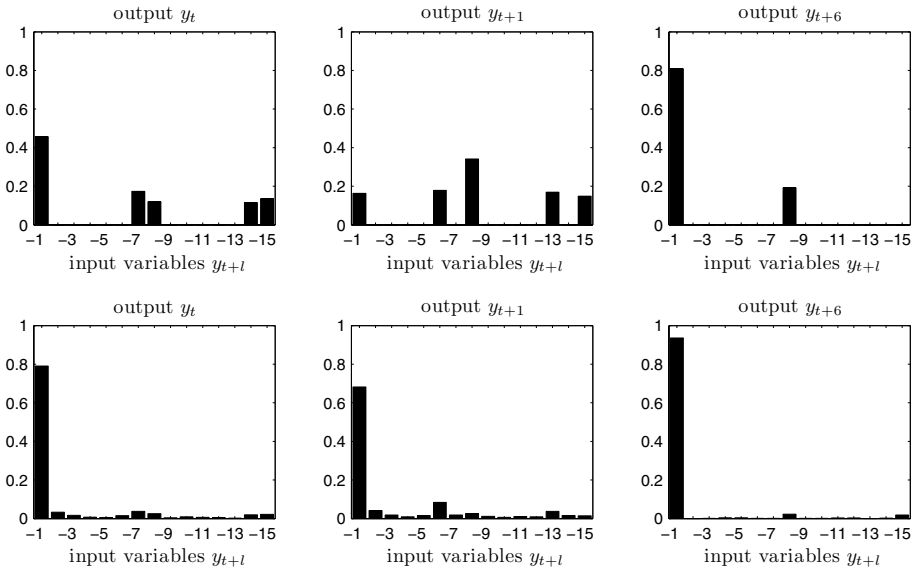


Fig. 4. Relative importances of the input variables $y_{t-l}, l = 1, \ldots, 15$ in 5-6-1 MLP network without WD (*above*), in 15-20-1 MLP with WD (*below*)

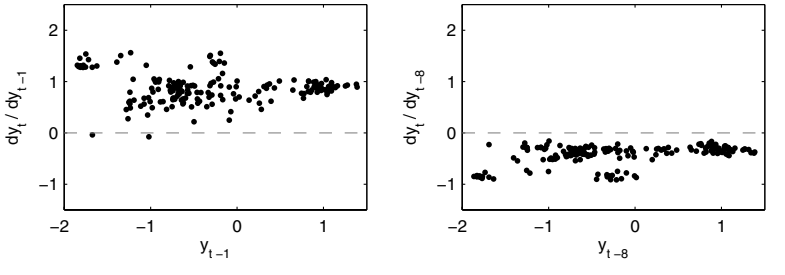


Fig. 5. The profiles of the inputs y_{t-1} (*left*) and y_{t-8} (*right*) in 5-6-1 MLP in one-day-ahead prediction

the relative importances of inputs in the MLP networks are nearly the same as with the linear model.

The contribution of the inputs y_{t-1} and y_{t-8} in the prediction of y_t with 5-6-1 MLP are shown in Fig. 5. The shown result is for the test set. The values of $\partial y_t / \partial y_{t-1}$ are positive, which means that y_t tends to increase while y_{t-1} increases. Although the relative importance of y_{t-8} is notably smaller than y_{t-1}, still the partial derivatives $\partial y_t / \partial y_{t-8}$ are clearly non-zero and negative. Thus, y_{t-8} has also contribution in the prediction. While y_{t-8} increases the output y_t tends to decrease.

6 Summary and Conclusions

A backward selection type algorithm with resampling for input selection in the context of time series prediction was presented. Experiments in an electricity load prediction demonstrated that the two phase strategy using input selection in a linear prediction model and subsequent non-linear modeling using MLP yields accurate prediction. In addition, this strategy was competitive to MLP network with all the inputs and a large number of neurons in the hidden layer trained with weight decay. The importance of inputs obtained using the linear models reflected also very well the importance of inputs in the non-linear models. The advantage of presented approach is sparsity in terms of the number of inputs and parameters in the final network. Sparsity of inputs makes the models more interpretable. A low number of parameters allows fast training of the networks and makes the models less prone to overfitting.

References

1. Weigend, A.S., Gershenfeld, N.A. (eds.): Time Series Prediction: Forecasting the Future and Understanding the Past. Addison-Wesley (1994)
2. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning – Data Mining, Inference and Prediction. Springer Series in Statistics. (2001)
3. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman & Hall/CRC (1993)
4. Tikka, J., Hollmén, J., Lendasse, A.: Input Selection for Long-Term Prediction of Time-Series. In: Proceedings of the 8th International Work-Conference on Artificial Neural Networks (IWANN 2005). 1002–1009
5. Ji, Y., Hao, J., Reyhani, N., Lendasse, A.: Direct and Recursive Prediction of Time Series Using Mutual Information Selection. In: Proceedings of the 8th International Work-Conference on Artificial Neural Networks (IWANN 2005). 1010–1017
6. Ljung, L.: System Identification – Theory for the User. 2nd Edition. Prentice–Hall. (1999)
7. Hornik, K., Stinchcombe, M., White, H.: Multilayer Feedforward Networks are Universal Approximators. Neural Networks. **2** (1989) 359–366
8. Bishop, C.: Neural Networks in Pattern Recognition. Oxford Press (1996)
9. Gevrey, M., Dimopoulos, I., Lek, S.: Review and Comparison of Methods to Study the Contribution of Variables in Artificial Neural Network Models. Ecological Modelling **160** (2003) 249–264

A Linguistic Approach to a Human-Consistent Summarization of Time Series Using a SOM Learned with a LVQ-Type Algorithm

Janusz Kacprzyk[1], Anna Wilbik[1], and Sławomir Zadrożny[1,2]

[1] Systems Research Institute, Polish Academy of Sciences
ul. Newelska 6, 01-447 Warsaw, Poland
[2] Warsaw Information Technology (WIT)
ul. Newelska 6, 01-447 Warsaw, Poland
{kacprzyk, wilbik, zadrozny}@ibspan.waw.pl

Abstract. The purpose of this paper is to propose a new, human consistent way to capture the very essence of a dynamic behavior of some sequences of numerical data. Instead of using traditional, notably statistical type analyses, we propose the use of fuzzy logic based linguistic summaries of data(bases) in the sense of Yager, later developed by Kacprzyk and Yager, and Kacprzyk, Yager and Zadrożny. Our main interest is in the summarization of trends characterized by: dynamics of change, duration and variability. To define the dynamic of change of the time series we propose to use for a preprocessing of data a SOM (self-organizing maps) learned with a LVQ (Learning Vector Quantization) algorithm, and then our approach for linguistic summaries of trends.

1 Introduction

We consider time series meant as sequences of numeric data, at equally spaced time moments. We try to discover trends, and also some other characteristic features of time series, notably their variability, periods of growth/ decrease/stability, etc. We try to derive some human consistent characterizations of them that are, in general, expressed in a (quasi)natural language that is obviously the only fully natural means of human articulation and communication. The main tool employed is that of a linguistic summary meant as a concise, human-consistent description of a data set. The very concept has been introduced by Yager [14] and further developed by Kacprzyk and Yager [2], and Kacprzyk, Yager and Zadrożny [3]. In this approach the content of a database is summarized via a natural language like expression, semantics of which is provided in the framework of the Zadeh's calculus of the linguistically quantified propositions [15].

Here we apply linguistic summaries to a specific type of data, namely time series, i.e. certain real valued functions of time. The summaries we propose refer to trends identified here with straight line segments of a piece-wise linear approximation of time series. Thus, the first step is a preprocessing, that is the construction of such an approximation. Our idea is to use for this purpose self-organizing maps (SOMs) learned using the LVQ algorithm as a clustering tool

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 171–180, 2006.

or classifier to derive classes of trends. The use of the LQV algorithm makes it possible to obtain supervised learning that in our case means that we classify to a predefined number of classes (labels describing the dynamics of change, in fact inclination, as in Figure 1). Clearly, there may be more or less linguistic terms assumed but, as it is well known, the so called Miller's magic number 7 ± 2 is a good choice.

Basically the summaries proposed by Yager are interpreted in terms of the number or proportion of elements possessing a certain property. In the framework considered here a summary might look like: "Most of the trends are short" or in a more sophisticated form: "Most long trends are increasing". Such expressions are easily interpreted using Zadeh's calculus of the linguistically quantified propositions. The most important element of this interpretation is a linguistic quantifier exemplified by "most". In Zadeh's approach it is interpreted in terms of a proportion of elements possessing a certain property (e.g., a length of a trend) among all the elements considered (e.g., all trends). In [9] we proposed to use Yager's linguistic summaries, interpreted in the framework of Zadeh's calculus, for the time series.

Another type of summaries we propose here do not use the linguistic quantifier based aggregation over the number of trends but over the time instants they take altogether. For example, an interesting summary may take the following form: "Trends taking most of time are increasing" or "Increasing trends taking most of the time are of a low variability".

In this paper we combine the use of the SOMs with the LQV function used for a supervised learning, with the use of Zadeh's calculus of linguistically quantified propositions to derive linguistic summaries of trends. Clearly, the former is basically meant as a tool for preprocessing, while the latter is more relevant for the approach proposed in this paper.

2 Characterization of Time Series

Time series in a sequence of data items, for simplicity we assume scalars, measured typically at uniformly spaced time moments. In our approach, a time series $\{(x_t, t = 0, 1, \dots)\}$ is divided into a fixed-size vectors by a time window of a length p time units. These segments, are characterized with the three following features:

- dynamics of change
- variability
- duration

In what follows we will briefly discuss these factors.

2.1 Dynamics of Change

Under the term *dynamics of change* we understand the speed of changes. It can be described by the slope of a line representing the trend, understood as

characteristic feature of consecutive values of time series over some time span (eg. the trend is quickly increasing or weakly decreasing, etc.).

The dynamic of change can be expressed as a real number $\langle -90, 90 \rangle$, an angle in degrees between the line representing the trend and horizontal line. However a usage of such a scale of real numbers directly while describing trends, may be beyond human comprehension and impractical. The user may construct a scale of linguistic terms corresponding to various directions of a trend line as, e.g.:

- quickly decreasing
- decreasing
- slowly decreasing
- constant
- slowly increasing
- increasing
- quickly increasing

They represent, on the one hand, a human-consistent granules of trend line inclination, from - 90 to + 90, (cf. Miller's magic number 7 ± 2). On the other hand, they may be viewed as prototypes of trends that are comprehensible by a human user.

Figure 1 illustrates the lines corresponding to the particular linguistic terms.

We propose to use self-organizing maps (SOMs), introduced in 1982 by Kohonen [10], as a clustering or classifying tool (cf. [1,12,13]). The vector quantinization property of SOMs may be used to perform clustering of time series vectors of a constant size. The self-organizing map will associate a vector with one of the groups, in other words classify it to one of the classes.

We will briefly describe the SOMs just to introduce the terminology and notation to be used. The goal of self-organizing maps is to convert a complex high-dimensional input signal into a simpler low-dimensional discrete map. The neurons are connected with one another, usually in form of one-dimensional chain or two-dimensional lattice. Learning of the SOM is an iterative process associated with the time points $t = 1, 2, \ldots$. During the learning stage, the user presents to the network some prototypes, e.g. given by the expert.

In each step, distances from the weight vectors of the current map and a randomly chosen input vector (prototype) are calculated and the best matching unit (a so called winning node) is found. The distance can be computed according to any metric, very often, also here the Euclidean distance is used. After finding the best matching unit, its weight vector is updated so that the best matching unit is moved closer to the current input vector. The topological neighbors of the best matching unit will be also updated. This adaptation procedure stretches the best matching unit and its topological neighbors towards the input sample vector. The SOM update rule for the weight vector of the unit i in the neighborhood of the best matching unit is

$$m_i(t+1) = m_i(t) + h_{c(t),i}(t)[x(t) - m_i(t)]; \qquad \forall i \in [1 \ldots n] \tag{1}$$

where $x(t)$ is the input vector randomly drawn from the input data set at time t; n is the number of neurons, $h_{c(t),i}(t)$ is the neighborhood kernel around the

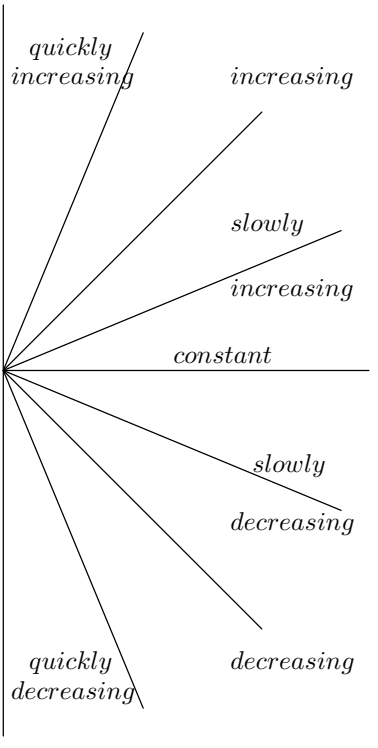


Fig. 1. A visual representation of trend line prototypes defining dynamics of change

winner unit c. The neighborhood kernel is a non-increasing function of time and of the distance of unit i from the winner unit c. It defines the region of influence that the input sample has on the SOM and often is defined as $h_{c(t),i}(t) = \alpha(t)exp\left(-||r_i - r_c||^2/2\sigma^2(t)\right)$, where $0 < \alpha < 1$ is the learning rate, monotonically decreasing, r_i and r_c are vectorial locations on the grid and $\sigma(t)$ corresponds to the width of the neighborhood function, also monotonically decreasing.

At the end of the learning stage, each cluster, or better to say a class, is associated with a region of neurons. Afterwards, SOM may cluster the real data.

Generally, SOM employs an unsupervised learning process. Nevertheless, the learning algorithm can be easily modified to represent supervised learning (cf. [10,11]) – cf. the LVQ methods, a class of algorithms, such as LVQ1, LVQ2, LVQ3. We assume that we know the class, defined by the model vector $m_i(t)$, that each of the sample vectors or prototypes $x(t)$ belong to. In our case the classes are labels describing the dynamics of change, as presented in Figure 1. Clearly, there may be more or less linguistic terms assumed but, as it is well known, the so called Miller's magic number 7 ± 2 is a good choice.

In the basic LVQ1 algorithm the weights of connections are changed as follows:

$$m_i(t+1) = m_i(t) + \alpha(t)s(t)\delta_{ci}[x(t) - m_i(t)], \tag{2}$$

where $s(t) = 1$ if x and m_i belong to the same class and $s(t) = -1$ if x and m_i belong to different classes. $\alpha \in (0, 1)$ is the learning rate, and δ_{ci} is the Kronecker delta.

We combine LVQ and SOM in a very simple way, as presented in [10]. We consider the basic learning equation (1) of unsupervised SOM and assume that if $x(t)$ and $m_i(t)$ belong to the same class, then $h_{c(t),i}(t)$ is positive, otherwise it is negative.

2.2 Variability

Variability refers to how "spread out" (in the sense of values taken on) a group of data is. There are five frequently used statistical measures of variability:

- the range (maximum - minimum). Although this range is computationally the easiest measure of variability, it is not widely used as it is only based on two extreme data points. This make it very vulnerable to outliers and therefore may not adequately describe real variability.
- the interquartile range (IQR) calculated as the third quartile (the third quartile is the 75th percentile) minus the first quartile (the first quartile is the 25th percentile) that may be interpreted as representing the middle 50% of the data. It is resistant to outliers and is computationally as easy as the range.
- the variance is calculated as $1/n \sum_i (x_i - \bar{x})^2$, where $\bar{x}$ is the mean value.
- the standard deviation – a square root of the variance. Both the variance and the standard deviation are affected by extreme values.
- the mean absolute deviation (MAD), calculated as $1/n \sum_i |x_i - \bar{x}|$. While it has a natural intuitive definition as the "mean deviation from the mean", the introduction of the absolute value makes analytical calculations using this statistics much more complicated.

We propose to measure the variability of a subseries in the time window as the distance of the data points covered by this subseries from the points describing standard trend hidden under a certain label.

We will treat variability as a linguistic variable. We map a single value characterizing the variability of a trend in the time window. Then we will say that a given trend has, e.g., "low variability to a degree 0.8", if $\mu_{low\ variability}(v) = 0.8$, where $\mu_{low\ variability}$ is the membership function of a fuzzy set representing the linguistic term "low variability" that is a best match for the variability characterizing the trend under consideration.

2.3 Duration

Duration describes the length of a single trend found by the SOM. We aggregate the neighboring trends, if they were associated with the same cluster by the SOM. The variability of aggregated longer trend is equal to the arithmetic mean of variabilities of aggregated trends.

Again we will treat duration as a linguistic variable. An example of its linguistic labels is "long trend" defined as a fuzzy set, whose membership function

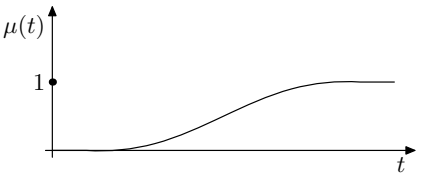


Fig. 2. Example of membership function describing the term "long" concerning the trend duration

might be assumed as in Figure 2, where OX is the axis of time measured with units that are used in the time series data under consideration.

3 Linguistic Summaries

A linguistic summary, as presented in [7,8] is meant as a natural language like sentence that subsumes the very essence of a set of data. This set is assumed to be numeric and is usually large, not comprehensible in its original form by the human being. In Yager's approach (cf. Yager [14], Kacprzyk and Yager [2], and Kacprzyk, Yager and Zadrożny [3]) we assume that:

- $Y = \{y_1, \ldots, y_n\}$ is a set of objects (records) in a database, e.g., the set of workers;
- $A = \{A_1, \ldots, A_m\}$ is a set of attributes characterizing objects from Y, e.g., salary, age, etc. in a database of workers, and $A_j(y_i)$ denotes a value of attribute A_j for object y_i.

A linguistic summary of a data set D consists of:

- a summarizer S, i.e. an attribute together with a linguistic value (fuzzy predicate) defined on the domain of attribute A_j (e.g. "low" for attribute "salary");
- a quantity in agreement Q, i.e. a linguistic quantifier (e.g. most);
- truth (validity) $\mathcal{T}$ of the summary, i.e. a number from the interval $[0, 1]$ assessing the truth (validity) of the summary (e.g. 0.7); usually, only summaries with a high value of $\mathcal{T}$ are interesting;
- optionally, a qualifier R, i.e. another attribute together with a linguistic value (fuzzy predicate) defined on the domain of attribute A_k determining a (fuzzy subset) of Y (e.g. "young" for attribute "age").

We will often for brevity identify summarizers and qualifiers with the linguistic terms they contain. In particular we will refer to the membership function μ_P or μ_R of the summarizer or qualifier to be meant as the membership functions of respective linguistic terms.

Thus, a linguistic summary may be exemplified by

$$\mathcal{T}(\text{most of employees earn low salary}) = 0.7 \tag{3}$$

A richer linguistic summary may include a qualifier (e.g. young) as, e.g.,

$$\mathcal{T}(\text{most of young employees earn low salary}) = 0.7 \tag{4}$$

Thus, basically, the core of a linguistic summary is a *linguistically quantified proposition* in the sense of Zadeh [15]. A linguistically quantified proposition, corresponding to (3) may be written as

$$Qy\text{'s are } S \tag{5}$$

and the one corresponding to (4) may be written as

$$QRy\text{'s are } S \tag{6}$$

Then, the component of a linguistic summary, $\mathcal{T}$, i.e., its truth (validity), directly corresponds to the truth value of (5) or (6). This may be calculated by using either original Zadeh's calculus of linguistically quantified propositions (cf. [15]), or other interpretations of linguistic quantifiers.

4 Trend Summarization

In order to characterize the summaries of trends we will refer to Zadeh's concept of a protoform (cf., Zadeh [16]). Basically, a protoform is defined as a more or less abstract prototype (template) of a linguistically quantified proposition. Then, summaries mentioned above might be represented by two types of the protoforms of the following forms:

- We may consider summaries based on frequency and we obtain:
  - a protoform of short form of linguistic summaries:

$$Q \text{ trends are } S \tag{7}$$

    and exemplified by:

    Most of trends have *a large variability*

  - an extended form:

$$QR \text{ trends are } S \tag{8}$$

    and exemplified by:

    Most of *slowly decreasing trends* have *a large variability*

- We may also consider the duration based summaries represented by the following protoforms:
  - a short form of linguistic summaries:

$$\text{The trends that took } Q \text{ time are } S \tag{9}$$

    and exemplified by:

    The trends that took *most* time have *a large variability*

- an extended form:

$$R \text{ trends that took } Q \text{ time are } S \tag{10}$$

and exemplified by:

Slowly decreasing trends that took *most* time have *a large variability*

Using Zadeh's [15] fuzzy logic based calculus of linguistically quantified propositions, a (proportional, nondecreasing) linguistic quantifier Q is assumed to be a fuzzy set in the interval $[0, 1]$ as, e.g.

$$\mu_Q(x) = \begin{cases} 1 & \text{for } x \geq 0.8 \\ 2x - 0.6 & \text{for } 0.3 < x < 0.8 \\ 0 & \text{for } x \leq 0.3 \end{cases} \tag{11}$$

The truth values (from [0,1]) of (7) and (8) are calculated, respectively as

$$\mathcal{T}(Qy\text{'s are } S) = \mu_Q\left(\frac{1}{n}\sum_{i=1}^{n}\mu_S(y_i)\right) \tag{12}$$

$$\mathcal{T}(QRy\text{'s are } S) = \mu_Q\left(\frac{\sum_{i=1}^{n}(\mu_R(y_i) \wedge \mu_S(y_i))}{\sum_{i=1}^{n}\mu_R(y_i)}\right) \tag{13}$$

The computation of truth values of duration based summaries is more complicated and requires a different approach. While analyzing a summary "the trends that took Q time are S" we should compute the time, which is taken by those trends which "trend is S". It is obvious, when "trend is S" to degree 1, as we can use then the whole time taken by this trend. However, what should we do, if "trend is S" to some degree? We propose to take only a part of the time, defined by the degree to which "trend is S". In other words we compute this time as $\mu(y_i)t_{y_i}$, where t_{y_i} is the duration of trend y_i. The obtained value (duration of those trends, which "trend is S") is then normalized by dividing it by the overall time T. Finally, we may compute to which degree the time taken by those trends which "trend is S" is Q. A similar line of thought might be followed for the extended form of linguistic summaries.

The truth value of the short form of duration based summaries (9) is calculated as

$$\mathcal{T}(y \text{ that took } Q \text{ time are } S) = \mu_Q\left(\frac{1}{T}\sum_{i=1}^{n}\mu_S(y_i)t_{y_i}\right) \tag{14}$$

The truth value of the extended form of summaries based on duration (10) is calculated as

$$\mathcal{T}(Ry \text{ that took } Q \text{ time are } S) = \mu_Q\left(\frac{\sum_{i=1}^{n}(\mu_R(y_i) \wedge \mu_S(y_i))t_{y_i}}{\sum_{i=1}^{n}\mu_R(y_i)t_{y_i}}\right) \tag{15}$$

T is the total time of the summarized trends and t_{y_i} is the duration of the ith trend.

Both the fuzzy predicates S and R are assumed above to be of a rather simplified, atomic form referring to just one attribute. They can be extended to cover more sophisticated summaries involving some confluence of various, multiple attribute values as, e.g, "slowly decreasing and short".

Alternatively, we may obtain the truth values of (9) and (10), if we divide every trend, which takes t_{y_i} time units, into t_{y_i} trends, each lasting one time unit. For this new set of trends use frequency based summaries with the truth values defined in (12) and (13).

5 Concluding Remarks

We proposed to apply linguistic summaries of time series data to characterize in a human consistent, linguistic form some basic characteristics like trends, variability, duration, etc. We proposed to use for a preprocessing self-organizing maps (SOMs) learned using the LVQ algorithm as a classifier to derive some predefined classes of trends(labels describing the dynamics of change). We derive linguistic summaries of the type: "most of the trends are short", "most long trends are increasing", "trends taking most of time are increasing" or "increasing trends taking most of the time are of a low variability", etc. We combined the use of the SOMs with the LQV function used for a supervised learning for a preprocessing, with the use of Zadeh's calculus of linguistically quantified propositions to derive linguistic summaries of trends.

References

1. E. German, D. G. Ece, O. N. Gerek (2005). Self Organizing Map (SOM) Approach for Classification of Power Quality Events. *LNCS 3696*, Springer Verlag, pp:403-408.
2. J. Kacprzyk and R.R. Yager (2001). Linguistic summaries of data using fuzzy logic. In *International Journal of General Systems*, 30:33-154.
3. J. Kacprzyk, R.R. Yager and S. Zadrożny (2000). A fuzzy logic based approach to linguistic summaries of databases. In*International Journal of Applied Mathematics and Computer Science*, 10: 813-834.
4. L.A. Zadeh and J. Kacprzyk, Eds. (1999) *Computing with Words in Information/Intelligent Systems. Part 1. Foundations, Part 2. Applications*, Springer–Verlag, Heidelberg and New York.
5. J. Kacprzyk and S. Zadrożny (1995). FQUERY for Access: fuzzy querying for a Windows-based DBMS. In P. Bosc and J. Kacprzyk (Eds.) *Fuzziness in Database Management Systems*, Springer-Verlag, Heidelberg, 415-433.
6. J. Kacprzyk and S. Zadrożny (1999). The paradigm of computing with words in intelligent database querying. In L.A. Zadeh and J. Kacprzyk (Eds.) *Computing with Words in Information/Intelligent Systems. Part 2. Foundations*, Springer–Verlag, Heidelberg and New York, 382-398.

7. J. Kacprzyk, S. Zadrożny (2005). Linguistic database summaries and their protoforms: toward natural language based knowledge discovery tools. In *Information Sciences* 173: 281-304.
8. J. Kacprzyk, S. Zadrożny (2005). Fuzzy linguistic data summaries as a human consistent, user adaptable solution to data mining. In B. Gabrys, K. Leiviska, J. Strackeljan (Eds.) *Do Smart Adaptive Systems Exist?* Springer, Berlin Heidelberg New York, 321-339.
9. J. Kacprzyk, A. Wilbik, S. Zadrożny (2006). Linguistic summarization of trends: a fuzzy logic based approach. (in press).
10. T. Kohonen, (1995). *Self-Organizing Maps*, Springer Verlag.
11. T. Kohonen, (1998). The self-organizing map. In *Neurocomputing*, 21: 1-6.
12. G. Simon, J. A. Lee, M. Verleysen (2005). On the need of unfold preprocessing for time series clustering. 5th Workshop on Self-Organizing Maps, Paris, 5-8 September 2005, pp. 251-258.
13. G. Simon, A. Lendasse, M. Cottrell, J.-C. Fort, M. Verleysen (2005). Time series forecasting: Obtaining long term trends with self-organizing maps. In *Pattern Recognition Letters* 26: pp.1795-1808.
14. R.R. Yager (1982). A new approach to the summarization of data. *Information Sciences*, 28: 69-86.
15. L.A. Zadeh (1983). A computational approach to fuzzy quantifiers in natural languages. In *Computers and Mathematics with Applications*, 9: 149-184.
16. L.A. Zadeh (2002). A prototype-centered approach to adding deduction capabilities to search engines – the concept of a protoform. BISC Seminar, University of California, Berkeley.

Long-Term Prediction of Time Series Using State-Space Models

Elia Liitiäinen and Amaury Lendasse

Neural Networks Research Centre, Helsinki University of Technology, Finland
eliitiai@cc.hut.fi, lendasse@hut.fi

Abstract. State-space models offer a powerful modelling tool for time series prediction. However, as most algorithms are not optimized for long-term prediction, it may be hard to achieve good prediction results. In this paper, we investigate Gaussian linear regression filters for parameter estimation in state-space models and we propose new long-term prediction strategies. Experiments using the EM-algorithm for training of nonlinear state-space models show that significant improvements are possible with no additional computational cost.

1 Introduction

Time series prediction [1] is an important problem in many fields like ecology and finance. The goal is to model the underlying system that produced the observations and use this model for prediction.

Often, obtaining prior information of a time series is difficult. In this kind of a situation, a black-box model can be used. An often used approach for prediction is nonlinear regression, which works well especially for deterministic time series.

An alternative to nonlinear regression is state-space modeling. A state-space model can model a wide variety of phenomena, but unfortunately learning a state-space presentation for data is challenging. Previous work on this topic includes dual Kalman filtering methods [2], variational Bayesian learning [3] and EM-algorithm for nonlinear state-space models [4].

In [4] an EM-algorithm for training of state-space models is proposed. From the point of view of time series prediction, the drawback of this algorithm is that the model is not exactly optimized for prediction.

In this paper, we have two goals. It is shown that the EKS which was used in [4] can be replaced by a more efficient linear regression smoother. We also show that with simple methods it is possible to improve the long-term prediction ability of the algorithm. The improvement comes with no additional computational cost and our methods can be applied also to other algorithms. In the experimental section, we test the proposed methods with the Poland electricity and Darwin sea level pressure datasets.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 181–190, 2006.

2 Gaussian Linear Regression Filters

Consider the nonlinear state-space model

$$x_k = f(x_{k-1}) + w_k \tag{1}$$
$$y_k = h(x_k) + v_k, \tag{2}$$

where $w_k \sim N(0, Q)$ and $v_k \sim N(0, r)$ are independent Gaussian random variables, $x_k \in R^n$ and $y_k \in R$. Here, $N(0, Q)$ means normal distribution with the covariance matrix Q. Correspondingly, r is the variance of the observation noise.

The filtering problem is stated as calculating $p(x_k|y_1, \ldots, y_k)$. If f and h are linear, this could be done using Kalman filter [5]. The linear filtering theory can be applied to nonlinear problems by using Gaussian linear regression filters (LRKF, [6]) that are recursive algorithms based on statistical linearization of the model equations. The linearization can be done in various ways resulting in slightly different algorithms.

Denote by $\tilde{p}(x_k|y_1, \ldots, y_k)$ a Gaussian approximation of $p(x_k|y_1, \ldots, y_k)$. The first phase in a recursive step of the algorithm is calculating $\tilde{p}(x_{k+1}|y_1, \ldots, y_k)$, which can be done by linearizing $f(x_k) \approx A_k x_k + b_k$ so that the error

$$\mathrm{tr}(e_k) = \mathrm{tr}\ (\int_{R^{n_x}} (f(x_k) - A_k x_k - b_k)(f(x_k) - A_k x_k - b_k)^T \tilde{p}(x_k|y_0, \ldots, y_k)\ dx_k) \tag{3}$$

is minimized. Here, tr denotes the sum of the diagonal elements of a matrix. In addition Q is replaced by $\tilde{Q}_k = Q + e_k$. The linearized model is used for calculating $\tilde{p}(x_{k+1}|y_1, \ldots, y_k)$ using the theory of linear filters. The measurement update

$$\tilde{p}(x_{k+1}|y_1, \ldots, y_k) \rightarrow \tilde{p}(x_{k+1}|y_1, \ldots, y_{k+1})$$

is done by a similar linearization. The update equations for the Gaussian approximations can be found in [7].

Approximating equation 3 using central differences would lead to the central difference filter (CFD, [5]) which is related to the unscented Kalman filter [5]. A first order approximation on the other hand would yield the widely used extended Kalman filter algorithm [8]. However, by using Gaussian nonlinearities as described in the following sections, no numerical integration is needed in the linearization. The filter based on closed form calculations is called a (Gaussian) linear regression filter.

The smoothed density $p(x_l|y_1, \ldots, y_k)$ $(l < k)$ is also of interest. In our experiments we use the Rauch-Tung-Striebel smoother [8] with the linearized model. Other alternatives are of course also of interested and will be possibly investigated in future research. See for example [9] in which three different smoothing methods are compared.

3 EM-Algorithm for Training of Radial Basis Function Networks

In this section, a training algorithm introduced in [4] is described. The EM-algorithm is used for parameter estimation for nonlinear state-space models. When Gaussian filters are used, the M-step of the algorithm can be solved in closed form resulting in an efficient method which in practice converges fast.

3.1 Parameter Estimation

Suppose that a sequence of observations $(y_k)_{k=1}^N$ is available. The underlying system that produced the observations is modelled as a state-space model. The functions f and g in equations 1 and 2 are parametrized using RBF-networks:

$$f = w_f^T \Phi_f \tag{4}$$

$$g = w_g^T \Phi_g, \tag{5}$$

where $\Phi_f = [\rho_1^f(x), \ldots, \rho_l^f(x)\ x^T\ 1]^T$ and $\Phi_g = [\rho_1^g(x), \ldots, \rho_j^g(x)\ x^T\ 1]^T$ are the neurons of the RBF-networks. The nonlinear neurons are of the form

$$\rho(x) = |2\pi S|^{-1/2} \exp(-\frac{1}{2}(x-c)^T S^{-1}(x-c)). \tag{6}$$

The free parameters of the model are the weights of the neurons, w_f and w_g in equations 4 and 5, and the noise covariances Q and r in equations 1 and 2. In addition, the initial condition for the states is chosen Gaussian and the parameters of this distribution are optimized.

The EM-algorithm is a standard method for handling missing data. Denoting by θ the free parameters of the model, the EM-algorithm is used for maximizing $p(y_1, \ldots, y_T|\theta)$. The EM-algorithm for learning nonlinear state-space models is derived in [4].

The algorithm is recursive and each iteration consists of two steps. In the E-step, the density $p(x_0, \ldots, x_N|y_1, \ldots, y_N, \theta)$ is approximated and in the M-step this approximation is used for updating the parameters.

Due to the linearity of the model with respect to the parameters, the M-step can be solved analytically. The update formulas for the weights w_f and w_g and covariances Q and R can be found in [4]. In our implementation Q is chosen diagonal.

The E-step is more difficult and an approximative method must be used. In addition to Gaussian filters, there exists many different methods for nonlinear smoothing, for example the particle smoother and numerical quadrature based methods [10]. In the case of neural networks, the problem is that function evaluation may be relatively expensive if a high number of neurons is used. We propose the use of the smoother derived in section 2 instead of the extended Kalman smoother used in [4]. The extended Kalman smoother uses rough approximations to propagate nonlinearities which may cause inaccuracy (see for example [11]). Our choice does not significantly increase the computational complexity of the algorithm which did not pose problems in our experiments.

3.2 Initialization

The state variables must be initialized to some meaningful values before the algorithm can be used. Consider a scalar time series (y_t). First the vectors

$$z_t = [y_{t-L}, \ldots, y_t, \ldots, y_{t+L}] \tag{7}$$

are formed. L is chosen large enough so that the vectors contain enough information. Based on Taken's theorem [12], it can be claimed that setting L to the dimension of the state-space is a good choice [3], but as many time series contain noise, this is not always the case. In our experiments, we use the value $L = 10$ which produced good results in our experiments and is large enough for most time series. In case of bad results, a lower value for L might sometimes give better initializations.

Next the dimension for the hidden states is chosen. Once the dimension is known, the vectors z_t are projected onto this lower dimensional space. This is done with the PCA mapping [13]. In highly nonlinear problems, it may be essential to use kernel PCA like was done in [14].

The rough estimates for the hidden states are used to obtain an initial guess for the parameters of the network.

3.3 Choosing Kernel Means and Widths

Choosing the centers of the neurons (c in equation 6) is done with the k-means algorithm [13]. The widths S_j are chosen according to the formula (see [15])

$$S_j = \frac{1}{l}(\sum_{i=1}^{l} \|c_j - c_{N(j,i)}\|^2)^{\frac{1}{2}} I, \tag{8}$$

where I is the identity matrix and $N(j, i)$ is the ith nearest neighbor of j. In the experiments, we use the value $l = 2$ as proposed in [15].

3.4 Choosing the Number of Neurons and the Dimension of the State-Space

To estimate the dimension of the state-space, we propose the use of a validation set to estimate the generialization error. For each dimension, the model is calculated for different number of neurons and the one which gives the lowest one-step prediction validation error is chosen. The linear regression filter is used for calculating the error.

For choosing the number of neurons, we propose the use of the likelihood of the model, which has the advantage that the whole data set is used for training. Usually there is a bend after which the likelihood decreases only slowly (see figure 1c). This kind of a bend is used as an estimate of the point after which overfitting becomes a problem.

4 Long-Term Prediction of Time Series

By long-term prediction of a time series we mean predicting y_{t+k} for $k > 1$ given the previous observations $y_1, \ldots, y_t$. The model in equations 1 and 2 can be used for prediction by calculating a Gaussian approximation to $p(y_{t+k}|y_1, \ldots, y_t)$ with the linear regression filter or the EKF. However, the drawback of this approach is that the EM-algorithm does not optimize the long-term prediction performance of the model and thus the prediction error is expected to grow fast as the prediction horizon k grows. Our claim is motivated for example by the results in [16], which show that for nonlinear regression direct prediction instead of recursive gives much better results on long-term prediction.

Instead of the previous method, we propose three alternative methods. For the first two methods the observation model in equation 5 is replaced by an RBF-network with the same centers and widths for the neurons than the original model but different weights that are optimized for long-term prediction. In the first method, the weights are optimized to minimize the prediction error when the linear regression filter with the original model is used to estimate $p(x_t|y_1, \ldots, y_t)$ and the EKF with the modified model for estimating $p(x_{t+k}|y_1, \ldots, y_t)$ and $p(y_{t+k}|y_1, \ldots, y_t)$.

In the second method, a similar optimization is performed in the case where the linear regression filter is used for estimating all three distributions.

As a third methods, we propose extending the first method so that also the weights of the network in equation 4 are reoptimized. This leads to a nonlinear optimization problem, which can be solved using standard methods, see for example [17].

The optimization of the weights for the methods is done using the same training set as for the EM-algorithm. The cost function is

$$\sum_{i=1}^{N-k} (y_{i+k} - \hat{y}_{i+k})^2,$$

where $\hat{y}_{i+k}$ are the predictions given the parameter values and observations up to time i. The three different methods are used for calculating $\hat{y}_{i+k}$. The first two lead to quadratic optimization problems which are easy to solve in closed form, whereas to optimize the cost for the third method any nonlinear optimization method can be used.

There certainly exists different methods than the ones we use. Our goal is not to test all different possibilities, only to show that when state-space models are used, an ordinary recursive prediction is not the best choice.

5 Experiments

In this section the proposed methods are tested on two different time series. The first time series represents the electricity consumption in Poland and is

interesting also from practical point of view as electricity companies can certainly make use of this kind of information. The other time series, Darwin sea level pressure, is also measurements of a real world phenomena. The data sets can downloaded from [18]. Unless we state explicitly otherwise, the linear regression filter based algorithm is used for training.

5.1 Poland Electricity Consumption

The Poland electricity consumption time series is a well-known benchmark problem [19]. For prediction results with nonlinear regression we refer to [16]. The series is plotted in figure 1a. The values $1 - 1000$ are used for training and the values $1001 - 1400$ for testing. For dimension selection, the values 601-1000 are kept for validation.

The model is tested using the dimensions 2 to 8 for the state-space. For each dimension, the validation errors are calculated for different number of neurons so that the number of neurons goes from 0 to 36 by step of 2 (we use the same amount of neurons for both networks). For each number of neurons, the training is stopped if the likelihood decreased twice in row or more than 40 iterations have been made.

The dimension 7 yields the lowest validation error and was thus chosen as the dimension for the states. In figure 1c is the corresponding likelihood curve for this dimension. Based on the likelihood, we choose 8 as the number of neurons.

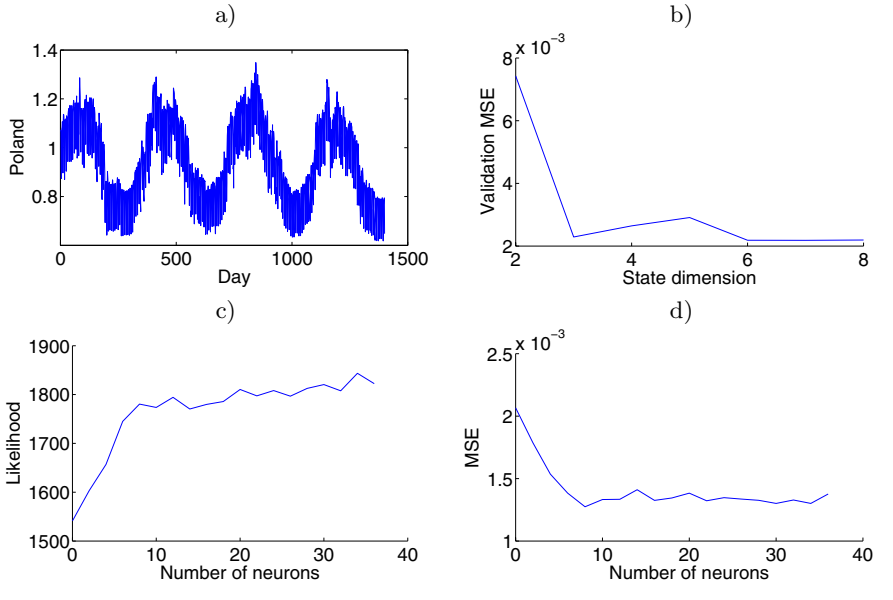


Fig. 1. a) The Poland electricity consumption time series. b) Validation errors for state dimensions 2-8. c) Likelihoods for dimension 7. d) One step prediction test errors for dimension 7.

From figure 1d, it can be seen that the short-term prediction performance of the model is good.

The chosen model is used for long-term prediction with different prediction horizons. We test the three methods introduced in section 4. The results are in figure 2a. In figure 2a, the test errors are calculated by estimating the state at each time point of the test set with the linear regression filter. After that the methods in section 4 are used to calculate the prediction estimates corresponding to the number of prediction steps.

In figure 2b for comparison we have implemented the EKS-based algorithm in [4], where for prediction the EKF has been used. Also for this algorithm we choose 7 as the dimension of the state-space and 8 as the number of neurons. Because the error for the algorithm in [4] has a quite high variance compared to our method, the curves in figure 2b are averaged over 5 simulations.

It can be seen that the linear regression filter in training improves the result compared to using EKS. Also, the stability of the algorithm is improved. The linear regression filter gives higher covariances for the smoothed state estimates which has a regularizing effect on the model.

For short-term prediction the prediction error is not significantly improved by the methods proposed in section 4. However, as the prediction horizon grows, the differences between the methods grow. Thus in this time series, the long-term prediction methods in section 4 should be used as they introduce no additional computational cost.

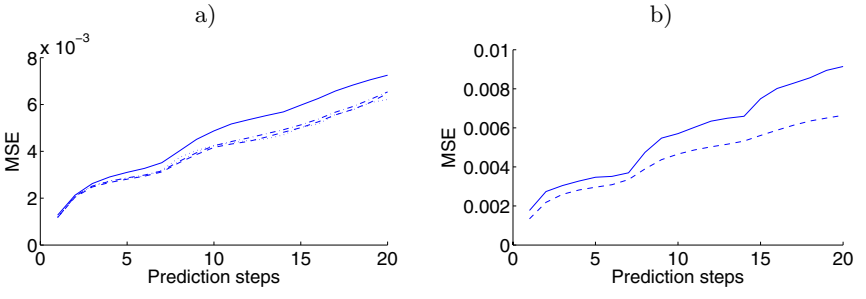


Fig. 2. Long-term prediction errors. a) The solid line is the recursive prediction error with the linear regression filter. The dashed, dashdotted and dotted lines give the prediction error for the other three methods in section 4 so that dashed corresponds to the first of these methods, dashdot to the second etc. b) Results with the original algorithm in [4] (solid line) and the basic version of our algorithm (dashed line). The curves in figure b are averaged over 5 simulations.

5.2 Darwin Sea Level Pressure

The Darwin sea level pressure time series consists of 1400 values which are drawn in figure 3a. The values from 1 to 1000 are used for training and the rest for testing the model.

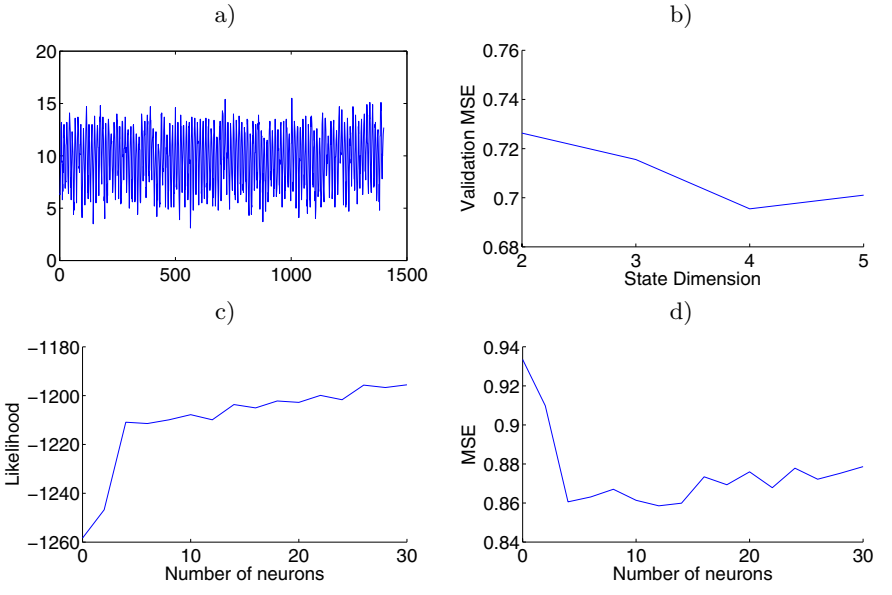


Fig. 3. a) The Darwin sea level pressure time series. b) Validation errors for state dimensions 2-5. c) Likelihoods for dimension 4. d) One step prediction test errors for dimension 4.

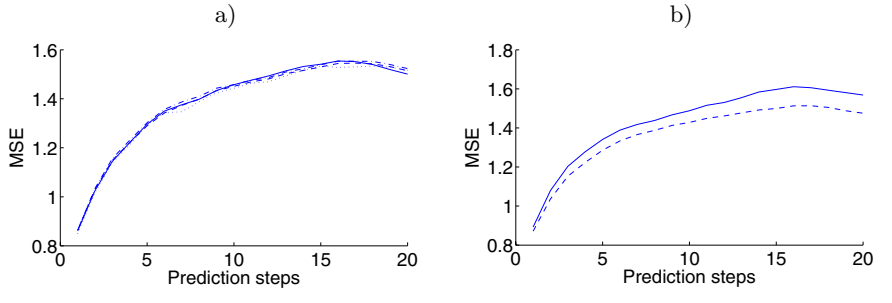


Fig. 4. a) The solid line is the recursive prediction error with the linear regression filter. The dashed, dashdotted and dotted lines give the prediction errors for the other three methods in section 4 so that dashed corresponds to the first of these methods, dashdot to the second etc. b) Results with the original algorithm in [4] (solid line) and the basic version of our algorithm (dashed line). The curves in figure b are averaged over 5 simulations.

As in the previous section, the dimension of the state-space is chosen using a validation set. The dimensions from 2 to 5 are tested. The lowest validation error is that of dimension 4. The corresponding likelihood curve is in figure 3c. There is a clear bend in the likelihood curve, which can be used for model selection. Based on the curve we choose 4 as the number of neurons.

From figure 4b, it can be seen that the original algorithm with EKS is again worse than our algorithm.

The long-term prediction results with the methods in section 4 are in figure 4a. In this problem, the difference between different prediction methods is much smaller. No significant improvement was obtained over recursive prediction. This is probably due to the linearity and periodicity of the data. In this time series, an accurate long-term prediction is possible.

Based on the experiments, we claim that the best of the methods proposed in section 4 is the one based on minimizing the prediction performance of the EKF. It seems that using Gaussian distributions instead of point values as a training set is not very useful.

6 Conclusion

In this paper the Gaussian linear regression smoother with the EM-algorithm is used as a method for learning a state-space presentation for a time series. It is shown that our approach brings real improvement over the original EKS based algorithm presented in [4] for significantly nonlinear problems. The proposed smoother gives implicitely additional regularization compared to the extended Kalman smoother. Still it is clear that the EM-algorithm does not minimize the long-term prediction error of the model which results in high prediction errors once the prediction horizon grows.

In this paper we proposed strategies for improving the long-term prediction capability of state-space models. The methods can be applied for many algorithms in addition to the EM-algorithm used in this paper. The experimental results show that clear improvement over the ordinary recursive approach is possible. However, for nearly linear time series the results were not as convincing and it can be concluded that big improvements can be obtained mainly for significantly nonlinear systems.

In the future methods for optimizing state-space models for long-term prediction will be investigated. Most current methods use a cost function that poorly fits to long-term prediction.

References

1. *Time Series Prediction: Forecasting the Future and Understanding the Past*. Addison-Wesley Publishing Company, 1994.
2. A. T. Nelson. *Nonlinear Estimation and Modeling of Noisy Time-series by Dual Kalman Filtering Methods*. PhD thesis, Oregon Graduate Institute, 2000.
3. H. Valpola and J. Karhunen. An unsupervised ensemble learning method for nonlinear dynamic state-space models. *Neural Computation*, 14(11):2647–2692, 2002.
4. Z. Ghahramani and S. Roweis. Learning nonlinear dynamical systems using an EM algorithm. In *Advances in Neural Information Processing Systems*, volume 11, pages 431–437. 1999.
5. R. van der Merwe. *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, Oregon Health & Science University.

6. T. Lefebvre, H. Bruyninckx, and J. De Schutter. Kalman filters for non-linear systems: a comparison of performance. *International Journal of Control*, 77(7):639–653, 2004.
7. K. Ito and K. Q. Xiong. Gaussian filters for nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 45(5):910–927, 2000.
8. S. Haykin, editor. *Kalman Filtering and Neural Networks*. Wiley Series on Adaptive and Learning Systems for Signal Processing. John Wiley & Sons, Inc., 2001.
9. T. Raiko, M. Tornio, A. Honkela, and J. Karhunen. State inference in variational bayesian nonlinear state-space models. In *6th International Conference on Independent Component Analysis and Blind Source Separation, ICA 2006, Charleston, South Carolina, USA, March 5-8, 2006*.
10. A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.
11. Eric A. Wan, R. van der Merwe, and A. T. Nelson. Dual estimation and the unscented transformation. In *Advances in Neural Information Processing Systems*, 2000.
12. D. A. Rand and L. S. Young, editors. *Dynamical Systems and Turbulence*, volume 898 of *Lecture Notes in Mathematics*, chapter Detecting strange attractors in turbulence. Springer-Verlag, 1981.
13. S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2nd edition, 1998.
14. A. Honkela, S. Harmeling, L. Lundqvist, and H. Valpola. Using kernel PCA for initialisation of variational bayesian nonlinear blind source separation method. In *Proceedings of the Fifth International Conference Independent Component Analysis and Blind Signal Separation (ICA 2004), Granada, Spain.*, volume 3195 of *Lecture Notes in Computer Science*, pages 790–797. Springer-Verlag, 2004.
15. J. Moody and C. J. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, pages 281–294, 1989.
16. Y. Ji, J. Hao, N. Reyhani, and A. Lendasse. Direct and recursive prediction of time series using mutual information selection. In *Computational Intelligence and Bioinspired Systems: 8th International Workshop on Artificial Neural Networks, IWANN 2005, Vilanova i la Geltrú, Barcelona, Spain, June 8-10,2005*, pages 1010–1017.
17. R. Fletcher. *Practical Methods of Optimization*, volume 1. John Wiley and Sons, 1980.
18. Available from www.cis.hut.fi/projects/tsp/download.
19. M. Cottrell, B. Girard, and P. Rousset. Forecasting of curves using classification. *Journal of Forecasting*, 17(5-6):429–439, 1998.

Time Series Prediction Using Fuzzy Wavelet Neural Network Model

Rahib H. Abiyev

Department of Computer Engineering, Near East University, Lefkosa, North Cyprus
rahib@neu.edu.tr

Abstract. The fuzzy wavelet neural network (FWNN) for time series prediction is presented in this paper. Using wavelets the fuzzy rules are constructed. The gradient algorithm is applied for learning parameters of fuzzy system. The application of FWNN for modelling and prediction of complex time series and prediction of electricity consumption is considered. Results of simulation of FWNN based prediction system is compared with the simulation results of other methodologies used for prediction. Simulation results demonstrate that FWNN based system can effectively learn complex nonlinear processes and has better performance than other models.

1 Introduction

Time-series prediction is one of important research and application area. Traditional methods used for prediction are based on technical analysis of time-series, such as looking for trends, stationarity, seasonality, random noise variation, moving average. Most of them are linear approaches. These are exponential smoothing method, well-known Box-Jenkins method [1] which have shortcomings.

Softcomputing methodologies such as neural networks, fuzzy logics, genetic algorithms are applied for prediction chaotic time series [2-7]. These methods have shown clear advantages over the traditional statistical ones [2]. In this paper the development of fuzzy system based on wavelet neural network is considered for time-series prediction, in particularly, for prediction of electricity consumption.

Neural networks are widely used for generating IF-THEN rules of fuzzy systems. One of type of neural networks is wavelet neural networks (WNNs) [8-13]. WNNs are feed-forward neural networks that use wavelets as activation function in hidden layer. The network based on wavelet has simple structure and good learning speed. It can converge faster and be more adaptive to new data. WNN can approximate complex functions to some precision very compactly and can be more easily designed and trained than other networks, such as multilayer perceptrons and radial based networks. The number of methods is implemented for initializing wavelets, such as orthogonal least square procedure, clustering method [8]. The optimal dilation and translation of the wavelet increases training speed and obtains fast convergence.

Wavelet neural networks are used for prediction of time-series, from limited number of data points [12,13]. Multiresolution character of wavelets permits to catch short term and long term variations. Fuzzy wavelet neural network (FWNN) combines

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 191 – 200, 2006.

wavelet theory to fuzzy logic and neural networks. The synthesis of FWNN system includes the finding of the optimal definitions of the premise and consequent part of fuzzy IF-THEN rules through the training capability of wavelet neural networks. In neuro-fuzzy systems one of difficulties is the correct linguistic interpretation of rules. The multiresolution techniques allow determining appropriate membership functions for given data points. The dictionary of membership functions forming multiresolution is used to determine which membership function most appropriate to describe the data points. The membership function defined for each linguistic term is well defined beforehand and are not modified during learning. In the paper fuzzy wavelet neural inference structure is applied for prediction of time series.

The paper is organized as follows. Section 2 describes structure of WNN. Section 3 presents the learning algorithms of FWNN system. Section 4 contains simulation results of FWNN used for prediction of chaotic time series and electricity consumption. Finally a brief conclusion is presented in section 5.

2 Wavelet Neural Network

A wavelet networks are nonlinear regression structure that represents input-output mappings. Wavelet networks use three-layer structure and wavelet activation function. Wavelet function is a waveform that has limited duration and average value of zero. There are number of wavelet functions. In this paper Mexican Hat wavelet (Fig. 1) is used for neural network.

$$\psi(z) = \alpha(1 - z^2) * e^{-\frac{z^2}{2}} \tag{1}$$

Here $\alpha = \frac{2}{\sqrt{3}} \pi^{-1/4}$. This wavelet function is used in hidden layer of network.

The structure of wavelet neural network (WNN) is given in fig. 2. Here x_1, x_2, …, x_m are network input signals. $\Psi(z_j)$ are wavelet functions which are used in hidden layer. z_i are calculated by the following expression.

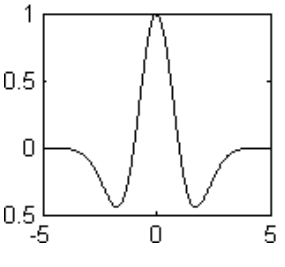


Fig. 1. Mexican Hat wavelet

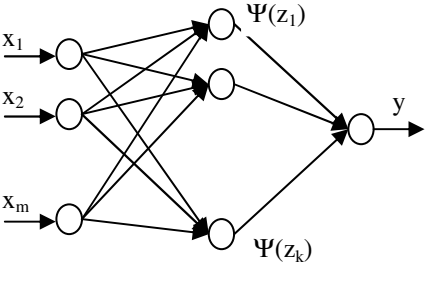


Fig. 2. Architecture of WNN

$$z_j = \sum_{i=1}^{m} (x_i a_{ij} - b_j) \tag{2}$$

here a_{ij} and b_j are network parameters, i=1,2,…,m; j=1,2,…,k

Using expression (1) and (2) the output signals of hidden layer are determined. These signals are input for the last- third layer. The output signal of network is calculated as

$$y = \sum_{j=1}^{k} w_j \psi(z_j) \tag{3}$$

w_i are weight coefficients between hidden and output layers, j=1,2,…,k.

The described WNN structure will be used in FWNN that will be presented in the next section.

3 Fuzzy Wavelet Neural Network

The kernel of fuzzy controller is fuzzy knowledge base. In fuzzy knowledge base the information that consists of input-output data points of the system is interpreted into linguistic interpretable fuzzy rules. In the paper the fuzzy rules that have IF-Then form are used. These fuzzy rules are constructed by using wavelets as:

If x_1 is A_{i1} and x_2 is A_{i2} and…and x_m is A_{im} Then y_i is $\sum_{k=1}^{K} w_{kj}(1 - z_{kj}^2) e^{-\frac{z_{kj}^2}{2}}$ (4)

Here x_j (j=1,..,m) are input variables, y_i (i=1,..,n) are output variables which are sum of Mexican Hat wavelet functions, A_{ij} is a membership function for i-th rule of the k-th input defined as Gaussian membership function. K is number of hidden neurons in WNN. Conclusion parts of rules contain WNNs.

The fuzzy model that is described by production rules can be obtained by modifying parameters of conclusion and premise parts of the If-Then rules. Sometimes the premise parts of these rules are constructed by experts, and only conclusion parts that are wavelet functions are adjusted in order to obtain correct fuzzy model.

The structure of fuzzy wavelet neural network is given in fig. 3. The FWNN consists of combination of two network structures. Upper side of figure 3 contains n wavelet neural networks that are denoted by WNN_1, WNN_2,…,WNN_n. These networks are included to the consequent parts of the fuzzy rules. Down side of figure contains network structure of fuzzy reasoning mechanism. These are premise parts of fuzzy rules.

In the first layer of fuzzy reasoning mechanism the number of nodes is equal to the number of input signals. In the second layer each node corresponds to one linguistic term. For each input signal entering the system the membership degree to which input value belongs to a fuzzy set is calculated. To describe linguistic terms the Gaussian membership function is used.

$$\mu 1_j(x_i) = e^{-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}}, \quad i=1..m, \quad j=1..J \tag{5}$$

Here m is number of input signals, J is number of nodes in second layer. c_{ij} and σ_{ij} are centre and width of the Gaussian membership functions of the j-th term of i-th input variable, respectively. $\mu 1_j(x_i)$ is membership function of i-th input variable for j-th term. m is number of external input signals. J is number of linguistic terms assigned for external input signals x_i.

In the third layer the number of nodes correspond to the number of rules R_1, R_2,…,R_n. Each node represents one fuzzy rule. Here to calculate the values of output signals of the layer AND (min) operation is used.

$$\mu_l(x) = \prod_j \mu 1_j(x_i), \; l=1,..,n, \; j=1,..,J \tag{6}$$

Π is min operation. The $\mu_l(x)$ signals are input signals for the next layer. This layer is a consequent layer. In this layer the output signals of previous layer are multiplied to the output signals of wavelet neural network and defuzzification is made to calculate the output of whole network.

$$\mu_l(x) = \sum_{l=1}^{n} \mu_l(x) y_l \Big/ \sum_{l=1}^{n} \mu_l(x) \tag{7}$$

Here y_l is the outputs of wavelet neural networks, u is output of whole network.

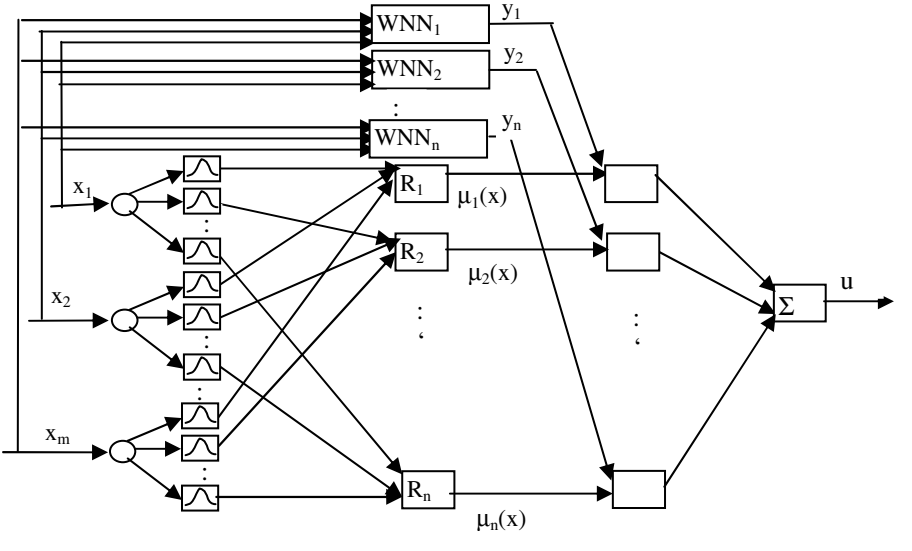


Fig. 3. Structure of fuzzy wavelet neural network

The outputs of each wavelet neural networks WNN_1, WNN_2,…,WNN_n in fig. 3 are calculated by using equations (1-3). After calculating output signal of FWNN the training of network start.

Training includes the adjusting of the parameters values of membership function $c_{ij}(t)$ and $\sigma_{ij}(t)$ (i=1,..,n, j=1,..,m) in the premise part and parameters values of wavelet neural network $w_j(t)$, $a_{ij}(t)$, $b_j(t)$ (i=1,..,n, j=1,..,k) in consequent part. At first step, in the network output the value of error is calculated.

$$E = \frac{1}{2}\sum_{i=1}^{O}(u_i^d - u_i)^2 \tag{8}$$

Here O is number of output signals of network (in given case O=1), u_i^d and u_i are desired and current output values of network, correspondingly. The parameters w_j, a_{ij}, b_j, (i=1,..,m, j=1,..,k) and c_{ij} and σ_{ij} (i=1,..,m, j=1,..n) of neuro-fuzzy structure are adjusted by using following formulas.

$$w_j(t+1) = w_j(t) + \gamma\frac{\partial E}{\partial w_j} + \lambda(w_j(t) - w_j(t-1)), \quad j = 1,..,\ k \tag{9}$$

$$a_{ij}(t+1) = a_{ij}(t) + \gamma\frac{\partial E}{\partial a_{ij}} + \lambda(a_{ij}(t) - a_{ij}(t-1)), \quad i = 1..m; \quad j = 1,..,\ k \tag{10}$$

$$b_j(t+1) = b_j(t) + \gamma\frac{\partial E}{\partial b_j} + \lambda(b_j(t) - b_j(t-1)), \quad j = 1,..,\ k \tag{11}$$

$$c_{ij}(t+1) = c_{ij}(t) + \gamma\frac{\partial E}{\partial c_{ij}},\ \sigma_{ij}(t+1) = \sigma_{ij}(t) + \gamma\frac{\partial E}{\partial \sigma_{ij}},\ i=1,..,m,\ j=1,..,n \tag{12}$$

Here γ is learning rate, λ is momentum rate, k is number of neurons in hidden layer of wavelet neural network, m is number of input signals of the network (input neurons) and n is number of rules (hidden neurons).

The derivatives in (9-11) are determined by the following formulas.

$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial u}\frac{\partial u}{\partial y_l}\frac{\partial y_l}{\partial w_j} = (u - u^d)\cdot\psi(z_j)\cdot\mu_l \Big/ \sum_{l=1}^{n}\mu_l \tag{13}$$

$$\frac{\partial E}{\partial a_{ij}} = \frac{\partial E}{\partial u}\frac{\partial u}{\partial y_l}\frac{\partial y_l}{\partial \psi_j}\frac{\partial \psi_j}{\partial z_i}\frac{\partial z_j}{\partial a_{ij}} = \delta_{lj}(z_j^3 - 3z_j)e^{-\frac{z_j^2}{2}}x_i, \tag{14}$$

$$\frac{\partial E}{\partial b_j} = \frac{\partial E}{\partial u}\frac{\partial u}{\partial y_l}\frac{\partial y_l}{\partial \psi_j}\frac{\partial \psi_j}{\partial z_j}\frac{\partial z_j}{\partial b_j} = -\delta_{lj}(z_j^3 - 3z_j)e^{-\frac{z_j^2}{2}} \tag{15}$$

Here $\delta_{lj} = \frac{\partial E}{\partial u}\frac{\partial u}{\partial y_l}\frac{\partial y_l}{\partial \psi_j} = (u - u^d)\cdot w_j\cdot\mu_l \Big/ \sum_{l=1}^{n}\mu_l$, i=1,..,m, j=1,..,k, l=1,..,n

The derivatives in (12) are determined by the following formulas.

$$\frac{\partial E}{\partial c_{ij}} = \sum_j \frac{\partial \mathrm{E}}{\partial u}\frac{\partial u}{\partial \mu_l}\frac{\partial \mu_l}{\partial c_{ij}} \quad , \quad \frac{\partial E}{\partial \sigma_{ij}} = \sum_j \frac{\partial \mathrm{E}}{\partial u}\frac{\partial u}{\partial \mu_l}\frac{\partial \mu_l}{\partial \sigma_{ij}} \tag{16}$$

$$\frac{\partial E}{\partial u} = u(t) - u^d(t),\ \frac{\partial u}{\partial \mu_l} = (y_l - u)\Big/\sum_{l=1}^{L}\mu_l\ ,\ \ \mathrm{i}=1,..,\mathrm{m};\mathrm{j}=1,..,\mathrm{n};\mathrm{l}=1,..,\mathrm{n}; \tag{17}$$

$$\frac{\partial \mu_l(x_j)}{\partial c_{ji}} = \begin{cases} \mu_l(x_j)\dfrac{2(x_j - c_{ji})}{\sigma_{ji}^2} & \text{if } j \text{ node} \\ \quad \text{is connected to rule node } l \\ 0, \quad \text{otherwise} \end{cases} , \quad \frac{\partial \mu_l(x_j)}{\partial \sigma_{ji}} = \begin{cases} \mu_l(x_j)\dfrac{2(x_j - c_{ji})^2}{\sigma_{ji}^3} & \text{if } j \text{ node} \\ \quad \text{is connected to rule node } l \\ 0, \quad \text{otherwise} \end{cases} \tag{18}$$

Using equations (13-18) the derivatives in (9-12) are calculated and the correction of the parameters of FWNN is carried out.

4 Simulation

4.1 Time Series Prediction

The FWNN structure and its learning algorithms are applied for predicting the future values of chaotic time series. As an example the Mackey-Glass time series data set was taken. This is a benchmark problem in the areas of neural network and fuzzy systems. This time series data set was created with the use of the following Mackey-Glass time-delay differential equation.

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{19}$$

This time series is chaotic, and the trajectory is highly sensitive to initial conditions. To obtain the data set, the fourth-order Runge-Kutta method is applied to find the numerical solution to the above Mackey-Glass equation. Here we assume that x(0)=1.2, τ=17, and x(t)=0 for t<0. The task is to predict the values x(t+pr) from input vectors [x(t-18) x(t-12) x(t-6) x(t)] for any value of the time t. Here pr is predicting step. The value of pr is taken as 6. Using statistical data, obtained from (19), the learning of FWNN has been carried out.

The 16 rules are used in neuro-fuzzy part of FWNN. As a performance criterion the nondimensional error index (NDEI) which is defined as the root mean square error (RMSE) divided by the standard deviation of target series is used. The 1000 data points (t=117 to 1118) are extracted from time-series and used as learning data. The first half 500 data points were used for learning, and second half 500 data points were used for testing. During learning the values of RMSE=0.003189 and NDEI=0.014364.

After learning in the generalization step the values of RMSE=0.003359 and NDEI=0.015076. In fig. 4 (a) the trajectories of desired and predicted values for both training and checking data for pr=6 are shown. Here solid line indicates the trajectory of statistical data and dashed line the predicted value of time series. The differences between them are very little. These differences might only be seen in large scale. In fig. 4(b) the prediction error is shown. For comparative analysis the feed-forward NN and WNN based prediction models are developed. The result of feed-forward NN based model is obtained when number of hidden neurons was 60. The result of WNN based prediction model is obtained for 16 hidden neurons. Table 1 demonstrates the offline prediction results of different models.

In second experiment, using 58 fuzzy rules the learning of FWNN for pr=84 have been performed. The value of RMSE=0.0114 and NDEI=0.046. Table 2 demonstrates the offline prediction results of different models used for Mackey-Glass time-series. As shown FWNN prediction error is lower than other models.

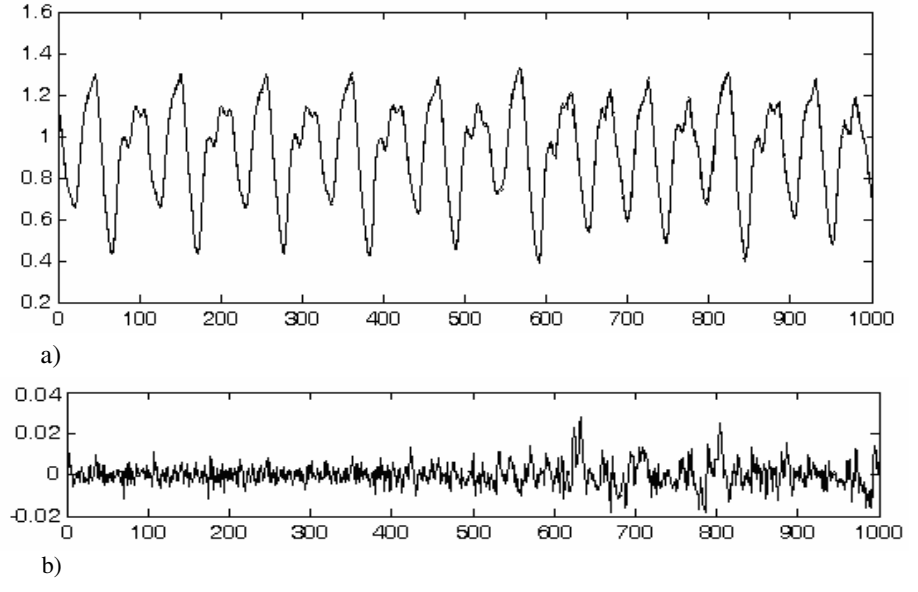


Fig. 4. a) Six-step ahead prediction for Mackey-Glass time-series for t=117-1118, b) Pre-diction error (first 500 points is used for training, second 500 data points for testing)

Table 1. Six step ahead prediction results for Mackey-Glass time-series

Method	Epochs	Testing NDEI
Feed-forward Neural Network	10000	0.041
Linear Predictive method	2000	0.55
WNN	2000	0.031
FWNN	1000	0.015

Table 2. Prediction results for pr=84

Method	Epochs	Testing NDEI
FWNN	500	0.046
Cascaded-Correlation NN	500	0.32
Sixth-order polynomial	500	0.85
Back-Propagation NN	500	0.05
Linear Predictive method	2000	0.60

4.2 Modelling of Electricity Consumption

A number of studies [19-21] have been published about modelling electricity consumption using econometric models. Electricity consumption models using climatic variables [19], using weather and population [20], using economical and weather variables [21] have been considered. These models are required for variety of utility activities and need measuring the number of climatic and economical variables. Sometimes obtaining the values of these variables is very difficult and these are not enough for accurate model development. In this problem one of main goal is to meet customer needs in future and organize the planning of utilities. In this study the FWNN is used to construct electricity consumption prediction model.

Cyprus imports petroleum from the abroad. The statistical data were obtained from KIB-TEK company reports for the period of 1996-2005. Problem was to determine volume of electricity consumed in the near future.

Five input data points [x(t-12) x(t-6) x(t-5) x(t-2) x(t)] are used as input for prediction model. The output training data corresponds to x(t+12). In other word since the electricity consumption is taken monthly, the value that is to be predicted will be after pr=12 months. The training input/output pairs for the prediction system will be a five dimension input vector, and one dimension predicted output vector.

To start the training, the FWNN structure is generated. It includes five input neurons and one output neuron. The 16 hidden neurons are used in hidden layer of neuro-fuzzy part of FWNN. Eleven neurons are used in hidden layer of WNN network. The initial values of membership function are generated in equally spaced and cover the whole input space. Membership functions are Gaussian functions. The training of the parameters was performed by using learning algorithms described in section 3.

For training of the system the statistical data describing monthly electricity consumption from January 1996 to December 2004 are taken. The data from January 2005 to December 2005 are taken for diagnostic testing. All input and output data are scaled in the interval [0 1]. The training is carried out for 1000 epochs. Once the FWNN has been successfully trained, it is then used for prediction the 2005 monthly electricity consumption. The training and test values of NDEI were 0.2251 and 0.2401 correspondingly. In fig. 6(a) the output of FWNN system for twelve-step ahead prediction of electricity consumption for learning and generalization step is shown. Here solid line is desired output, dashed line is FWNN output. Fig. 6(b) demonstrates twelve-step ahead prediction of FWNN for 2005. The plot of prediction error is shown in fig. 7. As shown from figure in generalization step (end part of error curve) the value of error is increased. The result of simulation of FWNN prediction model is

compared with result of simulation of NN based prediction model. In table 3 the comparative results of simulations are given. As shown from table the performance of FWNN prediction is better than performance of NN model. The obtained result from the simulation satisfies the efficiency of application of FWNN technology in constructing prediction model.

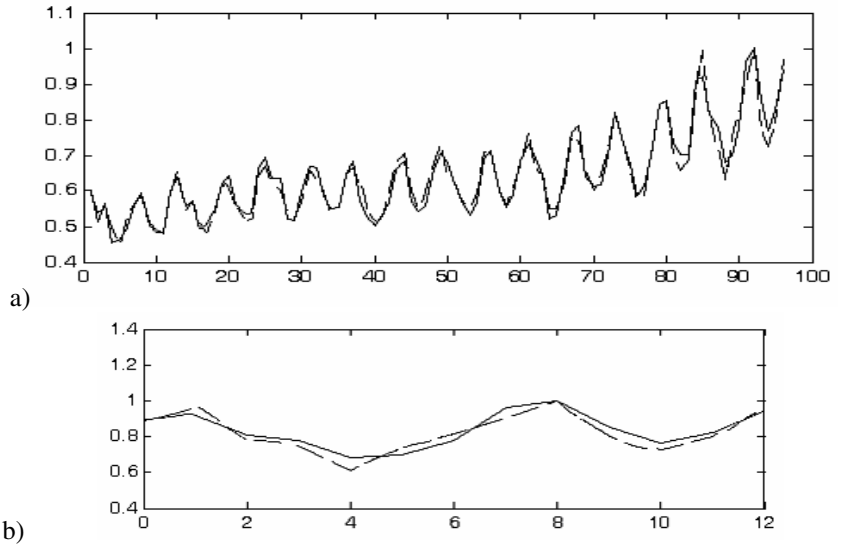


Fig. 6. Twelve step ahead prediction by FWNN (dotted line) and predicted signal (solid line). a) Curves describing learning and testing data together, b) curves describing testing data.

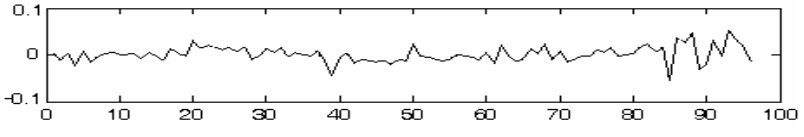


Fig. 7. Plot of prediction error

Table 3. Comparative results of simulation

	Number of rules	Epochs	RMSE	NDEI
Feedforward Neural Network	60	10000	0.03768	0.3187
FWNN Model	16	1000	0.02837	0.2401
FWNN Model	58	1000	0.01713	0.1449

5 Conclusion

In this paper FWNN is developed for time-series prediction. By using wavelets the fuzzy rules are constructed. The gradient algorithm is applied for learning parameters

of premise and consequent parts of fuzzy rules in FWNN structure. The structure and learning algorithms of FWNN system is applied for modeling and prediction of complex time-series. The developed FWNN structure is also applied for predicting future values of electricity consumption. This process is high order nonlinear. Using statistical data the prediction model is constructed. The simulation results are compared with the results of simulations of other prediction models. Comparative results demonstrate that the FWNN prediction model has better performance than other models.

References

[1] G.E.P. Box, G.M.Jenkins, G.C.Reinsel. Time-series analysis, Forecasting and Control. Third edition. Prentice-Hall, Inc. Englewood Cliffs, NJ 07632 (1994)
[2] G.S.Maddala. Introduction to econometrics. Englewood Cliffs,NJ: Prentece-Hall (1996)
[3] Smaoui N. An Artificial Neural Network Noise Reduction Method for Chaotic Attractors. Intern J. Computer Math., Vol.73 (2000) 417-431.
[4] R.S.Crowder III. Predicting the Mackey-Glass time series with cascade correlation learning. In D.Touretzky, G.Hinton and T.Sejnowski, eds., Proceeding of the Connectionist Models Summer School, Carnegic Mellon University (1990) 117-123
[5] Nunnari G, Nucifora A, Randieri C.The application of neural techniques to the modelling of time series of atmospheric pollution data. Ecological Modelling 111 (1998) 187-205
[6] Hill, T., O'Connor,M., Remus, W. Artificial neural network models for forecasting and decision making. Int. Journal of Forecasting, 10 (1994) 5-15.
[7] Tang Z., de Almeida C., Fishwick P.A.. Time-series forecasting using neural network versus Box-Jenkins methodology. Simulation, 57 (1991) 303-310.
[8] Kugarajah T, Q.Zhang. Multidimensional wavelet frames. IEEE Transaction on Neural Networks 6 (1995) 1552-1556.
[9] Szu H., Telfer B., Garcia J. Wavelet Transforms and Neural Networks for Compression and recognition. Neural Networks 9 (1996) 695-708.
[10] Chui, C. K. An Introduction to Wavelets. Academic Press, New York, NY, U.S.A (1992)
[11] Y. Cheng, B. Chen, F. Shiau. Adaptive Wavelet Network Control Design for Nonlinear Systems. *Proc. Natl. Sci. Counc. ROC(A),* Vol. 22, No. 6 (1998) 783-799
[12] Chang P.R., Weihui Fu, Minjun Yi. "Short term load forecasting using wavelet networks". Engineering Intel.Syst. for Electrical Engineering and Communications 6 (1998)217-223
[13] L.Cao, Y.Hong, H.Fang, G.He. "Predicting Chaotic time-series with wavelet networks", Physica D (1995) 225-38.
[14] Abdel-Aal RE,Al-Garni AZ,Al-Nassar YN. Modelling and forecasting monthly electricity consumption in eastern Saudi Arabia using abductive networks.Energy 22 (1997).
[15] Yan YY. Climate and residential electricity consumption in Hong Kong. Energy; 23(1) (1998) 17-20.
[16] Rajan M,Jain VK.Modelling of electrical energy consumption in Delhi.Energy,24: (1999)
[17] Thuillard M. Fuzzy logic in the wavelet framework. Proc. Toolmet'2000, Oulu (2000)
[18] Rahib H.Abiyev. Controller Based of Fuzzy Wavelet Neural Network for Control of Technological Processes. IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, Giardini Naxos, Italy (2005).

OFDM Channel Equalization Based on Radial Basis Function Networks

Giuseppina Moffa

IUSS, Institute for Advanced Studies of Pavia
giusi.moffa@bristol.ac.uk

Abstract. The universal approximation property makes neural networks very attractive for system modelling and identification. Channel estimation and equalization for digital communications are good examples. We explore the application of a Radial Basis Function Network to approximate the frequency response of a wireless channel, under the settings established by the IEEE 802.11 family of standards for wireless LAN architecture. We aim to exploit the channel impulse response correlation in the frequency domain to reduce the effect of noise. We obtain a smoother reconstructed function than by using a single tap Zero Forcing frequency domain equalizer. This is achieved by using a smaller number of basis functions, in the approximating Radial Basis Function Network, than the number of sub-carriers used by the OFDM modulation technique adopted in the transmission system. Although the training of the network following the Least Squares criterion requires the inversion of a matrix, this is feasible given the relatively small number of sub-carriers in the WLAN. Simulations show that the proposed algorithm behaves considerably better with respect to a simple single tap Zero Forcing algorithm, by reducing the bit error rate by more than a half. We also outline a possible solution based on the Kalman filter to update the network parameters adaptively and thus exploit any time correlation of the channel impulse response.

Keywords: channel equalization, OFDM modulation, Radial Basis Function.

1 Introduction

Classical approaches to channel estimation are statistical approaches based on the maximum likelihood, maximum a posteriori (MAP) or minimum mean-squared error (MMSE) criteria. Improvements of these techniques keep being investigated; as an example, Baccarelli and Galli [1] introduce a new algorithm for blind de-convolution based on the MAP method and a nonlinear Kalman like estimator. Applications of the Expectation-Maximization algorithm [2], have also recently been considered for channel estimation in high rate wireless data communication systems utilizing transmitter diversity [3,4,5,6]. Approaches based on the minimization of the error-entropy have also been investigated [7,8], as

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 201–210, 2006.

opposed to the popular MMSE criterion. This type of information theoretic criterion has also been considered as a training process for a multi-layer perceptron based equalization scheme for nonlinear channel models [9].

The universal approximation property [10] explains the numerous neural network based adaptive equalizers that have have been introduced in the literature to overcome inter-symbol interference, non-linear distortion and filter noise [11]. Radial Basis Function (RBF) based implementations of the Bayesian equalizer for a time-invariant channel have been suggested [12,13,14], and a complex valued version has been considered for being structurally similar to the optimal Bayesian equalizer [15,16].

Applications of Fuzzy Logic Systems to communications have also been widely studied and techniques to implement a Bayesian equalizer or eliminate co-channel interference have been proposed [17,18]. Type-2 Fuzzy Adaptive Filters have also been suggested for the equalization of non-linear time-varying channels [19,20,21]. Nonlinear channel models are frequently encountered in data transmission over digital satellite links because of the nonlinear behaviour of Traveling Wave Tube amplifiers when working close to saturation.

It is worth noting here that due to the prohibitive computational complexity associated with the optimal solution under the Bayesian framework, great effort has been put in developing sub-optimal solutions for signal reception problems. However the Bayesian Monte Carlo methodologies which have recently gained much interest in statistics outline feasible solutions that can approach the theoretical optimum [22,23].

This paper is organized as follows. In section 2 the specific scenario studied is described, with a particular focus on OFDM modulation (§2.1), while section 3 describes the implementation of a Recursive Least Squares (RLS) algorithm in this setting. Section 4 presents a possible application of RBF networks to OFDM channel equalization; we also outline a proposal for an adaptive updating of the network coefficients by means of the Kalman filter (§4.3).

2 Description of the Scenario

A key problem in communications is the efficient and reliable transmission of information signals over imperfect channels. Given the random nature of radio channels they are typically treated by means of stochastic models, based on measurements made for a specific environment [24]. Due to reflection and diffraction the electromagnetic waves travel from the transmitter to the receiver following different paths, leading to multipath fading. Coupled with the fast varying nature of the channel this results in a signal at the receiver antenna whose amplitude and phase can vary significantly making the accurate reconstruction of the transmitted signal a very challenging task.

2.1 OFDM Modulation

OFDM modulation [25,26] relies on the division of the channel bandwidth into several narrow (therefore slower) sub-channels. The frequency response for each

of them turns out to be relatively flat, and spreading the signal over different sub-carriers increases its robustness to possible forms of impulse noise. For these reasons OFDM efficiently addresses the issues concerned with multipath reception. The modulation/demodulation can be practically performed numerically by resorting to inverse/direct Fast Fourier Transform algorithms with rather low computational complexity.

At the receiver the symbols can be expressed as:

$$Y[m,k] = X[m,k]H[m,k] + v[m,k]$$
$$m = 1,\ldots,N_{\mathrm{f}}, \quad k = 0,\ldots,N_{\mathrm{c}} - 1 \tag{1}$$

where $X[m,k]$ is the symbol (properly modulated) transmitted during the m-th time slot on the k-th sub-carrier, $H[m,k]$ is the frequency response of the transmission path for carrier number k during the m-th time slot, $v[m,k]$ is the complex noise component, N_{c} is the number of sub-carriers and N_{f} is the number of OFDM symbols in a time frame.

The OFDM scheme includes two types of pilot symbols to help channel estimation: OFDM symbols consisting entirely of pilot sub-carriers periodically transmitted and "scattered pilots" evenly distributed in each symbol, as shown in figure 1. The latter are usually meant for synchronization, to make the coherent detection robust against frequency offsets and phase noise. However one could think of deriving channel estimates for the scattered pilots and obtaining a frequency response for all sub-carriers by interpolation. Cui and Tellambura [27] suggest an adaptively trained RBF channel estimator for this purpose.

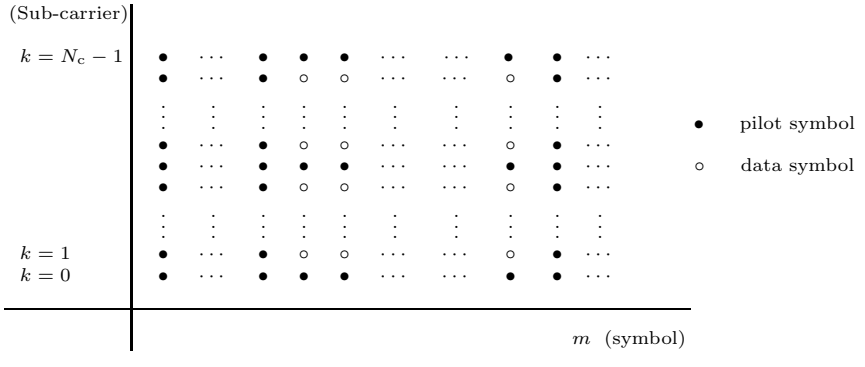


Fig. 1. OFDM transmission scheme

2.2 Wireless LAN

This work focuses on the WLAN architecture defined by the IEEE 802.11a standard, the first one to use OFDM in packet based communications [25,26]. Since the packet length is short enough to consider the channel constant, it seems that, for this specific kind of transmission systems, resorting to a preamble of pilots symbols is the most suitable approach to estimate the channel. Moreover only

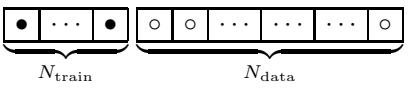
using scattered pilots would introduce unacceptable delays before the estimates converge to reasonable values.

If a sequence of training symbols is available, it follows from equation (1) that the frequency response can be estimated by

$$\widehat{H}[m,k] = Y[m,k]/X[m,k] \tag{2}$$

where $Y[m,k]$ is the observed output corresponding to the known transmitted symbol $X[m,k]$, hence the recovered symbol will be $\widehat{X}[m,k] = Y[m,k]/\widehat{H}[m,k]$. This actually corresponds to implement a single tap Zero Forcing (ZF) frequency domain equalizer, with complex coefficient $C_{k,\mathrm{ZF}} = 1/\widehat{H_k}$.

Assuming the preamble scheme for the pilots, the general structure of a time frame transmitted on each sub-carrier is as shown below:



N_{train}	Training symbols
N_{data}	Data symbols
$N_{\mathrm{f}} = N_{\mathrm{train}} + N_{\mathrm{data}}$	Frame size

The choice of the preamble length is obviously a tradeoff between a short training time and an accurate channel estimation. The estimates of the channel frequency response corresponding to the pilot symbols are simply:

$$\widehat{H}[lN_{\mathrm{f}} + i, k] = \frac{Y[lN_{\mathrm{f}} + i, k]}{X[lN_{\mathrm{f}} + i, k]} \qquad i = 1, \ldots, N_{\mathrm{train}}$$

where l is the frame index.

3 State-Space Model and RLS Algorithm

Since OFDM is designed on purpose to be extremely resistant to multipath and highly spectrally efficient, it conveniently meets the demands of high speed wireless communication systems. Nevertheless it still needs a proper equalization at the receiver for the signal to be adequately reconstructed.

If we consider the OFDM scenario described in the previous chapter and call $\widehat{H}[l,k]$ the mean of the frequency response over the training symbols:

$$\widehat{H}(l,k) = \frac{1}{N_{\mathrm{train}}} \sum_{i=1}^{N_{\mathrm{train}}} \widehat{H}(lN_{\mathrm{f}} + i, k) \tag{3}$$

the following state-space model can be defined for each sub-carrier k:

$$\begin{cases} H[l,k] = H[l-1,k] + w[l,k] \\ Y[l,k] = H[l,k]X[l,k] + v[l,k] \end{cases} \tag{4}$$

where $H[l,k]$ and $Y[l,k]$ are defined as in equation 3, $w[l,k]$ and $v[l,k]$ are the corresponding noise components.

A rich collection of algorithms to adapt filter coefficients and improve performances with respect to a simple ZF algorithm is available in the literature [28].

As an example we apply the classic RLS method for channel estimation; which results in the following estimator:

$$\widehat{H}(l+1,k) = \widehat{H}(l,k) + K_{\text{RLS}} \cdot [Y(l+1,k) - X(l+1)\widehat{H}(l,k)]$$

where the constant gain $K_{\text{RLS}} = K_{\text{value}} \cdot X(l+1,k)$ and $K_{\text{value}} \in (0,1]$. When $K_{\text{value}} = 1$ the equation yields $\widehat{H}(l+1,k) = Y(l+1,k)X(l+1)$ which corresponds to an RLS algorithm with a zero forgetting factor. This means the past is completely neglected and just the current observation is taken into account, and it might be the best choice in a very fast varying channel. On the other hand $K_{\text{value}} = 0$ gives a non-adaptive estimate, taking the channel as static.

Figure 2 shows the bit error rate (BER) against K_{value}, obtained with the following simulation parameters: doppler bandwidth $B_d = 100$Hz, SNR = 20dB, Bit Rate $R = 24$Mb/s. It is clear that the performances improve for higher values, indeed they can be considered equivalent for $0.7 \lesssim K_{\text{value}} \leq 1$; this is consistent with the time varying nature of the channel and leads to the conclusion that it is not worth keeping track of the past in such conditions.

It is worth comparing the RLS solution against the simple ZF algorithm for the optimal value of K_{value} (namely 0.7). From figure 3 we see that the RLS method behaves only slightly better, but it is likely that its performance improves for lower values of SNR and slowly time-varying channels, where the minimum value of the BER will probably be reached in correspondence of smaller K_{value}.

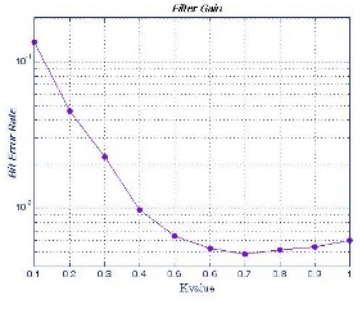


Fig. 2. RLS performance

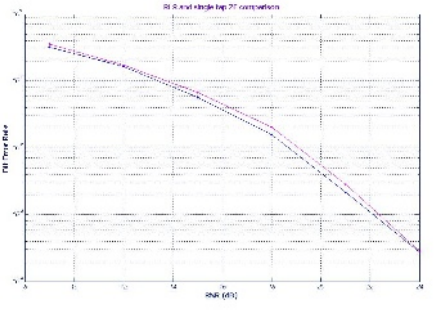


Fig. 3. Comparison of RLS and simple ZF

4 OFDM Equalization Via an RBF Network

The RLS based channel estimation depicted in the previous chapter only exploits the temporal correlation of the channel. Naturally one would try to take advantage of any correlation in the frequency domain as well, as suggested by Zhou and Wang [29]. Here we take a similar approach, by constructing a RBF Network (RBFN) which interpolates the frequency response of the channel.

4.1 RBF Networks

The attractiveness of neural networks arises from properties such as parallel distributed architecture, self-organization, adaptivity and the universal approximation characteristic already mentioned. A RBFN is a two-layer feed forward

network whose hidden layer is made of a set of basis functions, which produce a localized response to an input signal. The output is obtained as a linear combination of the basis functions evaluated in the hidden units.

Given a continuous non-linear function $\phi(\cdot) : \mathbb{R}^+ \rightarrow \mathbb{R}$, an RBFN with M neurons in the hidden layer and an F-dimensional input can be defined to implement a mapping $f : \mathbb{R}^F \rightarrow \mathbb{R}$, according to:

$$f(\mathbf{x}) = \sum_{j=1}^{M} \theta_j \phi(||\mathbf{x} - \mu_\mathbf{j}||, \sigma_j) \triangleq \sum_{j=1}^{M} \theta_j \phi_j(\mathbf{x})$$

where $\mathbf{x} \in \mathbb{R}^F$ is the input vector, $||\cdot||$ is the L_2 norm, $\{\theta_j\}_{j=1}^M$ is the vector of link weights, $\{\mu_j\}_{j=1}^M$ a vector representing the locations of the radial basis functions and $\{\sigma_j\}_{j=1}^M$ the vector of standard deviations, which determines the spread of the basis functions.

It is worth emphasizing that devising the most suitable architecture to model the underlying function of a certain mapping is not always obvious; the network topology is rather a parameter to be determined. An RBFN is thus specified by the hidden unit activation function, the number of processing units, a specific topology adequate for the problem at hand and a training algorithm to find the network parameters. The learning procedure is typically of an unsupervised form for the location update in the hidden layer and of a supervised form for the weight update in the ouput layer. The mean square error is often assumed as a cost function to evaluate the goodness of fitness.

A common choice for the non-linearity of the hidden layer is a Gaussian, which results in the mapping $f(\mathbf{x}) = \sum_{j=1}^{M} \theta_j exp(\frac{-||\mathbf{x}-\mu_\mathbf{j}||^2}{\sigma_j^2})$. It is clear that the maximum activation for each unit is achieved when the data sample coincides with the mean vector $\mu_\mathbf{j}$.

4.2 The Equalizer

We present here a simple RBFN based equalizer when the network has got a smaller number of basis functions than the number of sub-carriers of the transmission system. The network acts as a smoother of the frequency response of the channel thus filtering out some noise. Referring to the state-space model defined by equation (4) a RBFN can be defined as shown in figure 4. The real and the imaginary part of the frequency response estimate $\widetilde{H}$, are represented as follows:

$$\mathbb{R}e\{\widetilde{H}[l,k]\} = \sum_{j=1}^{M} \theta_j^R \phi_j(k), \qquad \mathbb{I}m\{\widetilde{H}[l,k]\} = \sum_{j=1}^{M} \theta_j^I \phi_j(k) \tag{5}$$

The inputs to the network are two real vectors:

$$\underline{H}_l^R \triangleq \mathbb{R}e\{(\widehat{H}[l,0], \ldots, \widehat{H}[l,N_c-1])\}', \quad \underline{H}_l^I \triangleq \mathbb{I}m\{(\widehat{H}[l,0], \ldots, \widehat{H}[l,N_c-1]]\}'$$

corresponding to the real and imaginary part of the estimates obtained from equation (3). The network is actually equivalent to a couple of independent

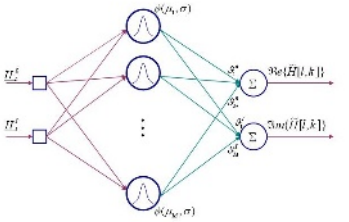


Fig. 4. Radial Basis Function Network

networks, for the real and imaginary part of the frequency response. In our simulations we have used gaussian basis functions. For simplicity's sake we have assumed their location is fixed at regular intervals, corresponding to one in three frequency indices. The standard deviations are assumed to be fixed as well and the same for all the functions. The weights θ_j^R and θ_j^I can then be derived by resorting to a supervised learning algorithm simply based on the Least Squares method, that is by minimizing the sums of squared residuals:

$$C_R = \sum_{k=0}^{N_c-1} [\mathbb{R}e\{\widehat{H}[l,k]\} - \mathbb{R}e\{\widetilde{H}[l,k]\}]^2, \quad C_I = \sum_{k=0}^{N_c-1} [\mathbb{I}m\{\widehat{H}[l,k]\} - \mathbb{I}m\{\widetilde{H}[l,k]\}]^2$$

Figure 6 shows how the BER obtained by the RBF equalizer compares to the one given the simple ZF algorithm. It is clear that the RBFN behaves better; at SNR = 24 dB for example the BER falls from 2.8×10^{-4} to 1.3×10^{-4}, which means it has been reduced by more than 50%. Performances can probably be improved further by a fine tuning of the spread σ. Indeed if the spread is too small or too big the network will provide an accurate approximation only for the frequencies corresponding to the location of the basis functions, on the other hand it will perform poorly for the interpolated values, resulting in an interpolated function which is too peaky or too flat with respect to the true shape of the function. Figure 5 shows a plot of the performances for different value of the standard deviation suggesting that $\sigma = 4$ is the optimal value for the given topology.

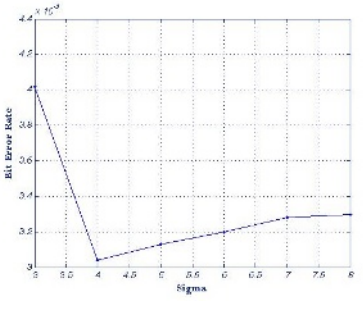


Fig. 5. BER vs basis function spread

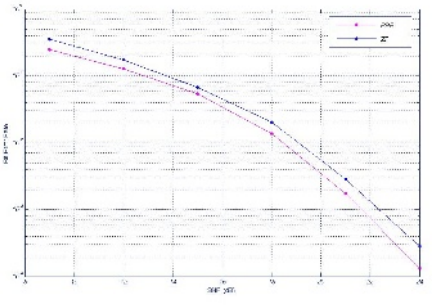


Fig. 6. Comparison of RBFN and ZF

4.3 Adaptive Algorithm Based on Kalman Filter

In order to exploit the frequency and the time correlation of the channel, at the same time, the RBF coefficients can be adaptively updated, for example by means of a Kalman filter. Given the RBF network structure described by equation (5), and by replacing the true value of the channel frequency response $H[m,k]$ with the estimate $\widetilde{H}[m,k]$, the output symbols at the receiver can be written as:

$$Y(m,k) = \left(\sum_{j=1}^{M} \theta_j^R \phi_j(k) + i \sum_{j=1}^{M} \theta_j^I \phi_j(k)\right) X(m,k) + v[m,k] \qquad 0 \le k \le N_c - 1$$

By defining the matrix

$$\mathbf{\Phi} \triangleq \begin{bmatrix} \phi_1(0) & \dots & \phi_M(0) \\ \vdots & \ddots & \vdots \\ \phi_1(N_c - 1) & \dots & \phi_M(N_c - 1) \end{bmatrix}$$

and the vectors

$$\underline{\theta}^R \triangleq (\theta_1^R, \dots, \theta_M^R)', \qquad \underline{\theta}^I \triangleq (\theta_1^I, \dots, \theta_M^I)'$$

the output vector $\underline{Y}(m) = [Y[m,0], \dots, Y[m,N_c - 1]]'$ can be written as:

$$\underline{Y}(m) = \mathbf{\Phi}(\underline{\theta}^R + i\underline{\theta}^I)\underline{X}'(m) + \underline{\nu}(m)$$

where $\underline{X}(m) = [X[m,0], \dots, X[m,N_c - 1]]'$ is the vector of data that make up the m-th OFDM symbol and $\underline{\nu}(m)$ is a complex gaussian process with zero mean and a given correlation matrix $\mathbf{\Sigma}_\nu$. Then the following state-space model can be defined:

$$\begin{cases} \underline{H}_l^R = \mathbf{\Phi}\underline{\theta}^R(l) + \underline{\eta}_l \\ \underline{\theta}^R(l+1) = \underline{\theta}^R(l) + \underline{w}_l \end{cases}$$

where $\underline{\eta}_l$, $\underline{w}_l$ are gaussian processes with zero mean and covariance matrix $\mathbf{\Sigma}_\eta = \sigma_\eta \mathbf{I}$, $\mathbf{\Sigma}_w = \sigma_w \mathbf{I}$ respectively. The Kalman estimator for the state $\underline{\theta}^R$ has the form:

$$\widehat{\underline{\theta}}^R(l+1) = \widehat{\underline{\theta}}^R(l) + K_{\text{Gain}}(\underline{H}_l^I - \mathbf{\Phi}\widehat{\underline{\theta}}^I(l))$$

where $K_{\text{Gain}} = \frac{\mathbf{\Phi}'}{\mathbf{\Phi}\mathbf{\Phi}' + \sigma_\eta^2/p}$ is the Kalman gain and p is the solution of the Riccati equation. For the imaginary part of the frequency response a completely analogous state-space model can be defined and the corresponding Kalman estimator can be straightforwardly derived.

5 Conclusions

We have explored an RBFN approach for OFDM channel equalization in a WLAN environment, as defined by the standard IEEE 802.11a. We have built

a neural network to approximate the channel frequency response, which aims to exploit the frequency domain correlation by using a smaller number of basis functions than the number of sub-carriers in the OFDM system. It acts as a smoother of the channel frequency response estimated by a single tap ZF frequency domain equalizer, mitigating the noise disturbance. Simulations have shown that the proposed algorithm behaves considerably better than a simple single tap ZF equalizer. Since channels encountered in WLAN environment typically do not suffer from very fast fading, further improvements could be obtained by adaptively improving the network parameters and exploiting the time correlation, rather than starting a new estimation procedure at every packet. We have outlined a solution based on the Kalman filter for this purpose. As a further step it would also be interesting to investigate the possibility of estimating the channel by only making use of the "scattered pilots" in the OFDM symbols, rather than resorting to the training sequences.

Acknowledgments

This work has been developed in association with The Mathworks, Turin. The author would also like to thank Giuseppe De Nicolao and Riccardo Bellazzi for useful discussions and guidance.

References

1. Baccarelli, E., Galli, S.: A new approach based on "soft statistics" to the nonlinear blind-deconvolution of unknown data channels. IEEE Transactions on Signal Processing (2001)
2. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from icomplete data via the em algorithm. Journal of the Royal Statistical Society (1977)
3. Alamouti, S.M.: A simple transmit diversity technique for wireless communications. IEEE Journal on Select Areas in Communications (1998)
4. Xie, Y., Georghiades, C.N.: An em-based channel estimation algorithm for ofdm with transmission deversity. IEEE Global Telecommunications Conference, GLOBECOM '01 (2001)
5. Xie, Y., Georghiades, C.N.: Two em-type channel estimation algorithms for ofdm with transmitter diversity. IEEE International Conference on Acoustics, Speech, and Signal Processing, Proceedings (ICASSP '02) (2002)
6. Panayirci, E., Aygolu, U., Pusane, A.: An em-based sequence estimation for wireless sistems with orthogonal transmit deversity. IEEE International Conference on Communications, ICC '03 (2003)
7. Santamaria, I., Erdogmus, D., Principe, J.C.: Entropy minimization for supervised digital communications channel equalization. IEEE Transactions on Signal Processing (2002)
8. Santamaria, I., Erdogmus, D., Principe, J.C.: An entropy minimization algorithm for digital communications channel equalization. IEEE Transactions on Signal Processing (2000)

9. Erdogmus, D., Rende, D., Principe, J.C., Wong, T.F.: Nonlinear channel equalization using multilayer perceptrons with information-theoretic criterion. Neural Networks for Signal Processing XI, Proceedings of the 2001 IEEE Signal Processing Society Workshop (2001)
10. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. Neural Computation **3** (1991)
11. Ibnkahla, M.: Applications of neural networks to digital communications - a survey. Signal Processing (1997)
12. Chen, S., Mulgrew, B., Grant, P.M.: A clustering technique for digital communications channel equalization using radial basis function networks. IEEE Transactions on Neural Networks (1993)
13. Chen, S., Mulgrew, B.: Overcoming co-channel interference using an adaptive radial basis function equalizer. Signal Processing (1992)
14. Charalabopoulos, G., Stavroulakis, P., Aghavami, A.H.: A frequency-domain neural network equalizer for ofdm. Globecom 2003, Wireless Communications Symposium (2003)
15. Chen, S., McLaughlin, S., Mulgrew, B.: Complex-valued radial basis function networks, part i: Network architecture and learning algorithms. Signal Processing (1994)
16. Chen, S., McLaughlin, S., Mulgrew, B.: Complex-valued radial basis function networks, part ii: Application to digital communication channel equalization. Signal Processing (1994)
17. Patra, S.K., Mulgrew, B.: Efficient architecture for bayesian equalization using fuzzy filters. IEEE Transactions on Circuits and Systems-II: Anlalog and Digital Signal Processing (1998)
18. Patra, S.K., Mulgrew, B.: Fuzzy implementation of bayesian equalizerin the precence of intersymbol and cochannel interference. IEE Proceedings - Communications (1998)
19. Karnik, N.N., Mendel, J.M.: Type-2 fuzzy logic systems. IEEE Transactions on Fuzzy Systems (1999)
20. Liang, Q., Mendel, J.M.: Equalization of non-linear time varying channels using type-2 fuzzy adaptive filters. IEEE Transactions on Fuzzy Systems (2000)
21. Liang, Q., Mendel, J.M.: Overcoming time-varying co-channel interference using type-2 fuzzy adaptive filters. IEEE Transactions on Circuits and Systems-II: Anlalog and Digital Signal Processing (2000)
22. Wang, X., Chen, R., Liu, J.: Monte carlo bayesian signal processing for wireless communications. Journal of VLSI Signal Processing (2001)
23. Yang, Z., Wang, X.: A sequential monte carlo blind receiver for ofdm systems in frequency-selective fading channels. IEEE Transactions on Signal Processing (2002)
24. Rappaport, T.S.: Wireless Communications, Principles and Practice. 2 edn. Communications Engineering and Emerging Technologies Series. Prentice Hall (2002)
25. Bahai, A.R.S., Saltzberg, B.R., Ergen, M.: Multi-carrier Digital Communications: Theory and Applications of OFDM. 2 edn. Springer (2004)
26. van Nee, R., Prasad, R.: OFDM for wireless multimedia communications. Universal Personal Communications. Artech House (2000)
27. Cui, T., Tellambura, C.: Channel estimation for ofdm systems based on adaptive radial basis function networks. IEEE Vehicular Technology Conference (2004)
28. Haykin, S.: Adaptive Filter Theory. 4 edn. Information and System Sciences Series. Prentice Hall (2002)
29. Zhou, X., Wang, X.: Channel estimation for ofdm systems using adaptive radial basis function networks. IEEE Transactions on Vehicular Technology **52** (2003)

A Quasi-stochastic Gradient Algorithm for Variance-Dependent Component Analysis

Aapo Hyvärinen[1] and Shohei Shimizu[1,2]

[1] Helsinki Institute for Information Technology, University of Helsinki, Finland
[2] The Institute of Statistical Mathematics, Japan
http://www.cs.helsinki.fi/hiit_bru/index_neuro.html

Abstract. We discuss the blind source separation problem where the sources are not independent but are dependent only through their variances. Some estimation methods have been proposed on this line. However, most of them require some additional assumptions: a parametric model for their dependencies or a temporal structure of the sources, for example. In previous work, we have proposed a generalized least squares approach using fourth-order moments to the blind source separation problem in the general case where those additional assumptions do not hold. In this article, we develop a simple optimization algorithm for the least squares approach, or a quasi-stochastic gradient algorithm. The new algorithm is able to estimate variance-dependent components even when the number of variables is large and the number of moments is computationally prohibitive.

1 Introduction

In blind source separation methods, the observed signals $x_i(t)$ $(i = 1 \cdots m)$ are typically assumed to be linear mixtures of sources $s_j(t)$ $(j = 1 \cdots n)$. Let $\bar{a}_{ij}$ denote the coefficients in the linear mixing between the sources $s_j(t)$ and the observed signals $x_i(t)$. Then the mixing can be expressed as

$$x_i(t) = \sum_{j=1}^{n} \bar{a}_{ij} s_j(t). \tag{1}$$

The problem of blind source separation is now to estimate both the source signals $s_i(t)$ and the mixing coefficient $\bar{a}_{ij}$ based on observations of the $x_i(t)$ alone [1].

The model (1) is called independent component analysis (ICA) model if $s_j(t)$ are assumed to be non-gaussian and independent [2]. The ICA model has been extensively studied for last two decades, and many estimation techniques for the model are available [3].

Recently, many extensions of the ICA model have started to be considered [4,5,6]. A quite interesting extension among them is the case where the source signals are not independent but dependent only through their variances [6]. To model such dependencies, [7] assumed that each source signal $s_i(t)$ can be represented as a product of two random signals $v_i(t)$ and $y_i(t)$:

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 211–220, 2006.

$$x_i(t) = \sum_{j=1}^{n} \bar{a}_{ij} v_j(t) y_j(t), \tag{2}$$

where $v_i(t)$ and $y_i(t)$ are independent, $y_i(t)$ are independent over time and are mutually independent of each other. No assumption on the distribution of $y_i(t)$ is made other than $y_i(t)$ have zero means. The variance signals $v_i(t)$ are nonnegative signals giving general activity levels and are allowed to be statistically dependent. Thus, the $v_i(t)$ could produce dependencies between sources $s_i(t) = v_i(t)y_i(t)$. No particular assumptions on the dependencies between $v_i(t)$ are made. This setting was called double-blind source separation problem because one neither observes the source signals $s_i(t)$ nor postulates a parametric model of their dependencies.

In [7], it was further assumed that the source signals have some time dependencies (autocorrelations) and a method was proposed that uses the time structure of the observed signals for separating the source signals. The time dependency assumption is the key to the method, and the method is not applicable to the case where the source signals are not temporally structured and has a more limited domain of applications, since many kinds of data do not have temporal structure in practice.

In [8], estimating functions for the model (2) was studied, and the quasi maximum likelihood estimation that requires no time dependencies was proposed. However, one has to appropriately choose the nonlinearity depending on whether the underlying independent signals $y_i(t)$ are supergaussian or subgaussian as in maximum likelihood methods for the ordinary ICA model. Moreover, they have to make certain extra assumptions on the signs of certain complicated nonlinear cross-moments of the sources, and it is not very clear when these are fulfilled.

In previous work [9], we proposed a generalized least squares approach using second- and fourth-order moment structures of observed signals in the general case where no temporal structure is available and it is unknown whether the underlying signals are supergaussian or subgaussian. However, its optimization using the ordinary gradient descent method is more difficult for larger variables since the number of moments increases enormously. In this paper, we provide a computationally efficient algorithm, or a *quasi*-stochastic gradient algorithm.

2 Model

We shall define the following model, which we will refer to as variance-dependent component analysis (VDCA) here. Let us collect the source signals in a vector $\boldsymbol{s} = [s_1, \cdots, s_n]^T$, and also construct the observed signal vector $\boldsymbol{x}$ in the same manner. (We omit the time indices in the subsequent part since we do not consider time structures.) Let us further collect the mixing coefficients in a matrix $\bar{\mathbf{A}} = [\bar{a}_{ij}]$. The VDCA model for the m-dimensional observed vector $\boldsymbol{x}$ is written as

$$\boldsymbol{x} = \bar{\mathbf{A}} \boldsymbol{s}, \tag{3}$$

where non-gaussian components s_i can be expressed as products of two signals v_i and y_i, $s_i = v_i y_i$, as in (2), where the y_i are zero-mean and mutually independent,

and that the set of the y_i is independent from the set of the v_j. No assumptions on the dependencies of the v_j with each other are made. An important point in the VDCA model is that no temporal structure is assumed, which is different from [7]. Here, we further assume $\bar{\mathbf{A}}$ to be square, which is a typical assumption in blind source separation [3].

An Illustrative Example

To illustrate the VDCA model, let us consider two stereotypical signals for which ordinary ICA does not work but VDCA does work. Let us define v_1, v_2, y_1 and y_2 as follows:

$$v_1 = 0.2 + \exp\{-4(t-7)^2\} + 0.5\exp\{-4(t-4)^2\} \tag{4}$$
$$v_2 = 0.2 + \exp\{-4(t-6.8)^2\} + 0.5\exp\{-4(t-4.2)^2\} \tag{5}$$
$$y_1 = \sin(50t) \tag{6}$$
$$y_2 = \cos(37t) \tag{7}$$
$$(t = 0, 0.01, 0.02, \cdots, 10).$$

Then we define variance-dependent signals $s_1 = v_1 y_1$, $s_2 = v_2 y_2$. Here, the underlying signals y_i are subgaussian and variance signals v_i are highly correlated. See Figure 1 for the original source signals s_i, estimated sources s_i by VDCA and FastICA with the hyperbolic tangent nonlinearity [10].

The point is that ICA tries to find a maximally non-gaussian linear combination of the source signals. Now it finds two conflicting goals: in the source

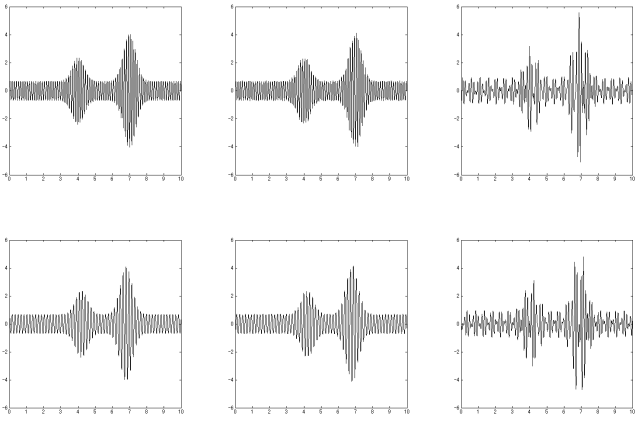


Fig. 1. Top left and bottom left: the original sources. Top center and bottom center: the estimated sources by VDCA. Top right and bottom right: the estimated sources by FastICA (tanh). The quasi-stochastic gradient algorithm with the stepsize 0.01 was run 10 times, and the estimates with the smallest value of the objective function were taken to avoid getting stuck in local minimum. Then our algorithm separated 100% of the sources (100 replications), whereas FastICA worked poorly (2%).

signals, the sinusoids y_i inside the envelopes are subgaussian, hence the original signals maximize subgaussianity inside the envelopes. In contrast, modulation by v_i make the signals s_i supergaussian, and hence an ICA algorithm should maximize supergaussianity to maximize non-gaussianity. This conflict between sub- and super-gaussianity makes ICA fail.

3 A Generalized Least Squares Approach

In previous work [9], we have proposed the generalized least squares approach (GLS) in estimation that utilizes higher-order moment to estimate $\bar{\mathbf{A}}$ in (3).

Let us denote by $\boldsymbol{\sigma}_2(\boldsymbol{\tau})$ the vector that consists of elements of the covariance matrix based on the model where any duplicates due to symmetry have been removed and by $\boldsymbol{\sigma}_4(\boldsymbol{\tau})$ the vector that consists of the tensor of fourth-order (cross-) moments where duplicate entries have been removed and by $\boldsymbol{\tau}$ the vector of source statistics and mixing coefficients that uniquely determines the second- and fourth-order moment structures of the model $\boldsymbol{\sigma}_2(\boldsymbol{\tau})$ and $\boldsymbol{\sigma}_4(\boldsymbol{\tau})$. Then the $\boldsymbol{\sigma}_2(\boldsymbol{\tau})$, $\boldsymbol{\sigma}_4(\boldsymbol{\tau})$ and $\boldsymbol{\tau}$ can be written as

$$\boldsymbol{\sigma}_i(\boldsymbol{\tau}) = \mathbf{H}_i E[\, \overbrace{\boldsymbol{x} \otimes \cdots \otimes \boldsymbol{x}}^{i \text{ times}} \,] \quad (i = 2, 4), \tag{8}$$

where the symbol $\otimes$ denotes the Kronecker product[1] and $\mathbf{H}_i$ is a selection matrix of order $\binom{m+i-1}{i} \times m^i$ $(i = 2, 4)$ that selects non-duplicated elements. The parameter vector $\boldsymbol{\tau}$ consists of $\bar{\mathbf{A}}$ and $E(s_p^2 s_q^2)$.

In [9], we proposed that the model is estimated using the principle of generalized least-square estimation. This is a method of matching the moments of the observed data $\boldsymbol{m}_i$ and those based on the model $\boldsymbol{\sigma}_i(\boldsymbol{\tau})$ in a weighted least-squares sense $(i = 2, 4)$.

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ be a random sample from the VDCA model as defined in Section 2, and define the sample counterparts to the moments in (8) as

$$\boldsymbol{m}_i = \frac{1}{N} \mathbf{H}_i \sum_{t=1}^{N} \overbrace{\boldsymbol{x}_t \otimes \cdots \otimes \boldsymbol{x}_t}^{i \text{ times}} \quad (i = 2, 4). \tag{9}$$

Let us denote by $\boldsymbol{\tau}_0$ the true parameter vector. The $\boldsymbol{\sigma}_i(\boldsymbol{\tau}_0)$ can be estimated by the $\boldsymbol{m}_i$ when N is enough large: $\boldsymbol{\sigma}_i(\boldsymbol{\tau}_0) \approx \boldsymbol{m}_i$ $(i = 2, 4)$.

The GLS estimator of $\boldsymbol{\tau}$ is obtained as

$$\widehat{\boldsymbol{\tau}} = \arg\min_{\boldsymbol{\tau}} \left\| \begin{bmatrix} \boldsymbol{m}_2 \\ \boldsymbol{m}_4 \end{bmatrix} - \begin{bmatrix} \boldsymbol{\sigma}_2(\boldsymbol{\tau}) \\ \boldsymbol{\sigma}_4(\boldsymbol{\tau}) \end{bmatrix} \right\|^2_{\widehat{\mathbf{U}}^{-1}}. \tag{10}$$

(For simplicity, the norm $\boldsymbol{y}^T \mathbf{M} \boldsymbol{y}$ of a vector $\boldsymbol{y}$ associated with a nonnegative definite matrix $\mathbf{M}$ is here expressed as $\|\boldsymbol{y}\|^2_{\mathbf{M}}$.) Here $\widehat{\mathbf{U}}$ is a weight matrix in

[1] The Kronecker product $\mathbf{X} \otimes \mathbf{Y}$ of matrices $\mathbf{X}$ and $\mathbf{Y}$ is defined as a partitioned matrix with (i, j)-th block equal to $x_{ij}\mathbf{Y}$.

GLS estimation and converges in probability to a certain positive definite matrix $\mathbf{U}$. The resultant GLS estimator $\widehat{\boldsymbol{\tau}}$ determined by (10) is then consistent and asymptotic normal [11]. We simply take the identity matrix as $\widehat{\mathbf{U}}$ in the following.

4 A Quasi-stochastic Gradient Algorithm

In this section, we propose a simple optimization algorithm for the least squares approach above. We assume that the data is prewhitened by a whitening matrix $\mathbf{V}$ in an ordinary way [3] and denote by $\boldsymbol{z} = \mathbf{V}\boldsymbol{x}$ the prewhitened signals. Then we can constrain $\mathbf{A} = \mathbf{V}\bar{\mathbf{A}}$ to be orthogonal, which stabilizes the algorithm below.

The total objective function (10) in the GLS approach becomes monstrous because we have sum over all the moments[2], and it only works for small dimensions. Denote by $\widetilde{\boldsymbol{a}}_i$ the i-th row of $\mathbf{A}$ and by $\mathbf{C}$ a symmetric matrix whose (i,j)-th element is $E(s_i^2 s_j^2)$. (Note that $\boldsymbol{\tau}$ consists of the elements of $\mathbf{A}$ and the lower triangular elements of $\mathbf{C}$.) A simple way to solve this problem would be to consider the objective function as a sum over the variable indices i, j, k, l:

$$\sum_{i,j,k,l} J_{ijkl}(\mathbf{A}, \mathbf{C}), \tag{11}$$

where

$$J_{ijkl}(\mathbf{A}, \mathbf{C}) = \left\{ \frac{1}{N} \sum_{t=1}^{N} z_{it} z_{jt} z_{kt} z_{lt} - E(\widetilde{\boldsymbol{a}}_i \boldsymbol{s}, \widetilde{\boldsymbol{a}}_j \boldsymbol{s}, \widetilde{\boldsymbol{a}}_k \boldsymbol{s}, \widetilde{\boldsymbol{a}}_l \boldsymbol{s}) \right\}^2 . \tag{12}$$

Let us compute the gradient of J_{ijkl} with respect to $\mathbf{A}$ and $\mathbf{C}$, denoted by $\nabla_{\mathbf{A}} J_{ijkl}$ and $\nabla_{\mathbf{C}} J_{ijkl}$, respectively (see Appendix A for the complete formulas). We can now update the estimate of $\mathbf{A}$ and $\mathbf{C}$ by taking *random* indices i, j, k, l at each iteration and using a simple gradient descent for J_{ijkl} (see Step 4 in the algorithm below). At each gradient step, we take new random indices i, j, k, l. This kind of a stochastic gradient descent finds the minimum of the sum of the J_{ijkl} that we wanted to minimize in the first place, because the gradient is *on the average* the same as the gradient of the whole sum.

To improve the convergence, it is quite useful to perform a projection of the gradient on the tangent surface of the set of orthogonal matrices [12]. This means replacing the gradient $\nabla_{\mathbf{A}} J_{ijkl}$ by

$$\nabla^{ort}_{\mathbf{A}} J_{ijkl} = \nabla_{\mathbf{A}} J_{ijkl} - \mathbf{A}(\nabla_{\mathbf{A}} J_{ijkl})^T \mathbf{A}. \tag{13}$$

Thus, the estimation consists of the following steps:

0. Remove the mean from the data and whiten it. Choose (random) initial values for the matrices $\mathbf{A}$ and $\mathbf{C}$.
1. Randomly choose four indices i, j, k, l.

[2] The number of fourth-order moments is of order n^4.

2. Compute the gradients with respect to $\mathbf{A}$ and $\mathbf{C}$ as given in Appendix A.
3. Compute the projected gradient with respect to $\mathbf{A}$ by (13).
4. Do a gradient step

$$\mathbf{A} \leftarrow \mathbf{A} - \mu \nabla^{ort}_{\mathbf{A}} J_{ijkl} \tag{14}$$

$$\mathbf{C} \leftarrow \mathbf{C} - \mu \nabla_{\mathbf{C}} J_{ijkl}, \tag{15}$$

   where μ is a small stepsize constant.
5. Orthogonalize $\mathbf{A}$ by

$$\mathbf{A} \leftarrow (\mathbf{A}\mathbf{A}^T)^{-1/2}\mathbf{A}. \tag{16}$$

The five steps 1-5 are repeated until $\mathbf{A}$ and $\mathbf{C}$ have converged. Then we obtain the estimate of $\bar{\mathbf{A}}$ by $\mathbf{V}^{-1}\mathbf{A}$.

5 Simulations

We conducted simulations to study the empirical performance of the algorithm above. The simulation consisted of 100 source separation trials with three different methods: 1) the quasi-stochastic gradient algorithm proposed in the paper; 2) FastICA using kurtosis and 3) FastICA using hyperbolic tangent function [10]. For the two FastICA, the symmetric orthogonalization was made. (The FastICA with the symmetric orthogonalization using hyperbolic tangent function as the nonlinearity is basically the same as the quasi-maximum likelihood estimation [13].) We took 0.1 as the stepsize and stopped the quasi-stochastic gradient iteration when the *average* change of orthogonalized mixing matrices measured by $1 - \min\{\mathrm{diag}(\mathbf{A}^T_{old}\mathbf{A}_{new})\}$ over the last 100 iterations is smaller than 0.0001[3].

In each trial, we generated 10 sources that were dependent through their variances and created observed signals following the VDCA model as defined in Section 2. First, we created a random signal v_0 with several sample sizes (3,000, 5,000, 10,000, 30,000) where their components were independently distributed according to the gaussian distribution with zero mean and unit variance. Outliers, defined as values larger than a threshold of 3 times the standard deviation, were eliminated from the resulting signals by reducing their values to the above-mentioned threshold. The variance signals v_i were then defined as the absolute values of the signal, that is, $v_i = |v_0|$ $(i = 1, \cdots, 10)$. The variance signals were completely dependent on each other since they were identical, but they were independent over time. (Therefore, the double-blind method [7] that used temporal correlations was not applicable to this case.)

Next the source signals s_i were created by multiplying the variance signals v_i by ten-dimensional random signals y_i, that is, $s_i = v_i y_i$. Here, the ten underlying signals y_i were i.i.d. (white) zero-mean subgaussian random processes to create enough variance dependencies [7]. (The subgaussian signals were signed fourth root of zero mean-uniform variables.) The source signals were normalized to have

[3] Here, the quasi-stochastic gradient algorithm was run once for each data.

zero means and unit variances. Finally, a random mixing matrix $\bar{\mathbf{A}}$ was created, and the signals were mixed to provide the observed signals $x_i, i = 1, \cdots, 10$.

The three methods were then applied on the data after prewhitening it. The performance of each method was assessed as follows. Denoting by $\mathbf{W}$ the transpose of the obtained estimate of the orthogonalized mixing matrix $\mathbf{A}$ (with permutation and sign indeterminacies), we looked at the matrix $\mathbf{W}\mathbf{V}\bar{\mathbf{A}}$. We computed how many elements in this matrix had an absolute value that was larger than 0.90. First of all, it must be noted that the matrix $\mathbf{W}\mathbf{V}\bar{\mathbf{A}}$ is rather exactly orthogonal (up to insignificant errors occurred in the estimation of the whitening matrix), so there can be no more than 10 such elements in the matrix, and no row or column can contain more than one such element. In the ideal case where $\mathbf{W}\mathbf{V}\bar{\mathbf{A}}$ is a signed permutation matrix, there would be exactly 10 such elements. Thus, this gave a measure of how many source signals had been separated.

The results are shown in Table 1. Our method separated more than 97.0% of the components for the reasonable sample sizes (5,000, 10,000, 30,000). On the other hand, both FastICAs could not separate the components at all (0%) since FastICA is based on independence of sources. Thus, our method was quite good, while not being perfect.

Table 1. Percentage of components recovered (100 replications)

	Sample size			
	3,000	5,000	10,000	30,000
Stoc. grad. alg.	87.4	97.6	97.8	97.6
FastICA (kurtosis)	0	0	0	0
FastICA (tanh)	0	0	0	0

6 Conclusions

We proposed a quasi-stochastic gradient algorithm for the GLS approach using second- and fourth-order moment structures of observed signals to the blind source separation of sources that are dependent only through their variances. In the approach, we do not have to assume that the sources have some temporal structures nor postulate any parametric models for their dependencies. This could be a big advantage of our approach over the conventional methods.

Although our method works well in simulations, moment-based methods often suffer from sensitivity to outliers when applied on certain kinds of real data. An important question for future research is to investigate how serious this problem is and, eventually, how it can be alleviated.

Acknowledgements

This work was partially carried out at Division of Mathematical Science, Osaka University and Transdisciplinary Research Integration Center, Research Organization of Information and Systems. A.H. was supported by the Academy of

Finland through an Academy Research Fellow Position and project #203344. S.S. was supported by Grant-in-Aid for Scientific Research from the Japan Society for the Promotion of Science.

References

1. Jutten, C., Hérault, J.: Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. Signal Processing **24** (1991) 1–10
2. Comon, P.: Independent component analysis. a new concept? Signal Processing **36** (1994) 62–83
3. Hyvärinen, A., Karhunen, J., Oja, E.: Independent component analysis. Wiley, New York (2001)
4. Bach, F.R., Jordan, M.I.: Tree-dependent component analysis. In: Proc. the 18th Conference on Uncertainty in Artificial Intelligence (UAI-2002). (2002)
5. Hyvärinen, A.: A unifying model for blind separation of independent sources. Signal Processing **85** (2005) 1419–1427
6. Hyvärinen, A., Hoyer, P.O., Inki, M.: Topographic independent component analysis. Neural Computation **13** (2001) 1525–1558
7. Hyvärinen, A., Hurri, J.: Blind separation of sources that have spatiotemporal dependencies. Signal Processing **84** (2004) 247–254
8. Kawanabe, M., Müller, K.R.: Estimating functions for blind separation when sources have variance-dependencies. In: Proc. 5th International Conference on ICA and Blind Source Separation, Granada, Spain. (2004) 136–143
9. Shimizu, S., Hyvärinen, A., Kano, Y.: A generalized least squares approach to blind separation of sources which have variance dependency. In: Proc. IEEE Workshop on Statistical Signal Processing (SSP2005). (2005)
10. Hyvärinen, A.: Fast and robust fixed-point algorithms for independent component analysis. IEEE Trans. on Neural Networks **10** (1999) 626–634
11. Ferguson, T.S.: A method of generating best asymptotically normal estimates with application to estimation of bacterial densities. Annals of Mathematical Statistics **29** (1958) 1046–1062
12. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM Journal on Matrix Analysis and Applications **20** (1998) 303–353
13. Hyvärinen, A.: The fixed-point algorithm and maximum likelihood estimation for independent component analysis. Neural Processing Letters **10** (1999) 1–5

A Gradient of the Objective Function

The gradients of the objective function in (12) are as follows:

$$\nabla_{\mathbf{A}} J_{ijkl} = -2\left\{\frac{1}{N}\sum_{t=1}^{N} z_{it}z_{jt}z_{kt}z_{lt} - E(z_i z_j z_k z_l)\right\}\frac{\partial E(z_i z_j z_k z_l)}{\partial \mathbf{A}} \tag{17}$$

$$\nabla_{\mathbf{C}} J_{ijkl} = -2\left\{\frac{1}{N}\sum_{t=1}^{N} z_{it}z_{jt}z_{kt}z_{lt} - E(z_i z_j z_k z_l)\right\}\frac{\partial E(z_i z_j z_k z_l)}{\partial \mathbf{C}}. \tag{18}$$

In what follows, we provide $E(z_i z_j z_k z_l)$ that were given by the VDCA model and their first derivatives with respect to $\mathbf{A}$ and $\mathbf{C}$ to compute $\nabla_{\mathbf{A}} J_{ijkl}$ and $\nabla_{\mathbf{C}} J_{ijkl}$ above.

We first provide the model-based expectations $E(z_i z_j z_k z_l)$:

$$\begin{aligned}
E(z_i^4) &= \sum_p a_{ip}^4 E(s_p^4) + 6\sum_{p<q} a_{ip}^2 a_{iq}^2 E(s_p^2 s_q^2)\\
E(z_i^3 z_j) &= \sum_p a_{ip}^3 a_{jp} E(s_p^4) + 3\sum_{p<q}(a_{ip}^2 a_{iq} a_{jq} + a_{ip} a_{jp} a_{iq}^2) E(s_p^2 s_q^2)\\
E(z_i^2 z_j z_k) &= \sum_p a_{ip}^2 a_{jp} a_{kp} E(s_p^4) + \sum_{p<q}(a_{ip}^2 a_{jq} a_{kq} + 2a_{ip} a_{jp} a_{iq} a_{kq}\\
&\quad + 2a_{ip} a_{kp} a_{iq} a_{jq} + a_{iq}^2 a_{jp} a_{kp}) E(s_p^2 s_q^2)\\
E(z_i^2 z_j^2) &= \sum_p a_{ip}^2 a_{jp}^2 E(s_p^4) + \sum_{p<q}(a_{ip}^2 a_{jq}^2 + a_{iq}^2 a_{jp}^2 + 4a_{ip} a_{jp} a_{iq} a_{jq}) E(s_p^2 s_q^2)\\
E(z_i z_j z_k z_l) &= \sum_p a_{ip} a_{jp} a_{kp} a_{lp} E(s_p^4) + \sum_{p<q}(a_{ip} a_{jp} a_{kq} a_{lq} + a_{ip} a_{jq} a_{kp} a_{lq}\\
&\quad + a_{ip} a_{jq} a_{kq} a_{lp} + a_{iq} a_{jq} a_{kp} a_{lp} + a_{iq} a_{jp} a_{kq} a_{lp} + a_{iq} a_{jp} a_{kp} a_{lq}) E(s_p^2 s_q^2).
\end{aligned}$$

Next, we give the first derivatives:

$$\begin{aligned}
&\frac{\partial E(z_i^4)}{\partial a_{ip}} = 4a_{ip}^3 E(s_p^4) + 12\sum_{q\neq p} a_{ip} a_{iq}^2 E(s_p^2 s_q^2),\\
&\frac{\partial E(z_i^4)}{\partial E(a_{rp})} = 0\ (r \neq i, l),\ \frac{\partial E(z_i^4)}{\partial E(s_p^4)} = a_{ip}^4,\ \frac{\partial E(z_i^4)}{\partial E(s_p^2 s_q^2)} = 6a_{ip}^2 a_{iq}^2\\
&\frac{\partial E(z_i^3 z_j)}{\partial a_{ip}} = 3a_{ip}^2 a_{jp} E(s_p^4) + 3\sum_{q\neq p}(2a_{ip} a_{iq} a_{jq} + a_{jp} a_{iq}^2) E(s_p^2 s_q^2)\\
&\frac{\partial E(z_i^3 z_j)}{\partial a_{jp}} = a_{ip}^3 E(s_p^4) + 3\sum_{q\neq p} a_{ip} a_{iq}^2 E(s_p^2 s_q^2)\\
&\frac{\partial E(z_i^3 z_j)}{\partial a_{rp}} = 0\ (r \neq i, j),\ \frac{\partial E(z_i^3 z_j)}{\partial E(s_p^4)} = a_{ip}^3 a_{jp}\\
&\frac{\partial E(z_i^3 z_j)}{\partial E(s_p^2 s_q^2)} = 3(a_{ip}^2 a_{iq} a_{jq} + a_{ip} a_{jp} a_{iq}^2)
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial E(z_i^2 z_j z_k)}{\partial a_{ip}} &= 2a_{ip}a_{jp}a_{kp}E(s_p^4) + \sum_{q\neq p}(2a_{ip}a_{jq}a_{kq} + 2a_{jp}a_{iq}a_{kq} + 2a_{kp}a_{iq}a_{jq})E(s_p^2 s_q^2) \\
\frac{\partial E(z_i^2 z_j z_k)}{\partial a_{jp}} &= a_{ip}^2 a_{kp}E(s_p^4) + \sum_{q\neq p}(2a_{ip}a_{iq}a_{kq} + a_{iq}^2 a_{kp})E(s_p^2 s_q^2) \\
\frac{\partial E(z_i^2 z_j z_k)}{\partial a_{kp}} &= a_{ip}^2 a_{jp}E(s_p^4) + \sum_{q\neq p}(2a_{ip}a_{iq}a_{jq} + a_{iq}^2 a_{jp})E(s_p^2 s_q^2) \\
\frac{\partial E(z_i^2 z_j z_k)}{\partial a_{rp}} &= 0 \ (r \neq i,j,k), \ \frac{\partial E(z_i^2 z_j z_k)}{\partial E(s_p^4)} = a_{ip}^2 a_{jp}a_{kp} \\
\frac{\partial E(z_i^2 z_j z_k)}{\partial E(s_p^2 s_q^2)} &= a_{ip}^2 a_{jq}a_{kq} + 2a_{ip}a_{jp}a_{iq}a_{kq} + 2a_{ip}a_{kp}a_{iq}a_{jq} + a_{iq}^2 a_{jp}a_{kp} \\
\frac{\partial E(z_i^2 z_j^2)}{\partial a_{ip}} &= 2a_{ip}a_{jp}^2 E(s_p^4) + \sum_{q\neq p}(2a_{ip}a_{jq}^2 + 4a_{jp}a_{iq}a_{jq})E(s_p^2 s_q^2) \\
\frac{\partial E(z_i^2 z_j^2)}{\partial a_{jp}} &= 2a_{ip}^2 a_{jp}E(s_p^4) + \sum_{q\neq p}(2a_{iq}^2 a_{jp} + 4a_{ip}a_{iq}a_{jq})E(s_p^2 s_q^2) \\
\frac{\partial E(z_i^2 z_j^2)}{\partial a_{rp}} &= 0 \ (r \neq i,j), \ \frac{\partial E(z_i^2 z_j^2)}{\partial E(s_p^4)} = a_{ip}^2 a_{jp}^2 \\
\frac{\partial E(z_i^2 z_j^2)}{\partial E(s_p^2 s_q^2)} &= a_{ip}^2 a_{jq}^2 + a_{iq}^2 a_{jp}^2 + 4a_{ip}a_{jp}a_{iq}a_{jq} \\
\frac{\partial E(z_i z_j z_k z_l)}{\partial a_{ip}} &= a_{jp}a_{kp}a_{lp}E(s_p^4) + \sum_{q\neq p}(a_{jp}a_{kq}a_{lq} + a_{jq}a_{kp}a_{lq} + a_{jq}a_{kq}a_{lp})E(s_p^2 s_q^2) \\
\frac{\partial E(z_i z_j z_k z_l)}{\partial a_{jp}} &= a_{ip}a_{kp}a_{lp}E(s_p^4) + \sum_{q\neq p}(a_{ip}a_{kq}a_{lq} + a_{iq}a_{kq}a_{lp} + a_{iq}a_{kp}a_{lq})E(s_p^2 s_q^2) \\
\frac{\partial E(z_i z_j z_k z_l)}{\partial a_{kp}} &= a_{ip}a_{jp}a_{lp}E(s_p^4) + \sum_{q\neq p}(a_{ip}a_{jq}a_{lq} + a_{iq}a_{jq}a_{lp} + a_{iq}a_{jp}a_{lq})E(s_p^2 s_q^2) \\
\frac{\partial E(z_i z_j z_k z_l)}{\partial a_{lp}} &= a_{ip}a_{jp}a_{kp}E(s_p^4) + \sum_{q\neq p}(a_{ip}a_{jq}a_{kq} + a_{iq}a_{jq}a_{kp} + a_{iq}a_{jp}a_{kq})E(s_p^2 s_q^2) \\
\frac{\partial E(z_i z_j z_k z_l)}{\partial a_{rp}} &= 0 \ (r \neq i,j,k,l), \ \frac{\partial E(z_i z_j z_k z_l)}{\partial E(s_p^4)} = a_{ip}a_{jp}a_{kp}a_{lp} \\
\frac{\partial E(z_i z_j z_k z_l)}{\partial E(s_p^2 s_q^2)} &= a_{ip}a_{jp}a_{kq}a_{lq} + a_{ip}a_{jq}a_{kp}a_{lq} \\
&\quad + a_{ip}a_{jq}a_{kq}a_{lp} + a_{iq}a_{jq}a_{kp}a_{lp} + a_{iq}a_{jp}a_{kq}a_{lp} + a_{iq}a_{jp}a_{kp}a_{lq}.
\end{aligned}
$$

Two ICA Algorithms Applied to BSS in Non-destructive Vibratory Tests

Juan-José González de-la-Rosa[1], Carlos G. Puntonet[2], R. Piotrkowski, I. Lloret[1], and Juan-Manuel Górriz

[1] University of Cádiz, Research Group TIC168 - Computational Instrumentation and Industrial Electronics, EPSA, Av. Ramón Puyol S/N. 11202, Algeciras-Cádiz, Spain
juanjose.delarosa@uca.es
[2] University of Granada, Department of Architecture and Computers Technology, ESII, C/Periodista Daniel Saucedo. 18071, Granada, Spain
carlos@atc.ugr.es

Abstract. Two independent component analysis (ICA) algorithms have been applied for blind source separation (BSS) in a synthetic, multi-sensor scenario, within a non-destructive pipeline test. The first one, CumICA, is based in the computation of the cross-cumulants of the mixed observed signals, and needs the aid of a digital high-pass filter to achieve the same SNR (up to -40 dB) as the second algorithm, Fast-ICA. Vibratory signals were acquired by a wide frequency range transducer (100-800 kHz) and digitalized by a 2.5 MHz, 8-bit ADC. Different types of commonly observed source signals are linearly mixed, involving acoustic emission (AE) sequences, impulses and other parasitic signals modelling human activity. Both ICA algorithms achieve to separate the impulse-like and the AE events, which often are associated to cracks or sudden non-stationary vibrations.

1 Introduction

Vibratory and acoustic emission (AE) signal processing usually deals with separation of multiple events which sequentially or simultaneously occur in different measurement points during a non-destructive test. In most situations, the tests involve the study of the behavior of secondary events, or reflections, resulting from an excitation (the main event). These echoes carry information related with the medium through which they propagate, as well as surfaces where they reflect [1].

But, in almost every measurement scenario, an acquired sequence contains information regarding not only the AE under study, but also additive noise processes (mainly from the measurement equipment) and other parasitic signals, e.g. originated by human activity or machinery vibrations. As a consequence, in non-favorable SNR cases, BSS should be accomplished before characterization [2], in order to obtain the most reliable spectral *fingerprint* of the AE event.

The purpose of this paper is twofold. First we show how two ICA algorithms separate the true AE event from the parasitics, taking a multi-sensor array of

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 221–229, 2006.

inputs (SNR=-40 dB). Secondly, we compare performances of Cum-ICA and Fast-ICA, resulting that Cum-ICA needs the aid of a post high-pass filter to achieve the same SNR as Fast-ICA. This comparison could be interesting for a future implementation of the code in an automatic test system.

The paper is structured as follows: in Section 2 we make a brief progress report on the characterization of vibratory emissions. Section 3 summarizes the ICA models and outlines their properties. Results are displayed in section 4. Finally, conclusions and achievements are drawn in section 5.

2 Acoustic Emission Signal Processing

Elastic energy travels through the material as a stress wave and is typically detected using a piezoelectric transducer, which converts the surface displacement (vibrations) to an electrical signal. AE signal processing is used for the detection and characterization of failures in non-destructive testing and identification of low-level biological signals [2]. Most AE signals are non-stationary and they consist of overlapping bursts with unknown amplitude and arrival time. These characteristics can be described by modelling the signal by means of neural networks, and using wavelet transforms [1],[3]. These second-order techniques have been applied in an automatic analysis context of the estimation of the time and amplitude of the bursts. Multiresolution has proven good performance in de-noising (up to SNR=-30 dB, with modelled signals) and estimation of time instances, due to the selectivity of the wavelets filters banks [4].

Higher order statistics (HOS) have enhanced characterization in analyzing biological signals due to the capability for rejecting noise [5]. This is the reason whereby HOS could be used as part of an ICA algorithm.

3 The ICA Model and Algorithms

3.1 Outline of ICA

BSS by ICA is receiving attention because of its applications in many fields such as speech recognition, medicine and telecommunications [6]. Statistical methods in BSS are based in the probability distributions and the cumulants of the mixtures. The recovered signals (the source estimators) have to satisfy a condition which is modelled by a contrast function. The underlying assumptions are the mutual independence among sources and the non-singularity of the mixing matrix [2],[7].

Let $\mathbf{s}(t) = [s_1(t), s_2(t), \ldots, s_m(t)]^T$ be the transposed vector of sources (statistically independent). The mixture of the sources is modelled via

$$\mathbf{x}(t) = \mathbf{A} \cdot \mathbf{s}(t) \tag{1}$$

where $\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_m(t)]^T$ is the available vector of observations and $\mathbf{A} = [a_{ij}] \in \Re^{m \times n}$ is the unknown mixing matrix, modelling the environment in which signals are mixed, transmitted and measured [8]. We assume that $\mathbf{A}$ is a

non-singular n×n square matrix. The goal of ICA is to find a non-singular n×m separating matrix $\mathbf{B}$ such that extracts sources via

$$\hat{\mathbf{s}}(t) = \mathbf{y}(t) = \mathbf{B} \cdot \mathbf{x}(t) = \mathbf{B} \cdot \mathbf{A} \cdot \mathbf{s}(t) \tag{2}$$

where $\mathbf{y}(t) = [y_1(t), y_2(t), \ldots, y_m(t)]^T$ is an estimator of the sources. The separating matrix has a scaling freedom on each row because the relative amplitudes of sources in $\mathbf{s}(t)$ and columns of $\mathbf{A}$ are unknown [7]. The transfer matrix $\mathbf{G} \equiv \mathbf{BA}$ relates the vector of independent (original) signals to its estimators.

3.2 CumICA

High order statistics, known as cumulants, are used to infer new properties about the data of non-Gaussian processes. Before, such processes had to be treated as if they were Gaussian, but second order statistics are phase-blind. The relationship among the cumulant of r stochastic signals and their moments of order $p, p \leq r$, can be calculated by using the *Leonov-Shiryayev* formula [9]:

$$\begin{aligned} Cum(x_1, ..., x_r) = \sum (-1)^k \cdot (k-1)! \cdot E\{\prod_{i \in v_1} x_i\} \\ \cdot E\{\prod_{j \in v_2} x_j\} \cdots E\{\prod_{k \in v_p} x_k\} \end{aligned} \tag{3}$$

where the addition operator is extended over all the set of v_i $(1 \leq i \leq p \leq r)$ and v_i compose a partition of *1,. . . ,r*.

A set of random variables are statistically independent if their cross-cumulants are zero. This is used to define a contrast function, by minimizing the distance between the cumulants of the sources $\mathbf{s}$(t) and the outputs $\mathbf{y}$(t). As sources are unknown, it is necessary to involve the observed signals. Separation is developed using the following contrast function based on the entropy of the outputs [2]:

$$H(\mathbf{z}) = H(\mathbf{s}) + log[det(\mathbf{G})] - \sum \frac{\mathbf{C}_{1+\beta, y_i}}{1+\beta} \tag{4}$$

where $\mathbf{C}_{1+\beta, y_i}$ is the $1+\beta$th-order cumulant of the ith output, $\mathbf{z}$ is a non-linear function of the outputs y_i, $\mathbf{s}$ is the source vector, $\mathbf{G}$ is the global transfer matrix of the ICA model and $\beta > 1$ is an integer verifying that $\beta + 1$-order cumulants are non-zero.

Using equation 4, the separating matrix can be obtained by means of the following recurrent equation [8]

$$\mathbf{B}^{(h+1)} = [\mathbf{I} + \mu^{(h)}(\mathbf{C}_{y,y}^{1,\beta} \mathbf{S}_y^{\beta} - I)]\mathbf{B}^{(h)} \tag{5}$$

where $\mathbf{S}_y^{\beta}$ is the matrix of the signs of the output cumulants. Equation 5 is interpreted as a quasi-Newton algorithm of the cumulant matrix $\mathbf{C}_{y,y}^{1,\beta}$. The learning rate parameters $\mu^{(h)}$ and η are related by:

$$\mu^{(h)} = \min(\frac{2\eta}{1+\eta\beta}, \frac{\eta}{1+\eta\|\mathbf{C}_{y,y}^{1,\beta}\|_p}) \tag{6}$$

with $\eta < 1$ to avoid $\mathbf{B}^{(h+1)}$ being singular; $\|.\|_p$ denotes de p-norm of a matrix. The adaptative equation 5 converges, if the matrix $\mathbf{C}_{y,y}^{1,\beta}\mathbf{S}_y^{\beta}$ tends to the identity.

3.3 FastICA

One of the independent components is estimated by $y = \mathbf{b}^T\mathbf{x}$. The goal of FastICA is to take the vector $\mathbf{b}$ that maximizes the non-Gaussianity (independence)of y, by finding the maxima of its negentropy [7]. The algorithm scheme is an approximative Newton iteration, resulting from the application of the *Kuhn-Tucker* conditions. This leads to the equation 7

$$E\{\mathbf{x}g(\mathbf{b}^T\mathbf{x}) - \beta\mathbf{b} = 0\} \tag{7}$$

where g is a non-quadratic function and β is an iteration parameter.

Provided with the mathematical foundations the experimental results are outlined.

4 Experimental Results

The inputs of the ICA algorithms comprise synthetics (laboratory mixtures), which have been obtained by mixing real AE events (the ones we are interested in getting the spectral *track*), impulse-like events, noise processes and damping sinusoids. The sensor used to capture the AE events was attached to the outer surface of the pipeline, which is under mechanical excitation.

A number of 20 AE events were captured. One of these vibratory signals is depicted in Fig. 1, where we can observe the main AE event and the secondary reflections or echoes.

Each digitalized sequence comprises 2502 points (sampling frequency of 2.5 MHz and 8 bits of resolution), and assembles the main AE event and the subsequent reflections (echoes).

Four types of sources have been considered and linearly mixed in the synthetics. These subsequent mixtures constitute the inputs of the algorithm: A real AE event, an uniform white noise (SNR=-40 dB), a damped sine wave and an impulse-like event. The damping sine wave models a mechanical vibration which may occur, e.g. as a consequence of a maintenance action. It has a damping factor of 2000 and a frequency of 8000 Hz. Finally, the impulse is included as a very common signal registered in vibration monitoring. Fig. 2 shows one possible input quartet.

One of the 20 results (output quartet) of CumICA is depicted in Fig. 3. The damping sinusoid is considered as a frequency component of the impulse-like event because IC3 and IC4 are almost the same. The final independent components are obtained filtering the independent components by a 5th-order *Butterworth* high-pass digital filter (20 kHz).

The resulting separated sources resulting from one of the Fast-ICA processing are depicted in Fig. 4.

Finally, to test the independence of the independent components, some relevant joint distributions have been included in Fig. 5 and in Fig. 6. The left

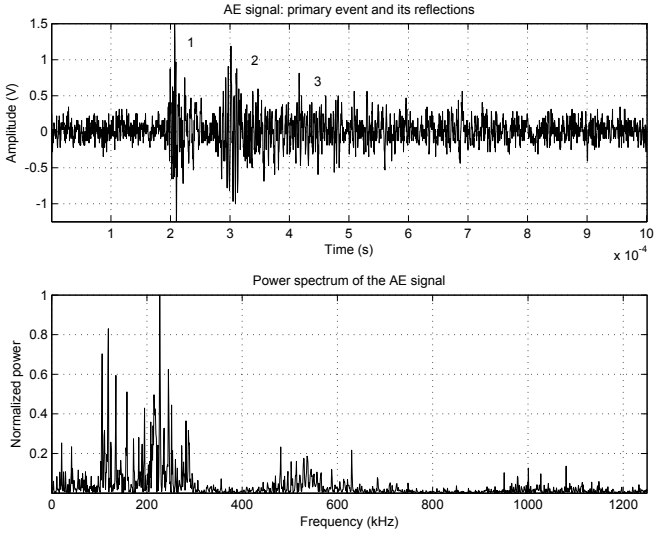


Fig. 1. One of the 20 AE events and its associated spectrum. Usually, these are the signals under study which constitute a main perturbation and its associated reflections. The main event (1) and two reflections (2,3) can be seen.

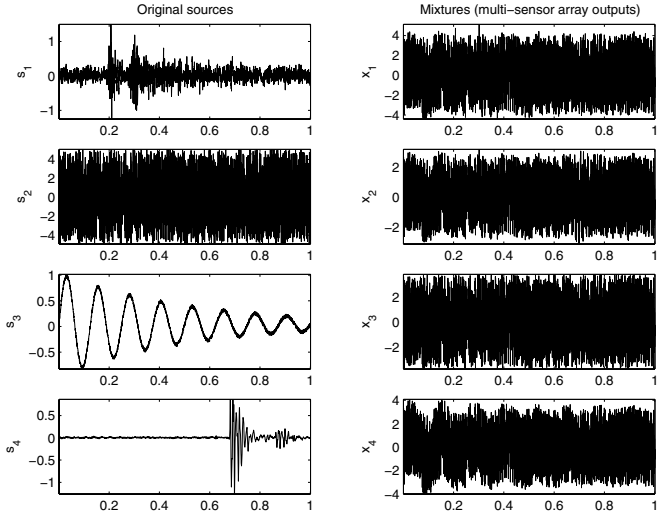


Fig. 2. Left column: One of the 20 quartets of original sources to be mixed, which in turn constitutes one of the 20 inputs to the ICA algorithms. Right column: The linear mixtures.

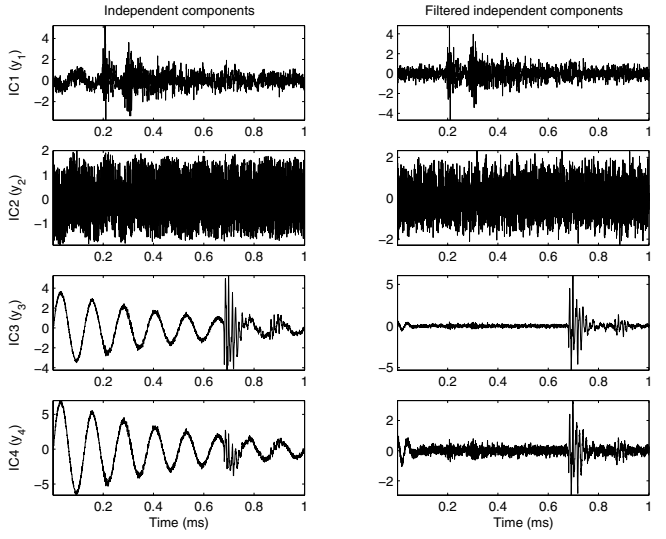


Fig. 3. Estimated and filtered sources via CumICA (ICs; Independent Components). Left column: AE event, noise, damping sine wave plus impulse, idem. Right column: Filtered signals.

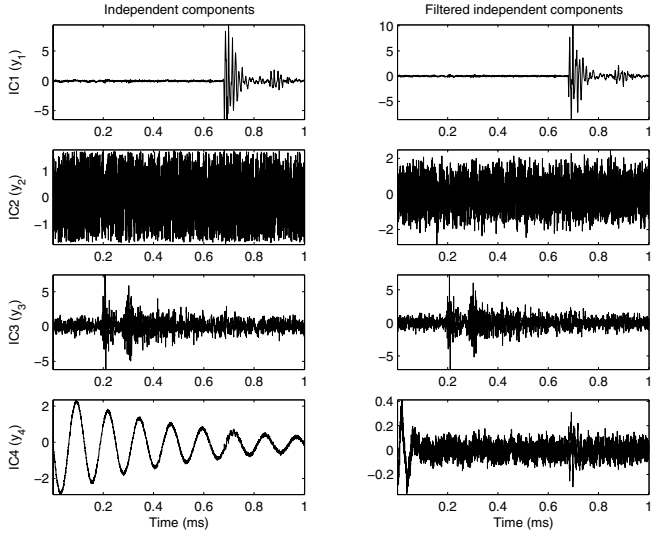


Fig. 4. Estimated and filtered sources (independent components, ICs) via FastICA. Right column (very similar to the left) top to bottom: Impulse, noise, AE event, noise. Post-filtering is not necessary to recover the AE event and the impulse.

column of both figures shows how for any IC, the values are quite random. This means that for a value (a point in the signal-to-signal graphic) of an IC, almost all the values of the another IC are allowed. On the other hand, the joint

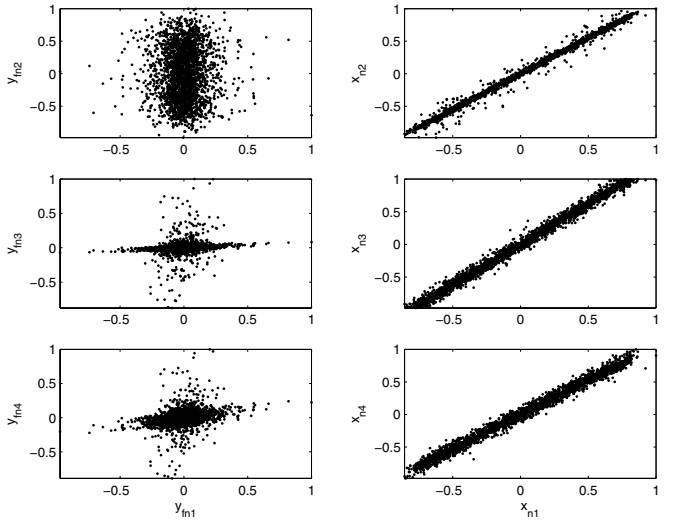


Fig. 5. Signal-to-signal diagram for the CumICA outputs. Left column: Independent components. Right column: Mixtures.

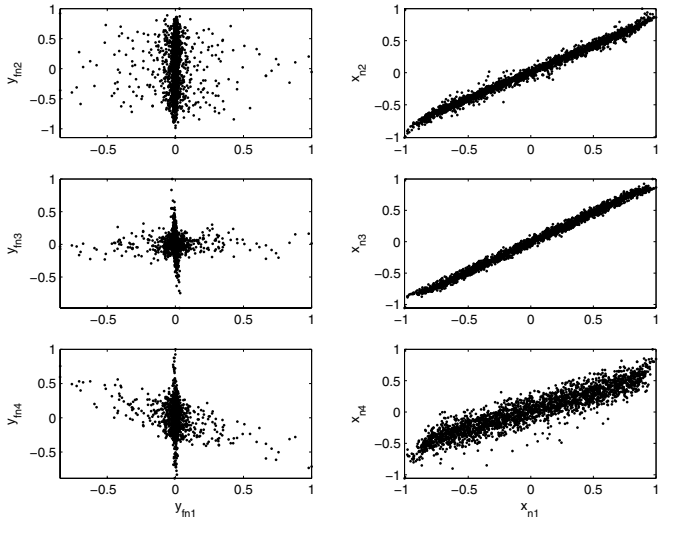


Fig. 6. Signal-to-signal diagram for the FastICA outputs. Left column: Independent components. Right column: Mixtures.

distributions of the mixtures are linearly shaped, which leads us to infer a dependency before separating sources by ICA.

These results lead us to conclude about the use of the algorithms.

5 Conclusions and Future Work

ICA is far different from traditional methods used to separate sources or to de-noise signals, as power spectrum or wavelet transforms, which obtain an energy diagram of the different frequency components, with the risk that low-level sounds or events could be masked. This experiment shows that both algorithms are able to separate the sources with small energy levels in comparison to the background noise. This is explained away by statistical independence basis of ICA, regardless of the energy associated to each frequency component. The post filtering action applied to Cum-ICA lets us work with very low SNR signals. FastICA kernel maximizes the non-Gaussianity, so it is not necessary a filter stage.

The next step regarding this research is oriented in a double direction. First, a stage involving four real mixtures will be developed. Secondly, and simultaneously, the computational complexity of the algorithms have to be reduced to perform a real implementation in a digital signal processor.

Acknowledgement

The authors would like to thank the *Spanish Ministry of Education and Science* for funding the projects DPI2003-00878, TEC2004-06096 and PTR1995-0824-OP.

References

1. De la Rosa, J.J.G., Lloret, I., Ruzzante, J., Piotrkowski, R., Armeite, M., Pumarega, M.L.: Higher-order characterization of acoustic emission signals. In: CIMSA 2005, Proceedings of the 2005 IEEE International Conference on Computational Intelligence for Measurement Systems and Applicacions, Giardini Naxos, Italy, 20-22 July 2005, ISBN 0-7803-9026-1; IEEE Catalog Number 05EX1037 (2005) 296–300 Paper CM5027. Oral Presentation in the Session 16 Advanced Signal Processing 2.
2. De la Rosa, J.J.G., Puntonet, C.G., Lloret, I.: An application of the independent component analysis to monitor acoustic emission signals generated by termite activity in wood. Measurement (Ed. Elsevier) **37** (2005) 63–76 Available online 12 October 2004.
3. Piotrkowski, R., Gallego, A., Castro, E., García-Hernández, M., Ruzzante, J.: Ti and Cr nitride coating/steel adherence assessed by acoustic emission wavelet analysis. Non Destructive Testing and Evaluation (NDT and E) International (Ed. Elsevier) **8** (2005) 260–267
4. De la Rosa, J.J.G., Puntonet, C.G., Lloret, I., Górriz, J.M.: Wavelets and wavelet packets applied to termite detection. Lecture Notes in Computer Science (LNCS) **3514** (2005) 900–907 Computational Science - ICCS 2005: 5th International Conference, GA Atlanta, USA, May 22-25, 2005, Proceedings, Part I.
5. Puntonet, C.G., De la Rosa, J.J.G., Lloret, I., Górriz, J.M.: Recognition of insect emissions applying the discrete wavelet transform. Lecture Notes in Computer Science (LNCS) **3686** (2005) 505–513 Third International Conference on Advances in Pattern Recognition, ICAPR 2005 Bath, UK, August 22-25, 2005, Proceedings, Part I.

6. Mansour, A., Barros, A.K., Onishi, N.: Comparison among three estimators for higher-order statistics. In: The Fifth International Conference on Neural Information Processing, Kitakyushu, Japan (1998)
7. Hyvärinen, A., Oja, E.: Independent Components Analysis: A Tutorial. Helsinki University of Technology, Laboratory of Computer and Information Science (1999)
8. De la Rosa, J.J.G., Puntonet, C.G., Górriz, J.M., Lloret, I.: An application of ICA to identify vibratory low-level signals generated by termites. Lecture Notes in Computer Science (LNCS) **3195** (2004) 1126–1133 Proceedings of the Fifth International Conference, ICA 2004, Granada, Spain.
9. Swami, A., Mendel, J.M., Nikias, C.L.: Higher-Order Spectral Analysis Toolbox User's Guide. (2001)

Reference-Based Extraction of Phase Synchronous Components

Jan-Hendrik Schleimer and Ricardo Vigário

Adaptive Informatics Research Centre
Helsinki University of Technology
P.O. Box 5400, FIN-02015 Espoo, Finland
schleime@cis.hut.fi, rvigario@cis.hut.fi

Abstract. Phase synchronisation is a phenomenon observed in measurements of dynamic systems, composed of several interacting oscillators. It can be quantified by the phase locking factor (PLF), which requires knowledge of the instantaneous phase of an observed signal. Linear sources separation methods treat scenarios in which measurements do not represent direct observations of the dynamics, but rather superpositions of underlying latent processes. Such a mixing process can cause spuriously high PLFs between the measurements, and camouflage the phase locking to a provided reference signal. The PLF between a linear projection of the data and a reference can be maximised as an optimisation criterion revealing the most synchronous source component present in the data, with its corresponding amplitude. This is possible despite the amplitude distributions being Gaussian, or the signals being statistically dependent, common assumptions in blind sources separation techniques without *a-priori* knowledge, *e.g.* in form of a reference signal.

1 Introduction

Interest in phase synchronisation phenomena has a long history, when studying the interaction of complex, natural or artificial, dynamic systems. A detailed documentation of the topic is given in Ref. [1]. Although not completely adopted, synchronisation was attributed a role in the interplay between different parts of the central nervous system (CNS) as well as across central and peripheral nervous systems. In that formulation, the elementary units are *self-sustained oscillators* $x_i(t)$, exhibiting stable limit cycles. If the coupling between the oscillators is of weak nature, any distortion that a mutual forcing would cause on the amplitudes, will be immediately compensated. Then the interactions of m self-sustained oscillators can be described with the Kuramoto model (*cf.* Ref. [2,3] for a review), solely on the phases dynamics

$$\dot{\phi}_i(t) = \omega_i(t) + \frac{1}{m}\sum_{j=1}^{m} \kappa_{ij} \sin\big(\phi_j(t) - \phi_i(t)\big), \tag{1}$$

where $\phi_i(t)$ and $\omega_i(t)$ denote the oscillators' instantaneous phases and frequencies; and κ can either be the scalar-valued global coupling strength or a matrix in

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 230–239, 2006.

which $[\boldsymbol{\kappa}]_{ij}$ describes the coupling between oscillators i and j. Postulated that the system in Eq. (1) is an adequate description of the dynamics of a phenomenon, it becomes meaningful to focus investigations of their interaction principles to phase synchronisation.

In the CNS, the basic unit — the *neuronal oscillator* — can be a single neuron, with an oscillating membrane potential, or a whole population of already synchronous neurons, that synchronises to another population at a different site of the brain. Examples of models for neuronal dynamics based on self-sustained oscillators can be found in Refs. [4,5].

The phase synchronisation is commonly quantified by the phase locking factor (PLF, for definition see Sec. 2). In many applications, direct measurements of the individual sources $\boldsymbol{x}(t)$ are not available, but instead global multi-sensor measurements $\boldsymbol{y}(t)$ of the whole system, which represent mixtures of $\boldsymbol{x}(t)$. See [6] for a general treatment of such problems. Often, this mixing process can be described by a linear transformation $\boldsymbol{y}(t) = \boldsymbol{A}\boldsymbol{x}(t)$. If the PLF is evaluated w.r.t. $\boldsymbol{y}(t)$ two problems arise: (*i*) calculating the PLF between observations $y_i(t)$ will, due to the presence of individual oscillators in several sensors, lead to an erroneous detection of interactions between them; (*ii*) since each sensor measurement contains more than one of the oscillators the PLF of the $y_i(t)$ to a reference will be reduced, obscuring the true interactions.

Here, an algorithm for the extraction of sources synchronised with a given reference is introduced (Sec. 2). The PLF is only evaluated in the source space, not for the observations, circumventing spurious synchronisations by cross-talk, and allowing the recovery of the true sources and their coupling strengths. The search for coupled oscillator networks is facilitated by the use of a reference signal, embodying existing information on the targeted networks. This can be a continuous stimulus to the complex system, an already extracted component of the system, or an external, more accessible part of the system. The algorithm is presented in a general gradient-based formulation, and can be applied to a variety of problems. Experimental results in a controlled simulated data set (Secs. 2.1,2.2), as well as in a preliminary investigation into cortico-muscular control are presented (Sec. 3).

2 Extraction of One Source Synchronised to a Reference

As stated above, assume that the observations result from a linear superposition of generative sources, $\boldsymbol{y}(t) = \boldsymbol{A}\boldsymbol{x}(t)$, with the restriction that $\boldsymbol{A}$ is invertible. The time index shall be discrete in the following and reside in a fixed interval $1 \leqslant t \leqslant T \in \mathbb{N}$. Further postulate that, for a given reference signal $u(t)$, a phase locking is taking place between the reference and at least one of the source signals, $x_i(t)$. Denote the analytic signals[1] as $\hat{y}_i(t) = Y_i(t)\, e^{\mathrm{i}\varphi_i(t)} = y_i(t) + \mathrm{i}\mathcal{H}[y_i](t)$, $\hat{u}(t) = U(t)\, e^{\mathrm{i}\psi(t)}$ and $\hat{s}(t) = S(t)\, e^{\mathrm{i}\phi(t)}$. $s(t)$ is the extracted source, an approximation of $x_i(t)$. For the phase difference between reference and the extracted source signal $\Delta\phi(t) = \phi(t) - \psi(t)$, define a function

[1] Here, $\mathcal{H}[x](t)$ is the Hilbert transform of a signal, $x(t)$.

$$\varrho\, e^{\mathrm{i}\Psi} = \frac{1}{T}\sum_{t=1}^{T} e^{\mathrm{i}\Delta\phi(t)} = \frac{1}{T}\sum_{t=1}^{T} \frac{\hat{s}(t)\hat{u}^*(t)}{|\hat{s}(t)\hat{u}(t)|}, \tag{2}$$

so that the amplitude ϱ measures the phase locking between the reference signal and the projection $s(t) = \boldsymbol{w}^{\mathsf{T}}\boldsymbol{y}(t)$. It is called the phase locking factor PLF and, since depending on the source signal, it is also a function of $\boldsymbol{w}$ and the data. Because the complex vector $\hat{s}(t)$ and $\hat{u}(t)$ in Eq. (2) are scaled to one, the PLF $\varrho \in \mathbb{R}$ lies in the interval $0 \leqslant \varrho \leqslant 1$. As the maximisation criterion for our algorithm we can use its square ϱ^2. The gradient w.r.t. $\boldsymbol{w}$ is given by the following expression

$$\nabla\varrho^2 = \frac{2\varrho}{T}\sum_{t=1}^{T} \frac{\sin(\Psi - \Delta\phi(t))}{S^2(t)}\boldsymbol{\Gamma}(t)\boldsymbol{w}, \tag{3}$$

with the amplitude ϱ and the mean phase Ψ as defined by Eq. (2), and a matrix $[\boldsymbol{\Gamma}(t)]_{ij} = Y_i(t)Y_j(t)\sin(\varphi_i(t) - \varphi_j(t))$, fully defined by the observations. The details of the derivation are shown in appendix A.

Eq. (9) can be used in a batch gradient ascent iteration to maximise ϱ. The learning rule reads

$$\Delta\boldsymbol{w} = \eta\,\frac{2\varrho}{T}\sum_{t=1}^{T} \frac{\sin(\Psi - \Delta\phi(t))}{S^2(t)}\boldsymbol{\Gamma}(t)\,\boldsymbol{w}. \tag{4}$$

For smoother convergence, the learning factor η can be chosen to decay in a variety of annealing strategies. Since $\varrho \leqslant 1$, a sufficient stopping criterion for the iteration, iff a phase locked component is present in the data, is $\varrho > 1 - \delta$ for $0 < \delta \ll 1$. A maximum number of iterations has to be specified, in case the reference signal has no phase locked component in the data, because then the objective function will not reach a high value. The batch algorithm is summarised in Algo. 1.

For larger data sets, with potential nonstationary phase locking behaviour, as can be produced by Eq. (1), the learning rule can be formulated in an online way

Algo. 1. Extraction of a phase locked component.

1: input: $\boldsymbol{y}(t)$, $\hat{u}(t)$, η, n_{itr};
2: init: $\boldsymbol{w} \sim \mathcal{N}(0, \boldsymbol{I})$; $k = 1$;
3: **repeat**
4: $\quad s(t) \leftarrow \boldsymbol{w}^{\mathsf{T}}\boldsymbol{y}(t)$;
5: $\quad \hat{s}(t) \leftarrow s(t) + \mathrm{i}\mathcal{H}[s](t)$;
6: $\quad P \leftarrow \frac{1}{T}\sum_t \hat{s}(t)\hat{u}(t)/|\hat{s}(t)\hat{u}(t)|$;
7: $\quad \Psi \leftarrow \mathrm{angle}(P)$; $\varrho \leftarrow |P|$;
8: $\quad \boldsymbol{\Delta} \leftarrow$ Eq. (9);
9: $\quad \boldsymbol{w} \leftarrow \boldsymbol{w} + \eta\boldsymbol{\Delta}$;
10: $\quad \boldsymbol{w} \leftarrow \boldsymbol{w}/\|\boldsymbol{w}\|$;
11: $\quad k \leftarrow k + 1$;
12: **until** $(\varrho > 1 - \delta) \wedge (\|\boldsymbol{\Delta}\| < \epsilon) \wedge (k > n_{\mathrm{itr}})$

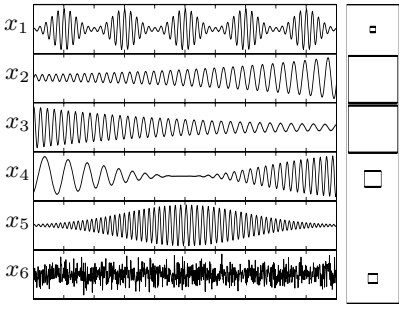


Fig. 1. Kurtosis values of the sources are $\mathrm{kurt}(x_1) = -0.02$, $\mathrm{kurt}(x_2) = 0.007$, $\mathrm{kurt}(x_3) = 0.02$, $\mathrm{kurt}(x_4) = 0.005$, $\mathrm{kurt}(x_5) = 0.006$ and $\mathrm{kurt}(x_6) = -0.18$

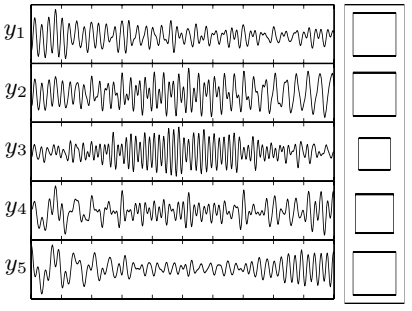


Fig. 2. The Linear mixtures $\boldsymbol{y}(t)$, with their PLF to the reference displayed as the area of opposite squares

comparable to stochastic gradient algorithms. Then, $\varrho_t\, e^{\mathrm{i}\Psi_t}$ will be evaluated in a time window and the update rule is

$$\Delta \boldsymbol{w}_t \propto \beta_t \, \frac{2\varrho_t \sin(\Psi_t - \Delta\phi(t))}{S^2(t)} \boldsymbol{\Gamma}(t)\, \boldsymbol{w}_t. \tag{5}$$

The evolution of the synchrony, or loss of it, for a component can be assessed by monitoring the quantity ϱ_t. The choice of forgetting factor β_t is then a critical element in the algorithm. A good choice will result in slowly varying component estimates.

If one suspects several components in the data to be synchronous with the reference signal, the algorithm can be applied several times, in a deflation manner. Each time a synchronous source $s(t)$ is found it needs to be removed from the data. The standard solution of projecting $s(t)$ back to the observation space and subtracting it from $\boldsymbol{y}(t)$, would require the data to be whitened. This can be achieved by an invertible linear transformation of $\boldsymbol{y}(t)$, prior to running Algo. 1. Since this presents just an additional linear mixing, Algo. 1 can, without any loss of generality, compensate for it. Let the whitened data be $\boldsymbol{z}(t)$, then each component can be subtracted, *e.g.* by $\boldsymbol{z}'(t) = \boldsymbol{z}(t) - (\boldsymbol{w}\boldsymbol{w}^{\mathsf{T}})^{-1}\boldsymbol{w}\, s(t)$, and the process continues. Even for two components with exactly the same phase evolution, *i.e.* identical PLFs, if their amplitudes vary, the algorithm would not converge to a mixture of those.

2.1 Simulation Examples

True blind source separation (BSS) algorithms use no explicit information about the sources to be extracted. The estimation relies typically on general assumptions such as statistical independence or non-Gaussianity of the sources' distributions. When in presence of oscillatory data, often a criterion based on temporal decorrelation can be employed (see [6] for an overview of various implementations of independent component analysis, one of the most used solutions to the

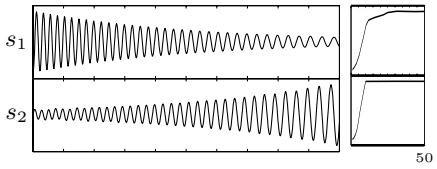


Fig. 3. Left: Examples of the two sources found by Algo.1. in the noiseless case ($\eta = 0.1$). Right: Corresponding objective function.

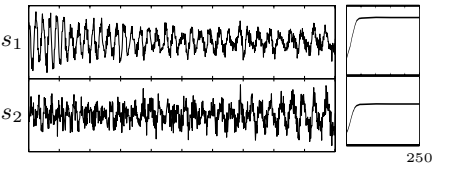


Fig. 4. Left: Examples of the two sources found by Algo.1. in the noisy case ($\eta = 0.1$). Right: Corresponding objective function.

BSS problem). No such requirements are necessary if knowledge of the source phase is available, up to an arbitrary constant phase lag.

To show the applicability of Algo. 1. to the search for components synchronous to a reference, we have generated a set of oscillatory signals $x_i(t) = A_i(t)\sin(\phi_i(t))$ (see Fig. 1). These can not be estimated from instantaneous linear mixtures by neither non-Gaussianity, nor temporal decorrelation criteria. This is because most of the sources have modulated amplitudes that insure histograms close to Gaussian. All have kurtosis close to that of $x_6(t)$, which corresponds to random Gaussian noise. Temporal decorrelation methods will fail also due to the varying frequency content of the sources.

The oscillators $x_i(t)|_{i=1,\cdots,6}$ are not phase coupled, thus the change of the instantaneous phase is proportional to their own natural frequency $\dot{\phi}_i = \omega_i(t)$. Only components 2 and 3 are coupled, such that $\phi_2(t) - \phi_3(t) = \text{const}$. Opposite to Fig. 1, is depicted the PLF of each source signal to the reference as the area of a square. The reference has the same phase dynamic as x_2 and x_3, but a different phase offset and an arbitrary amplitude, thus $\varrho_{x_2} = \varrho_{x_3} \approx 1$. This choice is just illustrative. Comparable results were reached using all other oscillators.

Figure 2 shows a set of linear mixtures of the signals in Fig. 1. Note that all mixtures have now a medium amount PLF to the reference, although clearly bellow those attained by the sources (no mixture has a PLF in excess of 0.75).

The perfect coupling between $x_2(t)$ and $x_3(t)$ suggests that any of the two can be found when the algorithm in Algo. 1 is used with a reference sharing their phase dynamics. Since any mixture of $x_2(t)$ and $x_3(t)$ is less synchronous to the reference, a single one is estimated at a time. The results depicted in Fig. 3 illustrate this fact. Note the correct estimation of the amplitude of the source signal, in addition to the phase recovery with the proper offset. In order to extract the second phase locked component, the first estimate should be removed by projecting it back to the observation space and subtracting it. This allows to extract the whole two dimensional subspace from the data, that is maximally phase locked to the reference.

As in many source extraction algorithms the global amplitude scale and the sign of the sources will remain undetermined. For that reason, the projection vector is arbitrarily normalised to unit norm in step 10 of the algorithm.

The convergence speed for a particular run of the method can be inspected on the right part of Fig. 3. The exact values can vary, depending on the choice

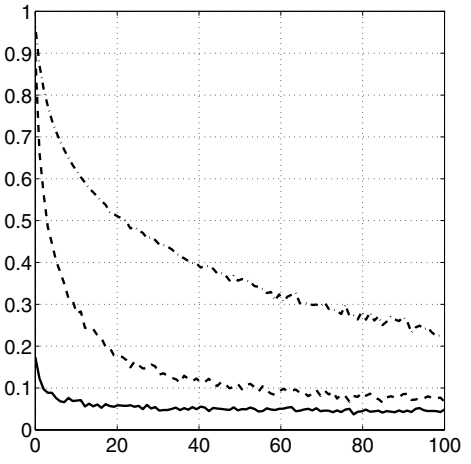


Fig. 5. *Ordinate:* PLF; *Abscissa:* σ^2. Maximal number of iterations and learning rate η are kept constant.

of η. If the phase of the reference signal is not present in the data, the algorithm will not reach a high PLF.

2.2 Sensitivity to Noise

The phase of a Gaussian white noise signal is typically mildly locked (PLF of ca. 0.1) to any other signal, including other Gaussian noise processes.

Let us assume that the observed mixtures $\boldsymbol{y}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \sigma\ \boldsymbol{\varepsilon}(t)$, as well as the reference signal $v(t) = u(t) + \varsigma\ \nu(t)$, are corrupted with noise of variance σ^2 and ς^2 respectively. $\boldsymbol{\varepsilon}(t)$ and $\nu(t)$ are both drawn form a Gaussian distribution having zero mean and $\mathsf{Cov}[\varepsilon_i(s)\varepsilon_j(t)] = \delta_{ij}\delta_{st}$, $\mathsf{Cov}[\nu(s)\nu(t)] = \delta_{st}$.

Figure 4 shows a replication of the experiment reported in Fig. 3, for the case of added observational noise of the same unit variance $\sigma^2 = 1$ as the data. The estimation is not as perfect as in the noiseless environment, possibly due to a non-zero phase locking between the reference signal and the noise (see Fig. 1). The PLF serves as a quality measure for the extracted components. The obtained PLFs are $\varrho_{s_1} = 0.83$ and $\varrho_{s_2} = 0.59$, which are significantly beneath those of the true sources. A common problem of deflation schemes is that the estimation error accumulates with the number of extracted components. Also the convergence speed is, as should be expected, reduced slightly with the noise source present.

In Figure 5 the PLF is plotted as a function of the observation noise magnitude σ^2. The different graphs correspond to values of ς^2 (the steeper slops for lower ς^2). The maximally achieved objective function value (keeping the maximum number of iterations fixed) deteriorates with increasing noise variance in both observations and reference.

The presence of noise has a profound influence in many real world applications. Furthermore, is it possible for real signals to exhibit very broad spectra, with

oscillatory dynamics in different frequency ranges. The Hilbert transform is not able to estimate a meaningful phase for such broad band signals. In conclusion, it is therefore advisable to remove, or reduce the noise and filter the signal in a targeted frequency band of interest, prior to the phase analysis. A way to combine filtering and phase estimation, that was reported to perform reliably on biological signals, is the convolution with complex Morlet wavelets [7]. Another valuable preprocessing approach is singular spectrum analysis (SSA), since it allows to decompose a signal into trends, oscillators and noise components, *cf.* Ref. [8].

3 Cortico-muscular Phase Locking in MEG Revisited

Strong coherence, *i.e.*, spectral cross-correlation (see [9] and references therein), and synchronisation have been observed between electrophysiological measurements from the brain and muscles (using electroencephalograms, EEG; magnetoencephalograms, MEG; and electromyograms, EMG). *Cortico-muscular* and *cortico-cortical* interactions were found in frequency bands centred around 15Hz, 20Hz and 40Hz. These have been supported by physiological consideration upon the biological processes involved.

An obstacle in these studies, *e.g.*, addressed in [10], is that the synchronisation among EEG or MEG channels is likely to result partially from cross-talk and volume conduction, *i.e.* the same oscillator being present in different measurements, because of a natural mixing process. Synchronisation between EEG/MEG and EMG, on the contrary, will be decreased as a result of the same process, since there is no single EEG/MEG channel that presents directly the underlying oscillator that is synchronous to the EMG.

In [11] the imaginary part of coherence has been introduced as a promising measure for brain interactions. It has the appealing property of not being sensitive to volume conduction, though it could possibly oversee interactions with very small phase lags. Such zero phase lag synchronisation could arise if the neuronal coupling between the two subsystems is strong and symmetric. On the other hand, the amount of phase lag between the compared signals does not affect the estimate of the PLF, on which Algo. 1. is based. Since it only measures synchrony between the *source* signals and the reference, the volume conduction and cross-talk should also be decreased. The algorithm's assumptions of a linear and instantaneous nature of the mixing process, are both substantiated from a theoretical viewpoint (see, *e.g.*, [12,13]).

We tested Algo. 1 on real measurements, using the data set described in [9]. It consists of simultaneous MEG recordings, with a 306-sensor Vectorview neuromagnetometer (Neuromag Ltd; 204 planar gradiometers and 102 magnetometers), together with left and right hand EMG's. The subject was instructed to simultaneously keep isometric contraction in left and right hand muscles, using a special squeezing device. Only the measurements of the planar gradiometers were analysed. The sampling rate is 600 Hz for a duration of 3 minutes.

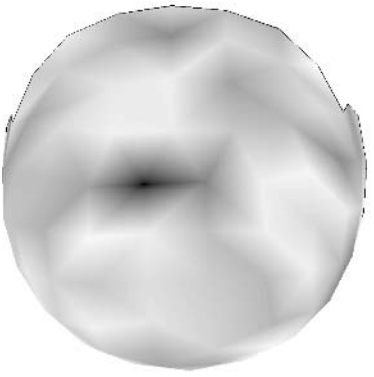


Fig. 6. Topographic map of the Neuromag MEG helmet at 18–20Hz

Based on physiological considerations, we have scanned a series of frequency ranges for targeting the algorithm. The results attained for the range 18–20Hz are shown in Fig. 6. This corresponds to the estimated projection between the extracted source and the measurements. This topographic map is conventionally called the component's field map. The view is taken from above, preserving right and left orientations, and with front facing up.

A comparison between the results shown, and the ones presented in [9], suggests the phase locked component to represent activity originating from the primary motor cortex. The variance of the extracted source is of the same magnitude as the measurements, indicating that the component has a significant presence in the recordings.

4 Concluding Remarks

Synchronisation plays a capital role in interacting oscillatory systems. It has been proposed in the literature that brain communication is implemented through synchronisation. We introduced a gradient based algorithm for the extraction of components, from measurements that are phase synchronous to a given reference signal. This can potentially elicit information about neuronal oscillator interactions from brain signals. The problem of noise was addressed in a controlled simulated environment. A preliminary study of its usage in cortico-muscular interactions was also presented.

In the future the robustness and convergence behaviour of the algorithm shall be determined in more detail. On a practical side, one should investigate which preprocessing techniques are useful when applying the algorithm to real world problems. Beyond the cortico-muscular example, we intend to investigate communication inside the central nervous system. This will require an extension of the algorithm in which the reference signal is estimated also from the measured signals in an unsupervised manner.

Acknowledgements

The authors would like to express their gratitude to Alexander Ilin and Jaakko Särelä for discussions on the topic of the article.

References

1. Pikovsky, A., Rosenblum, M., Kurths, J.: Synchronization – A universal concept in nonlinear sciences. Volume 12 of Cambridge Nonlinear Science Series. Cambridge University Press, UK (2001)
2. Kuramoto, Y.: Chemical Oscillations, Waves and Turbulences. Springer Berlin (1984)
3. Strogatz, S.H.: From Kuramoto to Crawford: Exploring the Onset of Synchronization in Populations of Coupled Oscillators. Physica D **143** (2000) 1–20
4. Hindmarsh, J.L., Rose, R.M.: A model of neuronal bursting using three coupled first–order differential equations. Proceedings of the Royal Society of London **221** (1984) 87–102
5. Frank, T.D., Daffertshofer, A., Pepper, C.E., Beek, P.J., Haken, H.: Towards a comprehensive theory of brain activity: Coupled oscillator systems under external forces. Physica D **144** (2000) 62–86
6. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons, Inc. (2001)
7. Lachaux, J.P., Rodriguez, E., Martinerie, J., Varela, F.J.: Measuring phase synchrony in brain signals. Human Brain Mapping **8**(4) (1999) 194–208
8. Ghil, M., Allen, M.R., Dettinger, M.D., Ide, K., Kondrashov, D., Mann, M.E., Robertson, A., Saunders, A., Tian, Y., Yiou, P.: Advanced Spectral Methods for Climatic Time Series. Reviews of Geophysics **40**(1) (2001)
9. Vigário, R., Jensen, O.: Identifying Cortical Sources of Corticomuscle Coherence During Bimanual Muscle Contraction by Temporal Decorrelation. In: Proceedings of IEEE International Symposium on Signal Processing and Its Applications. (2003)
10. Meinecke, F.C., Ziehe, A., Kurths, J., Müller, K.R.: Measuring Phase Synchronization of Superimposed Signals. Physical Review Letters **94**(8) (2005)
11. Nolte, G., Bai, O., Wheaton, L., Mari, Z., Vorbach, S., Hallett, M.: Identifying true brain interaction from EEG data using the imaginary part of coherency. Clinical Neurphysiology **115** (2004) 2292–2307
12. Hämäläinen, M., Hari, R., Ilmoniemi, R., Knuutila, J., Lounasmaa, O.V.: Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. Reviews of Modern Physics **65**(2) (1993) 413–497
13. Vigário, R., Särelä, J., Jousmäki, V., Hämäläinen, M., Oja, E.: Independent component approach to the analysis of EEG and MEG recordings. IEEE Transactions on Biomedical Engineering **47**(5) (2000) 589–593

A Gradient of ϱ^2

The gradient can be written out as $\nabla\varrho^2 = \nabla\big(\varrho\cos(\Psi)\big)^2 + \nabla\big(\varrho\sin(\Psi)\big)^2$, which equals

$$2\varrho\big(\cos(\Psi)[\nabla\varrho\cos(\Psi)] + \sin(\Psi)[\nabla\varrho\sin(\Psi)]\big). \tag{6}$$

From the definition in Eq. (2) it follows that $\varrho\cos(\Psi) = \frac{1}{T}\sum_{t=1}^{T}\cos(\Delta\phi(t))$. Inserting this and the equivalent identity for $\varrho\sin(\Psi)$ into Eq. (6) and further evaluating the gradient yields

$$\frac{2\varrho}{T}\sum_{t=1}^{T}\Big[\sin(\Psi)\cos\big(\Delta\phi(t)\big) - \cos(\Psi)\sin\big(\Delta\phi(t)\big)\Big]\nabla\phi(t). \tag{7}$$

The phase $\phi(t)$ is the angle of $\hat{s}(t)$ in the complex plane. This is given as $\phi(t) = \operatorname{angle}\hat{s}(t) = \operatorname{arctan2}(\mathcal{H}[s](t), s(t))$, where the two arguments arctan maps the angle into the correct quadrant. Let the Hilbert transform $\mathcal{H}[\cdot]$ operate on the coordinates of a vector. For the gradient of $\phi(t) = \operatorname{arctan2}(\boldsymbol{w}^\mathsf{T}\boldsymbol{y}(t), \boldsymbol{w}^\mathsf{T}\mathcal{H}[\boldsymbol{y}](t))$, the arctan2-function can be substituted by the normal arctan, so that

$$\nabla\phi(t) = \nabla\arctan\Big(\frac{\boldsymbol{w}^\mathsf{T}\mathcal{H}[\boldsymbol{y}](t)}{\boldsymbol{w}^\mathsf{T}\boldsymbol{y}(t)}\Big) = \frac{(\boldsymbol{w}^\mathsf{T}\boldsymbol{y}(t))\mathcal{H}[\boldsymbol{y}](t) - (\boldsymbol{w}^\mathsf{T}\mathcal{H}[\boldsymbol{y}](t))\boldsymbol{y}(t)}{\Big(1+\big(\frac{\boldsymbol{w}^\mathsf{T}\mathcal{H}[\boldsymbol{y}](t)}{\boldsymbol{w}^\mathsf{T}\boldsymbol{y}(t)}\big)^2\Big)(\boldsymbol{w}^\mathsf{T}\boldsymbol{y}(t))^2}.$$

The first factor in the denominator is the derivative of arctan and the second is the result of the quotient rule of differentiation. This can be rearranged to

$$\nabla\phi(t) = \frac{(\mathcal{H}[\boldsymbol{y}](t)\boldsymbol{y}^\mathsf{T}(t) - \boldsymbol{y}(t)\mathcal{H}[\boldsymbol{y}]^\mathsf{T}(t))\,\boldsymbol{w}}{(\boldsymbol{w}^\mathsf{T}\boldsymbol{y}(t))^2 + (\boldsymbol{w}^\mathsf{T}\mathcal{H}[\boldsymbol{y}](t))^2}, \tag{8}$$

reviling the denominator to be the square magnitude of extracted source $s^2(t) + \mathcal{H}[s]^2(t) = S^2(t)$. The matrix $\boldsymbol{\Gamma}(t) = (\mathcal{H}[\boldsymbol{y}](t)\boldsymbol{y}^\mathsf{T}(t) - \boldsymbol{y}(t)\mathcal{H}[\boldsymbol{y}]^\mathsf{T}(t))$ in the numerator can also be written in terms of the phase of the observation signal as

$$\begin{aligned}[\boldsymbol{\Gamma}(t)]_{ij} &= Y_i(t)Y_j(t)\sin(\varphi_i(t))\cos(\varphi_j(t)) - Y_i(t)Y_j(t)\cos(\varphi_i(t))\sin(\varphi_j(t)) \\ &= Y_i(t)Y_j(t)\sin(\varphi_i(t) - \varphi_j(t)).\end{aligned}$$

The same simplification can be applied to Eq. (7), to finally arrive at

$$\nabla\varrho^2 = \frac{2\varrho}{T}\sum_{t=1}^{T}\frac{\sin(\Psi - \Delta\phi(t))}{S^2(t)}\boldsymbol{\Gamma}(t)\,\boldsymbol{w}. \tag{9}$$

Analytic Solution of Hierarchical Variational Bayes in Linear Inverse Problem

Shinichi Nakajima[1] and Sumio Watanabe[2]

[1] Nikon Corporation, 201-9 Miizugahara, Kumagaya, 360-8559 Japan
nakajima.s@nikon.co.jp, swatanab@pi.titech.ac.jp
http://watanabe-www.pi.titech.ac.jp/~nkj23/index.html
[2] Tokyo Institute of Technology, Mailbox R2-5, 4259 Nagatsuda, Yokohama, 226-8503 Japan

Abstract. In singular models, the Bayes estimation, commonly, has the advantage of the generalization performance over the maximum likelihood estimation, however, its accurate approximation using Markov chain Monte Carlo methods requires huge computational costs. The variational Bayes (VB) approach, a tractable alternative, has recently shown good performance in the automatic relevance determination model (ARD), a kind of hierarchical Bayesian learning, in brain current estimation from magnetoencephalography (MEG) data, an ill-posed linear inverse problem. On the other hand, it has been proved that, in three-layer linear neural networks (LNNs), the VB approach is asymptotically equivalent to a positive-part James-Stein type shrinkage estimation. In this paper, noting the similarity between the ARD in a linear problem and an LNN, we analyze a simplified version of the VB approach in the ARD. We discuss its relation to the shrinkage estimation and how ill-posedness affects learning. We also propose the algorithm that requires simpler computation than, and will provide similar performance to, the VB approach.

1 Introduction

It is known that the Bayes estimation provides better generalization performance than the maximum likelihood (ML) estimation when we use a model having singularities, on which the Fisher information matrix is singular, in the parameter space. However, Markov chain Monte Carlo (MCMC) methods, used for approximation of the Bayes posterior distribution, require huge computational costs. The variational Bayes (VB) approach was proposed as a tractable alternative [1, 2], and is often applied to singular models, for example, mixture models and hidden Markov models. Recently, the VB approach has been applied also to the automatic relevance determination model (ARD) [3] in a linear inverse problem, i.e., brain current estimation from magnetoencephalography (MEG) data [4]. Although the advantage of the VB approach has been experimentally shown in many applications, its generalization performance had been theoretically clarified in no singular model until quite recently. Last year, proving the asymptotic equivalence between the VB approach and a positive-part James-Stein (JS) type shrinkage estimation [5], we have clarified the generalization error of the VB approach in three-layer linear neural networks (LNNs), the simplest singular models [6].

In this paper, noting the similarity between the ARD in a linear problem and an LNN, we clarify some properties of the VB approach in the ARD and then propose

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 240–249, 2006.

the alternative that requires less computational costs. In Section 2, we shortly describe the framework of the VB approach. In Section 3, we explain the brain current estimation and the ARD, and then, discuss the similarity between the ARD and an LNN. In Section 4, we, in detail, describe the setting assumed in our theoretical analysis. After that, in Section 5, we analyze the VB approach in the ARD, and show its relation to the JS type shrinkage estimation. Discussion including our proposal is in Section 6, and finally, conclusions and future work are in Section 7.

2 Variational Bayes Approach

Let $Y^n = \{y_1, \ldots, y_n\}$ be arbitrary n training samples independently and identically taken from the true distribution. In the framework of the Bayes estimation, the posterior distribution of the parameter w of a model $p(y|w)$ is given by

$$p(w|Y^n) = \frac{\phi(w)\prod_{i=1}^n p(y_i|w)}{Z(Y^n)}, \quad \text{where} \quad Z(Y^n) = \int \phi(w)\prod_{i=1}^n p(y_i|w)dw \quad (1)$$

is the marginal likelihood, and $\phi(w)$ is the prior distribution. The predictive distribution is defined as the average of the model over the posterior distribution.

In the variational Bayes (VB) approach [2], called the mean field approximation in statistical physics, we first define the following functional, called the generalized free energy in this paper, of an arbitrary trial posterior distribution $r(w|Y^n)$:[1]

$$\bar{F}(r) = -S(r) + nE(r), \qquad \text{where} \quad (2)$$
$$S(r) = -\left\langle \log r(w) \right\rangle_{r(w)} \quad \text{and} \quad E(r) = -n^{-1}\left\langle \log\left(\phi(w)\prod_{i=1}^n p(y_i|w)\right)\right\rangle_{r(w)}$$

are the entropy and the energy, respectively. Here, $\langle\cdot\rangle_p$ denotes the expectation value over a distribution p. Note that $n^{-1}\bar{F}(r)$ corresponds to the Helmholtz free energy if we consider n to be the inverse temperature, hence, the Bayes posterior distribution, Eq.(1), which minimizes the Helmholtz free energy, corresponds to the equilibrium distribution [7]. Then, in the VB approach, restricting the space of allowed $r(w)$, we minimize the generalized free energy, Eq.(2). The optimum of $r(w)$ is called the VB posterior distribution, over which the expectation value of w is called the VB estimator.

3 Model

3.1 MEG Inverse Problem

Magnetoencephalography (MEG) is one of the major recording means of brain activity, in which we estimate the electric current distribution in a brain from the magnetic fields that are induced by the current and observed on the head [8]. For simplicity, we assume that the current and the field are scalar. Let $a' \in \mathbb{R}^M$ be the current vector of which each element corresponds to the current value at each site in a brain, and $y \in \mathbb{R}^N$ the field

[1] We will hereafter abbreviate $r(w|Y^n)$ by $r(w)$ or by r.

vector of which each element corresponds to the field value at each site on the head. We utilize the following linear regression model:

$$y = Va' + \varepsilon, \tag{3}$$

where V is the $N \times M$ constant matrix, called the lead field matrix, that represents the field induced by the current, and $\varepsilon \in \mathbb{R}^N$ is an observation noise [8,4]. By $\mathcal{N}_d(\mu, \Sigma)$ we denote the d-dimensional normal distribution with average μ and covariance matrix Σ, and by $\mathcal{N}_d(\cdot; \mu, \Sigma)$ its probability density. Assume that the noise, ε in Eq.(3), is subject to $\mathcal{N}_N(0, \sigma_y^2 I_N)$, where $0 < \sigma_y^2 < \infty$ and I_d is the $d \times d$ identity matrix. Then, the probability density of the field is given by

$$p(y|a') = \mathcal{N}_N(y; Va', \sigma_y^2 I_N). \tag{4}$$

In typical MEG estimation problems, the number of sites at which the fields are observed is smaller than the number of sites at which we want to know the brain currents, i.e., $N < M$, hence, the MEG estimation is an ill-posed problem. Therefore, in the a' space, the region in which any point gives the maximum likelihood is not a point, given an observed field. So, a prior assumption is needed to select one point in that region. One of the most popular methods is the minimum norm method, in which the point giving the minimum norm is selected from the points giving the maximum likelihood [8]. We can easily find that the maximum a posterior (MAP) estimation with the following prior distribution provides the minimum norm solution as well: $\phi(a') = \mathcal{N}_M(a'; 0, I_M)$.

3.2 Automatic Relevance Determination

The automatic relevance determination model (ARD), a kind of hierarchical Bayesian learning, was proposed to eliminate irrelevant connections from neural networks [3]. In the ARD, we first introduce a prior distribution of the parameters, i.e., the weight vectors, with the hyperparameters corresponding to the variances. Then, we introduce a prior distribution of the hyperparameters, called a hyperprior. If we apply the Bayes estimation to this model, many weight vectors tend to be eliminated as irrelevant connections, because of the singularities caused by the hierarchy. (See [9] for detail.)

Now, we focus on its application to the MEG inverse problem, whose distribution is given by Eq.(4). We use the following prior distribution of a':

$$\phi(a'\|B^{-2}) = \mathcal{N}_M(a'; 0, B^2), \tag{5}$$

where B^{-2} is the hyperparameter. We consider in this paper the simplest ARD, where B^{-2} is diagonal.[2] Then, increasing the (m,m)-th element of B^{-2} eliminates the m-th element, a'_m, of the current as an irrelevant one. We can estimate the value of B^{-2} based on the empirical Bayes (EB) approach [10], where the hyperparameter is estimated by maximizing the marginal likelihood, $Z(Y^n)$ in Eq.(1), or can estimate the posterior distribution of B^{-2} by introducing the hyperprior, such as

$$\phi(B^{-2}) = \prod_{m=1}^{M} \Gamma\left(B_{mm}^{-2}; \bar{\kappa}_{0m}, \bar{\nu}_{0m}\right), \tag{6}$$

[2] The differences between the setting assumed in this paper and that in [4] is summarized at the end of Section 4.2.

where we denote by $\Gamma(\kappa, \nu)$ the Gamma distribution with shape parameter κ and scale parameter ν, and by $\Gamma(\cdot; \kappa, \nu)$ its probability density. To the latter method, we can apply the VB approach and obtain the iterative algorithm, restricting the posterior distribution such that a' and B^{-2} are independent of each other [4].

The following point is important: MEG data are time series, and we want to know the current at each point of time; on the other hand, the hyperparameter B^{-2} is considered to be invariant during some time period in [4], which essentially affects learning and enhances elimination of irrelevant elements, as will be shown in the following sections.

3.3 Similarity to Linear Neural Networks

Let $x' \in \mathbb{R}^M$ be the formal input vector of which all the elements are equal to one. Then, the transform $a' \to Ba$, where $a \in \mathbb{R}^M$, makes the model distribution, Eq.(4), and the prior distribution, Eq.(5), as

$$p(y|x', A, B) = \mathcal{N}_N(y; VBAx', \sigma_y^2 I_N), \tag{7}$$

$$\phi(a) = \mathcal{N}_M(a; 0, I_M), \tag{8}$$

where A is the $M \times M$ diagonal matrix whose (m, m)-th element is equal to the m-th element of a. We thus find that the model, Eq.(7), is similar to a linear neural network model (LNN),[3] in which the VB approach has been analyzed in [6]. However, there is an important difference, i.e., the existence of the lead field matrix, V in Eq.(7), although the ARD is equivalent to an LNN when V is general diagonal. We do not like to transform the basis of the current vector, a', space, so that V is general diagonal, since the purpose of that application is to find the few sites where synapses fire.

4 Setting

4.1 Restriction on Posterior Distribution

As discussed in Section 3.3, the ARD in the linear inverse problem, Eq.(4), with the prior distribution, Eq.(5), is equivalent to the model, Eq.(7), with the prior distribution, Eq.(8). By b we denote the M-dimensional vector whose m-th element is equal to the (m, m)-th element of B. We introduce the following prior distribution of b, which is substituted for the hyperprior of B^{-2} in the ARD:

$$\phi(b) = \mathcal{N}_M(b; 0, c_b^2 I_M), \tag{9}$$

where $0 < c_b^2 < \infty$ is a *constant* hyperparameter. Note that B_{mm}^2 is then subject to $\Gamma(c_b^2/2, 2)$. For symmetry, we also introduce a *constant* hyperparameter $0 < c_a^2 < \infty$ in the prior distribution of a as follows:

$$\phi(a) = \mathcal{N}_M(a; 0, c_a^2 I_M). \tag{10}$$

Actually, it will be shown in the following sections that the values of the *constant* hyperparameters, c_a^2 and c_b^2, do not asymptotically affect learning as far as they are positive

[3] The definition of the LNN is described in Appendix A.

and finite; while whether the hyperparameter B^{-2} is constant or estimated from observation strongly affects learning even in the asymptotic limit.

Now we apply the VB approach to the transformed ARD model, Eq.(7), with the prior distributions, Eqs.(10) and (9). We restrict the trial posterior distribution such that a and b are independent of each other:

$$r(a,b) = r(a)r(b). \tag{11}$$

Then, the generalized free energy, Eq.(2), can be written as follows:

$$\bar{F}(Y^n) = \int r(a)r(b) \log \frac{r(a)r(b)}{p(Y^n|a,b)\phi(a)\phi(b)} dadb. \tag{12}$$

Using the variational method [2], we obtain the following condition:

$$r(a) \propto \phi(a) \exp\langle \log p(Y^n|a,b) \rangle_{r(b)}, \quad r(b) \propto \phi(b) \exp\langle \log p(Y^n|a,b) \rangle_{r(a)}. \tag{13}$$

We find from Eq.(13) that the VB posterior distribution is the normal, because the log-likelihood, $\log p(Y^n|a,b)$, is a biquadratic function of a and b, and we use the normal prior distributions, Eqs.(10) and (9). In this paper, we furthermore restrict $r(a,b)$ such that all the elements are independent of each other for simplicity, which results that

$$r(a_m) \propto \phi(a_m) \exp\langle \log p(Y^n|a,b) \rangle_{r(a)r(b)/r(a_m)}, \tag{14}$$

$$r(b_m) \propto \phi(b_m) \exp\langle \log p(Y^n|a,b) \rangle_{r(a)r(b)/r(b_m)}. \tag{15}$$

4.2 Summary of Setting

We summarize our setting in the following. Let $A^{(u)}$ be an $M \times M$ diagonal parameter matrix at the time u, B another $M \times M$ diagonal parameter matrix, which is assumed to be invariant during the time period $u = 1, \ldots, U$, and $y^{(u)}$ an N-dimensional observed vector. By $a^{(u)}$ we denote the M-dimensional parameter vector representing the diagonal elements of $A^{(u)}$, i.e., $a_m^{(u)} = A_{mm}^{(u)}$, and by b the M-dimensional parameter vector representing the diagonal elements of B, i.e., $b_m = B_{mm}$. Suppose that we have n training samples, i.e., n sets of U time series data, denoted by Y^n.

In this paper, restricting the trial posterior distribution $r(a,b)$ such that all the elements are independent of each other, we analyze the VB approach in the model

$$p(\{y^{(u)}\}|\{a^{(u)}\},b) = \prod_{u=1}^{U} \mathcal{N}_N(y^{(u)}; V \sum_{m=1}^{M} b_m a_m^{(u)} 1_m, \sigma_y^2 I_N) \tag{16}$$

with the prior distributions

$$\phi(\{a^{(u)}\}) = \prod_{u=1}^{U} \mathcal{N}_M(a^{(u)}; 0, c_a^2 I_M), \qquad \phi(b) = \mathcal{N}_M(b; 0, c_b^2 I_M), \tag{17}$$

where $V = (v_1, \ldots, v_M)$ is an $N \times M$ constant matrix, and 1_m denotes the M-dimensional vector whose m-th element is equal to unity and all the other elements are equal to zero. The noise variance, σ_y^2, is assumed to be known.

Finally, we summarize the major differences of our setting from that in [4]:

1. The spatial correlation of the brain current distribution is considered by introducing the smoothness prior, where the hyperparameter B^{-2} in Eq.(5) is not assumed to be diagonal, in [4]; while B is assumed to be diagonal in this paper.
2. The restriction on the VB posterior distribution is only the independence between a' and B^{-2} in [4]; while the independence among the elements of a, as well as those of b, is also assumed in this paper.
3. The prior distribution of b_m^{-2} is $\Gamma\left(b_m^{-2}; \bar{\kappa}_{0m}, \bar{\nu}_{0m}\right)$ in [4]; while that of its inverse, b_m^2, is $\Gamma(b_m^2; c_b^2/2, 2)$ in this paper.
4. The number of samples for estimation of each site and each point of time is only one, i.e., $n = 1$, in [4]; while we consider the case that we have sufficiently large n training samples in this paper. However, we will derive also the non-asymptotic solution in the case that $U = 1$, at the end of Section 5.2.

5 Theoretical Analysis

5.1 Variational Condition

Define the following M-dimensional vector:

$$j^{(u)}(Y^n) = n^{-1}\sum_{i=1}^n V^t y_i^{(u)}, \text{i.e.,} \qquad j_m^{(u)}(Y^n) = n^{-1}\sum_{i=1}^n v_m^t y_i^{(u)}, \tag{18}$$

where t denotes the transpose of a matrix or vector. We hereafter abbreviate $j^{(u)}(Y^n)$ as $j^{(u)}$. We find from Eqs.(14) and (15) that the VB posterior distribution factorizes as

$$r(\{a^{(u)}\}, b) = \prod_{m=1}^M \left\{ \left(\prod_{u=1}^U \mathcal{N}_1(a_m^{(u)}; \mu_{a_m}^{(u)}, \sigma_{a_m}^{2(u)} I_M) \right) \mathcal{N}_1(b_m; \mu_{b_m}, \sigma_{b_m}^2 I_M) \right\}, \tag{19}$$

where $\mu_{a_m}^{(u)}$, μ_{b_m}, $\sigma_{a_m}^{2(u)}$, and $\sigma_{b_m}^2$ are scalar for $m = 1, \ldots, M$ and $u = 1, \ldots, U$. Note that the VB estimator of the m-th element of the current at the time u is given by $(\hat{a}_m'^{(u)})_{\mathrm{VB}} = (\hat{b}_m \hat{a}_m^{(u)})_{\mathrm{VB}} = \mu_{b_m}\mu_{a_m}^{(u)}$. By $\tilde{\ }$ we denote the U-dimensional time series vector, for example, $\tilde{a}_m = (a_m^{(1)}, \ldots, a_m^{(U)})^t$. Then, we obtain the following variational condition by substituting Eq.(19) into Eqs.(14) and (15):

$$\tilde{\mu}_{a_m} = n\sigma_y^{-2}\|v_m\|^2 \sigma_{a_m}^2 \tilde{z}_m \mu_{b_m}, \tag{20}$$

$$\sigma_{a_m}^{2(u)} = n^{-1}\sigma_y^2 \left(\|v_m\|^2 (\mu_{b_m}^2 + \sigma_{b_m}^2) + n^{-1}\sigma_y^2 c_a^{-2} \right)^{-1}, \tag{21}$$

$$\mu_{b_m} = n\sigma_y^{-2}\|v_m\|^2 \sigma_{b_m}^2 \tilde{z}_m^t \tilde{\mu}_{a_m}, \tag{22}$$

$$\sigma_{b_m}^2 = n^{-1}\sigma_y^2 \left(\|v_m\|^2 (\|\tilde{\mu}_{a_m}\|^2 + U\sigma_{a_m}^2) + n^{-1}\sigma_y^2 c_b^{-2} \right)^{-1}, \tag{23}$$

where

$$\tilde{z}_m = \|v_m\|^{-2}(\tilde{j}_m - \textstyle\sum_{m' \neq m} \mu_{b_{m'}} \tilde{\mu}_{a_{m'}} v_m^t v_{m'}). \tag{24}$$

Here, we denote $\sigma_{a_m}^{2(u)}$ by $\sigma_{a_m}^2$ in Eqs.(20) and (23), since Eq.(21) implies that it is invariant for u. Similarly, substituting Eq.(19) into Eq.(12), we also have the following form of the generalized free energy:

$$2\bar{F}(Y^n) = \sum_{m=1}^M \Big\{ -\log \sigma_{a_m}^{2U} \sigma_{b_m}^2 - 2n\sigma_y^{-2}\|v_m\|^2 \left(\tilde{z}_m^t \tilde{\mu}_{a_m} \mu_{b_m} \right) + n\sigma_y^{-2}\|v_m\|^2 (\|\tilde{\mu}_{a_m}\|^2 + U\sigma_{a_m}^2)(\mu_{b_m}^2 + \sigma_{b_m}^2) \Big\} + \text{const.}. \tag{25}$$

5.2 Variational Bayes Solution

The variational condition, Eqs.(20)–(23), can be analytically solved, which leads to the following theorem:

Theorem 1. *The VB estimator of the m-th element of the current is given by*

$$(\hat{b}_m\hat{\tilde{a}}_m)_{VB} = \begin{cases} 0 & \text{for } m \text{ such that } v_m = 0 \\ \mathcal{S}(\tilde{z}_m; \sigma_y^2 U/\|v_m\|^2) + O_p(n^{-1}) & \text{for } m \text{ such that } v_m \neq 0 \end{cases}, \quad (26)$$

$$\text{where} \quad \mathcal{S}(z;\chi) = \theta(n\|z\|^2 > \chi)\left(1 - \chi/n\|z\|^2\right) z \quad (27)$$

is the positive-part James-Stein (JS) type shrinkage operator with the degree of shrinkage $\chi > 0$.[4] *Here $\theta(\cdot)$ denotes the indicator function of an event.*

(Outline of the proof) We will find the solution of the variational condition, Eqs.(20)–(23). We easily have the solution for the elements such that $v_m = 0$. For the other elements such that $v_m \neq 0$, we have the following variances by solving Eqs.(21) and (23):

$$\hat{\sigma}^2_{a_m} = \frac{-(\hat{\eta}_m^2 - n^{-1}\sigma_y^2\|v_m\|^2(U-1)) + \sqrt{(\hat{\eta}_m^2 + n^{-1}\sigma_y^2\|v_m\|^2(U+1))^2 - 4n^{-1}\sigma_y^4 U\|v_m\|^4}}{2U\|v_m\|^2(\|v_m\|^2\hat{\mu}^2_{b_m} + n^{-1}\sigma_y^2 c_a^{-2})}, \quad (28)$$

$$\hat{\sigma}^2_{b_m} = \frac{-(\hat{\eta}_m^2 + n^{-1}\sigma_y^2\|v_m\|^2(U-1)) + \sqrt{(\hat{\eta}_m^2 + n^{-1}\sigma_y^2\|v_m\|^2(U+1))^2 - 4n^{-1}\sigma_y^4 U\|v_m\|^4}}{2\|v_m\|^2(\|v_m\|^2\|\hat{\tilde{\mu}}_{a_m}\|^2 + n^{-1}\sigma_y^2 c_b^{-2})}, \quad (29)$$

$$\text{where} \quad \hat{\eta}_m^2 = (\|v_m\|^2\|\hat{\tilde{\mu}}_{a_m}\|^2 + n^{-1}\sigma_y^2 c_b^{-2})(\|v_m\|^2\hat{\mu}^2_{b_m} + n^{-1}\sigma_y^2 c_a^{-2}). \quad (30)$$

By using Eqs.(20), (22), (28), and (29), we have

$$\hat{\eta}_m^2 = \left(1 - \frac{\sigma_y^2}{n\|v_m\|^2\|\tilde{z}_m\|^2}\right)\left(1 - \frac{\sigma_y^2 U}{n\|v_m\|^2\|\tilde{z}_m\|^2}\right)\|\tilde{z}_m\|^2, \quad (31)$$

$$\sigma_y^2(U c_a^{-2}\hat{\delta}_m - c_b^{-2}\hat{\delta}_m^{-1}) = n(U-1)(\|\tilde{z}_m\| - \|v_m\|^2\hat{\gamma}_m), \quad (32)$$

$$\text{where} \quad \hat{\gamma}_m = \|\hat{\tilde{\mu}}_{a_m}\|\hat{\mu}_{b_m} \quad \text{and} \quad \hat{\delta}_m = \|\hat{\tilde{\mu}}_{a_m}\|/\hat{\mu}_{b_m}. \quad (33)$$

Solving Eqs.(30)–(32), we obtain the VB estimator in Theorem 1. (Q.E.D.)

Moreover, we obtain the following non-asymptotic expression of the VB estimator when $U = 1$:

Theorem 2. *The VB estimator of the m-th element of the current when $U = 1$ and $v_m \neq 0$ is given by*

$$(\hat{b}_m\hat{a}_m)_{VB} = \text{sign}(z_m) \cdot \max\left(0, \|\mathcal{S}(z_m; \sigma_y^2/\|v_m\|^2)\| - \sigma_y^2(n c_a c_b\|v_m\|^2)^{-1}\right), \quad (34)$$

where sign$(\cdot)$ *denotes the sign of a scalar.*

(Outline of the proof) We find from Eq.(32) that $\|\hat{\delta}_m\| = c_a/c_b$, which makes Eqs.(30) and (31) rigorously solvable and leads to Theorem 2. (Q.E.D.)

Note that neither Theorem 1 nor Theorem 2 provides any explicit expression of the VB estimator, since $\tilde{z}_m$, given by Eq.(24), depends on the other elements of the VB estimator, i.e., $(\hat{b}_{m'}\hat{\tilde{a}}_{m'})_{\text{VB}}$ for $m' \neq m$. So, further consideration is needed.

[4] The positive-part JS type shrinkage estimator, as well as operator, is explained in Appendix B.

5.3 Comparison with Shrinkage Estimation

By $(\cdot)^-$ we denote the Moore-Penrose generalized inverse of a matrix. Consider the following positive-part JS type shrinkage estimator based on the minimum norm maximum likelihood (MNML) estimator:

$$(\hat{b}_m\hat{\tilde{a}}_m)_{\rm PJS} = \mathcal{S}\left((\hat{b}_m\hat{\tilde{a}}_m)_{\rm MN}; \sigma_y^2 U/\|v_m\|^2\right), \text{ where } (\hat{B}\hat{a}^{(u)})_{\rm MN} = (V^tV)^- j^{(u)} \quad (35)$$

is the MNML estimator. Hereafter, we compare the VB estimator, Eq.(26), and the shrinkage estimator, Eq.(35). From the definition of $\tilde{z}_m$, given by Eq.(24), we find that $\tilde{z}_m$ is the *unique* ML estimator and hence the VB and the shrinkage estimators of the m-th element are asymptotically equivalent to each other, if $v_m^t v_{m'} = 0$ for $^\forall m' \neq m$. However, nonorthogonality and linear dependence, which causes ill-posedness, of the set of the lead field column vectors, i.e., $\{v_m\}$, makes a difference between them.

Consider the simplest ill-posed case where all the lead field vectors, $\{v_m\}$ for $m = 1, \ldots, M$, are parallel to each other. Then, the MNML estimator at u is given by

$$(\hat{B}\hat{a}^{(u)})_{\rm MN} = (\textstyle\sum_{m=1}^M \|v_m\|^2)^{-1} j^{(u)}, \quad (36)$$

from which we find that all the elements of the MNML estimator, naturally, have the same sign. Hence, we find from Eq.(24) that the fact that $\|(\hat{b}_m\hat{a}_m^{(u)})_{\rm VB}\| < \|(\hat{b}_m\hat{a}_m^{(u)})_{\rm MN}\|$ leads to the fact that $\|(\hat{b}_m\hat{a}_m^{(u)})_{\rm MN}\| < \|z_m^{(u)}\|$. Consequently, we conclude that, in this case, the amplitude of the positive-part JS type shrinkage estimator gives the lower bound of the amplitude of the VB estimator, i.e.,

$$\|(\hat{b}_m\hat{a}_m^{(u)})_{\rm PJS}\| < \|(\hat{b}_m\hat{a}_m^{(u)})_{\rm VB}\|, \quad (37)$$

because $\|\mathcal{S}(z;\chi)\|$ is an increasing function of $\|z\|$. However, if there is any pair of v_m and $v_{m'}$ that are neither orthogonal nor parallel to each other, neither the asymptotic equivalence between the VB solution and the shrinkage estimator nor Inequality (37) necessarily hold. Further consideration is future work.

6 Discussion

6.1 Features

The authors of [4] compared their approach with a previous work, the MAP estimation or the Wiener filter method with inaccurate prior information, where the hyperparameter, B^{-2} in Eq.(5), is regarded as a constant. Consider the situation, with which all the model selection and the regularization methods have been proposed to cope, when we do not accurately know the true prior and may use a model with irrelevant elements or redundant degree of freedom. Because the MAP estimation causes no singularity, it provides the generalization performance asymptotically equivalent to that of the regular models. On the other hand, because an LNN is singular, it provides different generalization performance even in the asymptotic limit [6]. The case in this paper corresponds to the case of a single-output (SO) LNN, i.e., an LNN with one output unit and one

hidden unit, with U input units. (See Appendix A.) Because it has been shown that the VB approach asymptotically dominates the ML, as well as the MAP, estimation in SOLNNs when U is sufficiently large [9],[5] we expect that the ARD will provide better performance than the MAP estimation. Moreover, in SOLNNs, the suppression of overfitting caused by the singularities is stronger in the VB approach than in the Bayes estimation, which means that the elimination of irrelevant elements is enhanced in the VB approach. In addition, note that the time period U significantly affects performance because the degree of shrinkage, χ, is proportional to U, as we find from Eq.(26).

6.2 Proposition

We propose to simply use the positive-part JS type shrinkage estimator, Eq.(35), based on the MNML estimator. It only requires the calculation of the Moore-Penrose generalized inverse like the MAP estimation; while it is expected to eliminate irrelevant elements to suppress overfitting like the VB approach, which has been shown to provide better performance than the MAP estimation [4] and requires relatively costly iterative calculation. If the noise variance, σ_y^2 in Eq.(35), is unknown, its ML estimator should be substituted for it.

Note that the shrinkage estimation, as well as the VB approach, is not coordinate-invariant unlike the ML estimator, and there is a difference between the shrinkage estimation and the VB solution in nonorthogonal cases, as shown in Section 5.3. Although Inequality (37) states that, in a special case, ill-posedness makes the elimination effect of the shrinkage estimation stronger than that of the VB approach, the discussion in Section 5.3 also seems to imply that the VB approach can be less affected by the nonorthogonality, and hence more desirable than the shrinkage estimation. Further analysis is future work.

7 Conclusions and Future Work

In this paper, noting the similarity between the automatic relevance determination model (ARD) in a linear problem and a linear neural network model, we have found the relation between the variational Bayes (VB) approach in the ARD and a positive-part James-Stein (JS) type shrinkage estimation. It has let us propose to use the shrinkage estimation as an alternative, which requires less costs and behaves like the VB approach.

The relation between the empirical Bayes (EB) approach in a linear model and the JS estimation was previously discussed in [10], where the JS estimator was derived as an EB estimator. We have recently pointed out the equivalence between the EB approach in a linear model and a subspace Bayes (SB) approach in a single-output LNN [9], and found the asymptotic equivalence between the VB and the SB approaches in LNNs [6]. The previous works above and this paper have revealed the similarity between the VB approach and the shrinkage estimation. But in this paper, it has also been found that the nonorthogonality of the basis makes a difference, on which we will focus from now. Consideration of what our simplification, i.e., the differences in setting between in [4] and in this paper, itemized at the end of Section 4.2, causes is also future work.

[5] It was conjectured that, in SOLNNs, the VB approach asymptotically dominates the ML estimation when $U \geq 5$ [9].

Acknowledgments

The authors would like to thank Dr. Okito Yamashita of ATR, Japan for the discussion on [4], which produced the motivation of this work.

References

1. Hinton, G.E., van Camp, D.: Keeping Neural Networks Simple by Minimizing the Description Length of the Weights. In: Proc. of COLT. (1993) 5–13
2. Attias, H.: Inferring Parameters and Structure of Latent Variable Models by Variational Bayes. In: Proc. of UAI. (1999)
3. Neal, R.M.: Bayesian Learning for Neural Networks. Springer (1996)
4. Sato, M., Yoshioka, T., Kajihara, S., Toyama, K., Goda, N., Doya, K., Kawato, M.: Hierarchical Bayesian Estimation for MEG inverse problem. Neuro Image **23** (2004) 806–826
5. James, W., Stein, C.: Estimation with Quadratic Loss. In: Proc. of the 4th Berkeley Symp. on Math. Stat. and Prob. (1961) 361–379
6. Nakajima, S., Watanabe, S.: Generalization Error and Free Energy of Variational Bayes Approach of Linear Neural Networks. In: Proc. of ICONIP, Taipei, Taiwan (2005) 55–60
7. Callen, H.B.: Thermodynamics. Wiley (1960)
8. Hamalainen, M., Hari, R., Ilmoniemi, R.J., Knuutila, J., Lounasmaa, O.V.: Magnetoencephalography — Theory, Instrumentation, and Applications to Noninvasive Studies of the Working Human Brain. Rev. Modern Phys. **65** (1993) 413–497
9. Nakajima, S., Watanabe, S.: Generalization Performance of Subspace Bayes Approach in Linear Neural Networks. IEICE Trans. **E89-D** (2006) 1128–1138
10. Efron, B., Morris, C.: Stein's Estimation Rule and its Competitors—an Empirical Bayes Approach. J. of Am. Stat. Assoc. **68** (1973) 117–130

A Definition of Linear Neural Networks

Let $x \in \mathbb{R}^M$ be an input vector, $y \in \mathbb{R}^N$ an output vector, and w a parameter vector. Assume that the output is observed with a noise subject to $\mathcal{N}_N(0, \Sigma)$. Then, the probability density of a three-layer linear neural network model (LNN) with H hidden units, also known as the reduced rank regression model with rank H, is given by

$$p(y|x, A, B) = \mathcal{N}_N(BAx, \Sigma), \tag{38}$$

where A and B are an $H \times M$ and an $N \times H$ parameter matrices, respectively. It has been proved that, in LNNs, the VB approach is asymptotically equivalent to a positive-part James-Stein type shrinkage estimation, and its generalization error has been clarified [6].

B James-Stein Type Shrinkage Estimator

A positive-part James-Stein type shrinkage estimator [5, 10], which can dominate the maximum likelihood (ML) estimator, of the parameter w is defined by

$$\hat{w}_{PJS} = \theta(n\|\hat{w}_{MLE}\|^2 > \chi)\left(1 - \chi/n\|\hat{w}_{MLE}\|^2\right)\hat{w}_{MLE} \equiv \mathcal{S}(\hat{w}_{MLE}; \chi), \tag{39}$$

where $\hat{w}_{MLE}$ is the ML estimator, $\theta(\cdot)$ is the indicator function of an event, and $\chi > 0$ is a constant, called the degree of shrinkage in this paper.

Nonnegative Matrix Factorization for Motor Imagery EEG Classification

Hyekyoung Lee[1], Andrzej Cichocki[2], and Seungjin Choi[1]

[1] Department of Computer Science
Pohang University of Science and Technology
San 31 Hyoja-dong, Nam-gu, Pohang 790-784, Korea
{leehk, seungjin}@postech.ac.kr
[2] Laboratory for Advanced Brain Signal Processing
Brain Science Institute, RIKEN
2-1 Hirosawa, Wako-shi, Saitama 351-0198, Japan
cia@brain.riken.jp

Abstract. In this paper, we present a method of feature extraction for motor imagery single trial EEG classification, where we exploit nonnegative matrix factorization (NMF) to select discriminative features in the time-frequency representation of EEG. Experimental results with motor imagery EEG data in BCI competition 2003, show that the method indeed finds meaningful EEG features automatically, while some existing methods should undergo cross-validation to find them.

1 Introduction

Brain computer interface (BCI) is a system that is designed to translate a subject's intention or mind into a control signal for a device such as a computer, a wheelchair, or a neuroprosthesis [1]. BCI provides a new communication channel between human brain and computer and adds a new dimension to human computer interface (HCI). It was motivated by the hope of creating new communication channels for disabled persons, but recently draws attention in multimedia communication, too [2].

The most popular sensory signal used for BCI is electroencephalogram (EEG) which is the multivariate time series data where electrical potentials induced by brain activities are recorded in a scalp. Exemplary spectral characteristics of EEG involving motor, might be μ rhythm (8-12 Hz) and β rhythm (18-25 Hz) which decrease during movement or in preparation for movement (event-related desynchronization, ERD) and increase after movement and in relaxation (event-related synchronization, ERS) [1]. ERD and ERS could be used as relevant features for the task of motor imagery EEG classification. However those phenomena might happen in a different frequency band for some subjects, for instance, in 16-20 Hz, not in 8-12 Hz [3]. Moreover, it is not guaranteed that a subject always concentrates on imagination during experiments. Thus, it is desirable to determine appropriate activated frequencies and associated features for each subject, during motor imagery experiments.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 250–259, 2006.

In this paper we present a method of discriminative feature extraction where we exploit the sparseness, L_1 norm, and nonnegative matrix factorization (NMF). Morlet wavelets are used to construct a nonnegative data matrix from the time-domain EEG data. We use the NMF with α-divergence that was recently proposed in [4,5,6]. The method is applied to the task of single-trial online classification of imaginary left and right hand movements using Data Set III of BCI competition 2003. As in [7], we use Gaussian probabilistic models for classification, where Gaussian class-conditional probabilities for a single point in time t are integrated temporally by taking the expectation of the class probabilities with respect to the discriminative power at each point in time. Numerical experiments show that our NMF-based method learns basis vectors indicating discriminative frequencies and determine useful features for the task of single-trial online classification of imaginary left and right hand movements.

2 Nonnegative Matrix Factorization

NMF is one of widely-used multivariate analysis methods for nonnegative data, which has many potential applications in pattern recognition and machine learning [8,9,10]. Suppose that N observed m-dimensional data points, $\{\boldsymbol{x}(t)\}$, $t = 1, \ldots, N$ are available. Denote the data matrix by $\boldsymbol{X} = [\boldsymbol{x}(1) \cdots \boldsymbol{x}(N)] = [X_{ij}] \in \mathbb{R}^{m \times N}$. NMF seeks a decomposition of the nonnegative data matrix $\boldsymbol{X}$ that is of the form:

$$\boldsymbol{X} \approx \boldsymbol{A}\boldsymbol{S}, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ contains basis vectors in its columns and $\boldsymbol{S} \in \mathbb{R}^{n \times N}$ is the associated encoding variable matrix. Both matrices $\boldsymbol{A}$ and $\boldsymbol{S}$ are restricted to have only nonnegative elements in the decomposition.

Various error measures for the factorization (1) with nonnegativity constraints, can be considered. Recently, Amari's α-divergence and its multiplicative algorithm were proposed in [5,6]. The α-divergence between $\boldsymbol{X}$ and $\boldsymbol{A}\boldsymbol{S}$ is given by

$$D_\alpha[\boldsymbol{X} \parallel \boldsymbol{A}\boldsymbol{S}] = \frac{1}{\alpha(1-\alpha)} \sum_{i,j} \left[\alpha X_{ij} + (1-\alpha)[\boldsymbol{A}\boldsymbol{S}]_{ij} - X_{ij}^{\alpha}[\boldsymbol{A}\boldsymbol{S}]_{ij}^{1-\alpha}\right]. \tag{2}$$

The α-divergence is a parametric family of divergence functional, including several well-known divergence measure: (1) KL divergence of $\boldsymbol{X}$ from $\boldsymbol{A}\boldsymbol{S}$ for $\alpha = 0$; (2) Hellinger divergence for $\alpha = 1/2$; (3) KL divergence of $\boldsymbol{A}\boldsymbol{S}$ from $\boldsymbol{X}$ for $\alpha = 1$; (4) χ^2-divergence for $\alpha = 2$. The parameter α is associated with the characteristics of a learning machine, in the sense that the model distribution is more inclusive (as α goes to ∞) more exclusive (as α approaches $-\infty$). The multiplicative algorithm regarding the minimization of the α-divergence of $\boldsymbol{A}\boldsymbol{S}$ from $\boldsymbol{X}$ in (2), is given by

$$S_{ij} \leftarrow S_{ij} \left[\frac{\Sigma_k [A_{ki}(X_{kj}/[\boldsymbol{A}\boldsymbol{S}]_{kl})^{\alpha}]}{\Sigma_l A_{li}}\right]^{\frac{1}{\alpha}}, \tag{3}$$

$$A_{ij} \leftarrow A_{ij} \left[\frac{\Sigma_k [S_{jk}(X_{ik}/[\boldsymbol{AS}]_{ik})^{\alpha}]}{\Sigma_l S_{jl}} \right]^{\frac{1}{\alpha}}. \tag{4}$$

More details on algorithms (3) and (4) can be found in [6].

3 Proposed Method

The overall structure of our proposed single trial EEG classification is illustrated in Fig. 1, where the method consists of three steps: (1) preprocessing involving wavelet transform; (2) NMF-based feature extraction; (3) probabilistic model-based classification. Each of these steps is described in detail.

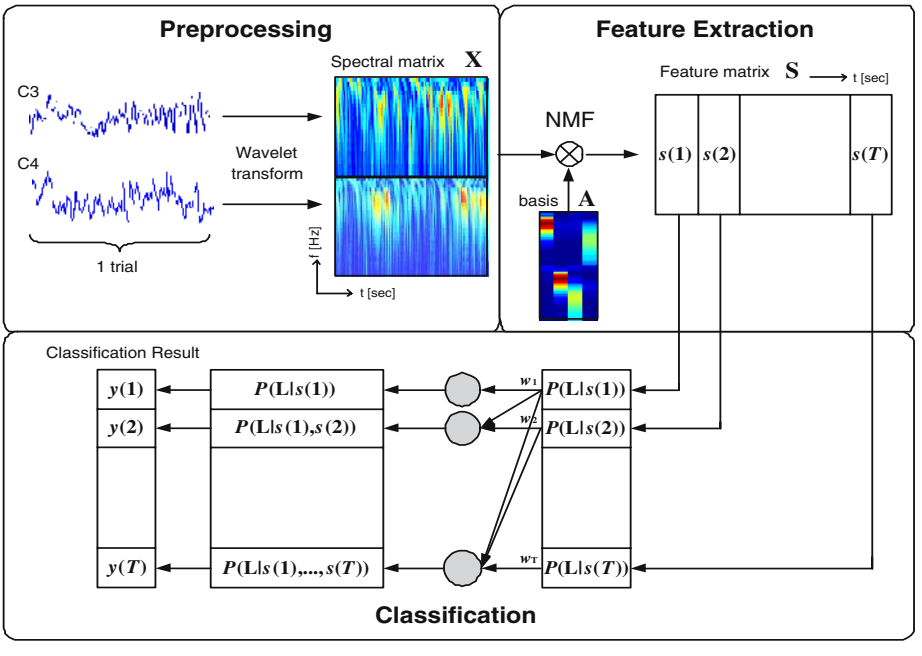


Fig. 1. The overall structure of the proposed EEG classification method is shown. In preprocessing, time-domain EEG waveforms are transformed into time-frequency representation by the Morlet wavelet transform. NMF is applied to determine representative basis vectors and associated with discriminant features. A probabilistic model-based classifier takes NMF-based features as inputs to make a decision.

3.1 Data Description

For our empirical study, we used one of BCI competition 2003 data sets, which was provided by the Department of Medical Informatics, Institute for Biomedical Engineering, Graz University of Technology, Austria [11]. The data set involves

left/right imagery hand movements and consists of 140 labelled trials for training and 140 unlabelled trials for test. Each trial has a duration of 9 seconds, where a visual cue (arrow) is presented pointing to the left or the right after 3-second preparation period and imagination task is carried out for 6 seconds. It contains EEG acquired from three different channels (with sampling frequency 128 Hz) C_3, C_z and C_4. In our study we use only two channels, C_3 and C_4, because ERD has contralateral dominance and C_z channel contains little information for discriminant analysis.

3.2 Preprocessing

We obtain the time-frequency representation of the EEG data, by filtering it with complex Morlet wavelets, where the mother wavelet is given by

$$\Psi_0(\eta) = \pi^{-1/4} e^{iw_0\eta} e^{-\eta^2/2}, \tag{5}$$

where w_0 is the characteristic eigenfrequency (generally taken to be 6). Scaling and temporal shifting of the mother wavelet, leads to $\Psi_{\tau,d(f)}$ controlled by the factor $\eta = (t-\tau)/d(f)$ where

$$d(f) = \frac{w_0 + \sqrt{2+w_0^2}}{4\pi f}, \tag{6}$$

where f is the main receptive frequency.

We denote by $C_{3,k}(t)$ and $C_{4,k}(t)$ the EEG waveforms measured from C_3 and C_4 channels, in the kth trial. The wavelet transform of $C_{i,k}(t)$ $(i = 3, 4)$ at time τ and frequency f is their convolution with scaled and shifted wavelets. The amplitude of the wavelet transform, $x_{i,k}(f, \tau)$, is given by

$$x_{i,k}(f,\tau) = \| \, C_{i,k}(t) * \Psi_{\tau,d(f)}(t) \, \|, \tag{7}$$

for $i = 3, 4$ and $k = 1, \ldots, K$ where K is the number of trials. Concatenating those amplitudes for $i = 3, 4$ and $(f_1, \ldots, f_{27}) = [4, \ldots, 30]$ Hz, leads to the vector $\boldsymbol{x}_k(t) \in \mathbb{R}^{54}$ that is of the form

$$\boldsymbol{x}_k(t) = [x_{3,k}(f_1,t) \cdots x_{3,k}(f_{27},t) \quad x_{4,k}(f_1,t) \cdots x_{4,k}(f_{27},t)]^\top. \tag{8}$$

Incorporating with T data points in each trial, we construct

$$\boldsymbol{X}_k = [\boldsymbol{x}_k(1) \cdots \boldsymbol{x}_k(T)] \in \mathbb{R}^{54\times T}. \tag{9}$$

Collecting K trials leads to the data matrix

$$\boldsymbol{X} = [\boldsymbol{X}_1 \cdots \boldsymbol{X}_K] \in \mathbb{R}^{54\times KT}. \tag{10}$$

Labelled and unlabelled data are distinguished by $\boldsymbol{X}_{train}$ and $\boldsymbol{X}_{test}$, respectively.

3.3 Feature Extraction

We extract feature vectors by applying NMF to the data matrix $\boldsymbol{X}$ constructed from the wavelet transform of EEG over the frequency range $f \in [4, \ldots, 30]$ Hz. The data matrix $\boldsymbol{X} \in \mathbb{R}^{54 \times KT}$ contains a large number of data vectors reflecting K trials and T data points of EEG. Instead of using the whole data vectors, we first select candidate vectors which are expected to be more discriminative, then use only those candidate vectors as inputs to NMF, in order to determine the basis matrix $\boldsymbol{A}$. The power spectrum in the localized frequency range such as μ or β band of C_3 and C_4 channels, is activated during the imagination of movement. Thus, we investigate the power and sparseness of each data vector to select candidate vectors. We use the sparseness measure proposed by Hoyer [12], described by

$$\xi(\boldsymbol{x}) = \frac{\sqrt{m} - (\sum |x_i|)/\sqrt{\sum x_i^2}}{\sqrt{m} - 1}, \tag{11}$$

where x_i is the ith element of the m-dimensional vector $\boldsymbol{x}$.

The candidate vector selection is performed in the following way. First, we compute the power of each column of $\boldsymbol{X}$, by summing its elements. For example, the power of $\boldsymbol{x}_i$, $\phi(\boldsymbol{x}_i)$, is the sum of all elements in $\boldsymbol{x}_i$, i.e., $\phi(\boldsymbol{x}_i) = \sum_{j=1}^{54} x_{ji}$ where x_{ji} is the jth element of the vector $\boldsymbol{x}_i$. The average power $\bar{\phi}$ is computed by

$$\bar{\phi} = \frac{1}{KT} \sum_{i=1}^{KT} \phi(\boldsymbol{x}_i). \tag{12}$$

The sparseness is computed for C_3 and C_4 channels, and each averaged sparseness is added, leading to the average sparseness. Data contributed by C_3 channel, corresponds to first 27 row vectors of $\boldsymbol{X}$ and the rest of row vectors are related to C_4 channels. For each column of $\boldsymbol{X}$, the sparseness is calculated for C_3 and C_4 channels, by considering the first 27 rows and the last 27 rows of $\boldsymbol{X}$, respectively. Averaged sparseness values for each channel are computed, then they are added, leading to the final average sparseness. We select candidate vectors from $\boldsymbol{X}$ if the data vector has the power greater than the average power and has the sparseness greater than 70% of the average sparseness.

We apply the NMF algorithm in (3) and (4), to the candidate data matrix $\widetilde{\boldsymbol{X}}$, leading to $\widetilde{\boldsymbol{X}} = \boldsymbol{A}\widetilde{\boldsymbol{S}}$. Then the basis matrix $\boldsymbol{A}$ is used to infer associated features $\boldsymbol{S}$, by applying the algorithm (3) to the original data matrix $\boldsymbol{X}$ with $\boldsymbol{A}$ fixed. In other words, the candidate matrix $\widetilde{\boldsymbol{X}}$ is used to determine the basis matrix $\boldsymbol{A}$ and the encoding variable matrix $\boldsymbol{S}$ (feature vectors) is inferred using the original data matrix $\boldsymbol{S}$. In our experiments, about 31% of data vectors were selected as candidate vectors. In our empirical study, basis vectors determined by the NMF of candidate vectors, showed better characteristics than those computed by the NMF of whole data vectors.

3.4 Classification

We denote by $y_k \in \{\mathrm{L}, \mathrm{R}\}$ the class label for the left or the right in the kth trial. Feature vectors $\boldsymbol{S} \in \mathbb{R}^{54 \times KT}$ consists of $\boldsymbol{s}_k(t)$ for $k = 1, \ldots, K$ and $t = 1, \ldots, T$.

For classification, we use the probabilistic model-based classifier proposed in [7], where Gaussian class-conditional densities for a single data point in time t are integrated temporally by taking the expectation of the class probabilities with respect to the discriminative power at each point in time. We assume feature vectors $\boldsymbol{s}(t)$ (the subscript k associated with trials, is left out if not necessary) follow Gaussian distribution at any time point $t \in [3, 9]$ sec, i.e.,

$$p(\boldsymbol{s}(t) \mid y) = \frac{1}{\mid 2\pi \boldsymbol{\Sigma}_{y,t} \mid^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2} \left(\boldsymbol{s}(t) - \boldsymbol{\mu}_{y,t}\right)^{\top} \boldsymbol{\Sigma}_{y,t}^{-1} \left(\boldsymbol{s}(t) - \boldsymbol{\mu}_{y,t}\right) \right\}, \quad (13)$$

where $\boldsymbol{\mu}_{y,t}$ and $\boldsymbol{\Sigma}_{y,t}$ are the mean vector and the covariance matrix for each class labelled by $y \in \{L, R\}$. These are estimated using features associated with labelled data, i.e.,

$$\boldsymbol{\mu}_{y,t} = E[\boldsymbol{s}_{y,k}(t)], \quad (14)$$

$$\boldsymbol{\Sigma}_{y,t} = E[(\boldsymbol{s}_{y,k}(t) - \boldsymbol{\mu}_{y,t})(\boldsymbol{s}_{y,k}(t) - \boldsymbol{\mu}_{y,t})^T]. \quad (15)$$

The prediction of the class label at time t, is performed using the posterior probability determined by Bayes rule:

$$p(y \mid \boldsymbol{s}(t)) = \frac{p(\boldsymbol{s}(t) \mid y)}{p(\boldsymbol{s}(t) \mid L) + p(\boldsymbol{s}(t) \mid R)}. \quad (16)$$

This posterior probability allows us to make a decision for the class label, at a single point in time. However, it is more desirable to take information across time into account. To this end, we consider

$$p(y \mid \boldsymbol{s}(1), \ldots, \boldsymbol{s}(t_0)) = \frac{\sum_{t \le t_0} w_t p(y \mid \boldsymbol{s}(t))}{\sum_{t \le t_0} w_t}, \quad (17)$$

where w_t are weights reflecting the discriminant power that is determined by minimizing Bayes misclassification error.

The Bayes error is defined by

$$p(\text{error}) = \int p(\text{error} \mid s(t)) p(s(t)) ds, \quad (18)$$

where

$$p(\text{error} \mid s(t)) = \min\left[p(L \mid s(t)), p(R \mid s(t))\right], \quad (19)$$

Following from the Chernoff bound

$$\min[a, b] \le a^{\beta} b^{1-\beta}, \quad a, b \ge 0, \quad 0 \le \beta \le 1, \quad (20)$$

its upper-bound is given by

$$p(\text{error}) \leq \int \{p(L \mid s(t))p(s(t))\}^{\beta_t} \{p(R \mid s(t))p(s(t))\}^{1-\beta_t} ds$$
$$= p(L)^{\beta_t} p(R)^{1-\beta_t} \int p(s(t) \mid L)^{\beta_t} p(s(t) \mid R)^{1-\beta_t} ds. \quad (21)$$

The larger the discriminant power is , the smaller the Bayes error is. Thus, weights are determined by

$$2w_t = 1 - \min_{0 \leq \beta_t \leq 1} \int p(s(t) \mid L)^{\beta_t} p(s(t) \mid R)^{1-\beta_t} ds. \quad (22)$$

The class label y by combining the information through t_0 is determined by

$$y = \begin{cases} L & \text{if } p(L \mid \boldsymbol{s}(1), \ldots, \boldsymbol{s}(t_0)) > p(R \mid \boldsymbol{s}(1), \ldots, \boldsymbol{s}(t_0)), \\ R & \text{otherwise.} \end{cases} \quad (23)$$

4 Numerical Experiments

We apply the proposed method to the single-trial online classification of imaginary left and right hand movements in BCI competition 2003 (Data Set III). The time-domain EEG data is transformed into the time-frequency representation by complex Morlet wavelets with $w_0 = 6$, $f = [4, \ldots, 30]$ Hz, and $\tau = [3, \ldots, 9]$ sec using (7). We select candidate spectral vectors using the method described in Sec. 3.3. Then we apply the NMF algorithm in (4) and (3) with $\alpha = 0.5, 1, 2$ and $n = 2, 4, 5, 6$ (the number of basis vectors), in order to estimate basis vectors that are shown in Fig. 2. As the number of basis vector increases, the spectral components such as μ rhythm (8-12 Hz), β rhythm (18-22 Hz), and sensori-motor rhythm (12-16 Hz), appear in the order of their importance. All rhythms have the property of contralateral dominance, so they are present in basis vectors associated with C_3 or C_4 channel, separately.

In our empirical study, the best performance was achieved when $\alpha = 0.5 or 1$ and $n = 5$ (5 basis vectors). The single trial on-line classification result, is shown in Fig. 3, where the classification accuracy is shown in (a) and the mutual information between the true class label and the estimated class label is plotted in (b). The classification accuracy is suddenly raised from 3.43 sec. The maximal classification accuracy is 88.57 % at 6.05 sec, which is higher than the result without the data selection step in the training phase (86.43 % at 7.14 sec). The mutual information (MI) hits the maximum, 0.6549 bit, which occurs at 6.05 sec. The result is better than the one achieved by the BCI competition 2003 winner (0.61 bit). Table 1 show the maximum mutual information in the time courses per a trial varying the value of α and the number of basis. The smaller the value of α, the better the mutual information, however, α is not critical of determining the performance.

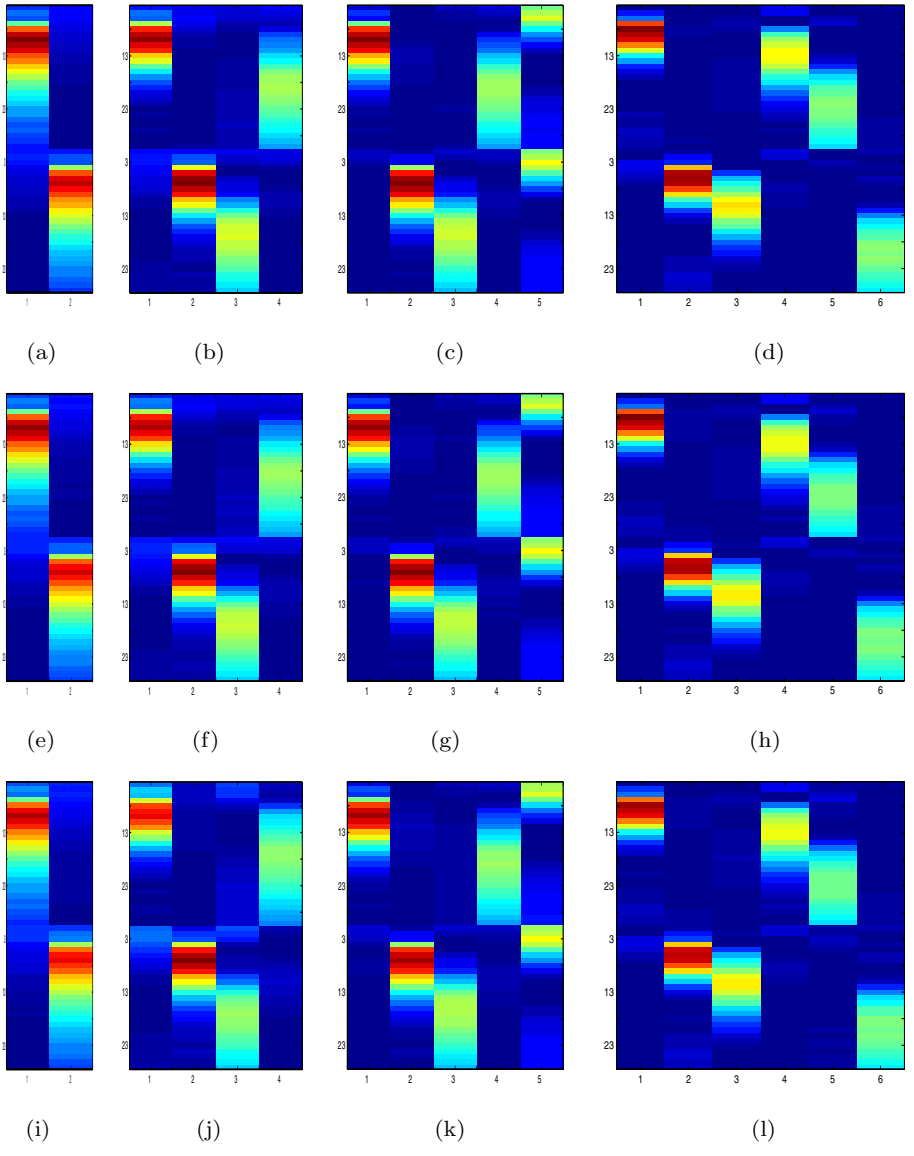
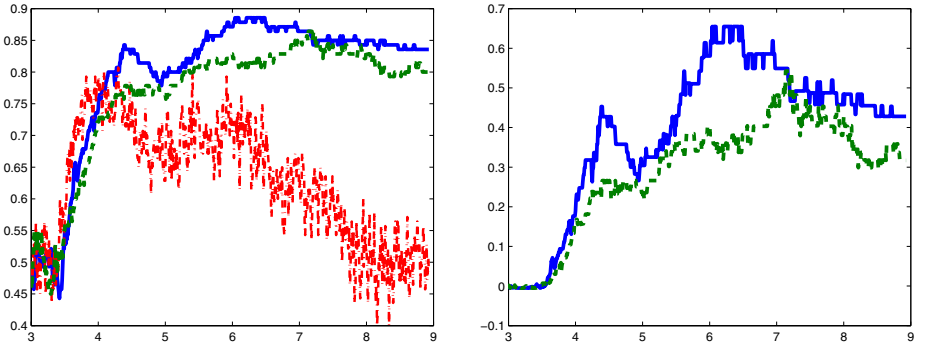


Fig. 2. Basis vectors determined by NMF are shown in the case of $\alpha = 0.5, 1, 2$ (from top to bottom) and $n = 2, 4, 5, 6$ (from left to right). In each plot, top $1/2$ is associated with C_3 and bottom $1/2$ is contributed by C_4. In each of those, the vertical axis represents frequencies between 4 and 30 Hz, the horizon axis is related to the number of basis vectors. Basis vectors reveals some useful characteristics: (1) μ rhythm (8-12 Hz); (2) β rhythm (18-22 Hz); (3) sensori-motor rhythm (12-16 Hz). ERD has the contralateral dominance, hence each rhythm occurs in each channel separately. Different values of α do not have much influence on basis vectors. However, it is observed that the lager the value of α is, the more smooth the distribution of basis vector is.



(a) (b)

Fig. 3. The on-line classification result is shown in terms of: (a) the classification accuracy; (b) the mutual information between the true class label and the estimated class label. In both plots, dotted lines (green color) are results without candidate data selection and solid lines (blue color) are results with the proposed data selection method. The data selection method improves the classification accuracy as well as the mutual information. The dot-dashed line (red color) in (a) is the result of the classifier based on the Gaussian probabilistic model taking a single time point into account. Combining the information across time, really improves the classification accuracy.

Table 1. Mutual information for different values of α and for different number of basis vectors

α	number of basis				
	2	4	5	6	7
0.5	0.5545	0.5803	0.6549	0.6256	0.5875
1	0.5545	0.5803	0.6549	0.6256	0.5803
2	0.5408	0.5745	0.6404	0.6256	0.5803

5 Conclusion

We have presented an NMF-based method of feature extraction for on-line classification of motor imagery EEG data. We have also introduced a method of data selection where the power and the sparseness was exploited. Empirical results confirmed that the data selection scheme really improved the classification accuracy by 2.14 % and the mutual information by 0.1127 bit. Existing methods should undergo the cross-validation several times, in order to select discriminative frequency features. However, we have shown that our NMF-based method could find discriminative and representative basis vectors (which reflected appropriate spectral characteristics) without the cross-validation, which improved the on-line classification accuracy. Our method improved the mutual information achieved by BCI competition 2003 winner, by 0.0449 bit, where two frequencies

(10 and 22 Hz) were selected using the leave-one-out cross validation. The value of α in the NMF algorithm, was not critical in our empirical study. However, it was confirmed that the parameter α is associated with the characteristics of a learning machine, showing that distributions of basis vectors become more smooth, as α goes to ∞.

Acknowledgments. This work was supported by KOSEF International Cooperative Research Program and KOSEF Basic Research Program (grant R01-2006-000-11142-0).

References

1. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. Clinical Neurophysiology **113** (2002) 767–791
2. Ebrahimi, T., Vesin, J.F., Garcia, G.: Brain-computer interface in multimedia communication. IEEE Signal Processing Magazine **20** (2003) 14–24
3. Lal, T.N., Schroder, M., Hinterberger, T., Weston, J., Bogdan, M., Birbaumer, N., Schölkopf, B.: Support vector channel selection in BCI. Technical Report 120, Max Planck Institute for Biological Cybernetics (2003)
4. Cichocki, A., Zdunek, R., Amari, S.: Csiszár's divergences for non-negative matrix factorization: Family of new algorithms. In: Proc. Int'l Conf. Independent Component Analysis and Blind Signal Separation, Charleston, South Carolina (2006)
5. Cichocki, A., Zdunek, R., Amari, S.: New algorithms for non-negative matrix factorization in applications to blind source separation. In: Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing, Toulouse, France (2006)
6. Cichocki, A., Choi, S.: Nonnegative matrix factorization with α-divergence. Pattern Recognition Letters (2006) submitted.
7. Lemm, S., Schäfer, C., Curio, G.: BCI competition 2003-data set III: Probabilistic modeling of sensorimotor μ rhythms for classification of imaginary hand movements. IEEE Trans. Biomedical Engineering **51** (2004)
8. Paatero, P., Tapper, U.: Least squares formulation of robust non-negative factor analysis. Chemometrics Intelligent Laboratory Systems **37** (1997) 23–35
9. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401** (1999) 788–791
10. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems. Volume 13., MIT Press (2001)
11. Blankertz, B., Müller, K.R., Curio, G., Vaughan, T.M., Schalk, G., Wolpaw, J.R., Schlögl, A., Neuper, C., Pfurtscheller, G., Hinterberger, T., Schroder, M., Birbaumer, N.: The BCI competition 2003: Progress and perspectives in detection and discrimination of EEG single trials. IEEE Trans. Biomedical Engineering **51** (2004)
12. Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints. Journal of Machine Learning Research **5** (2004) 1457–1469

Local Factor Analysis with Automatic Model Selection: A Comparative Study and Digits Recognition Application

Lei Shi and Lei Xu

Chinese University of Hong Kong, Shatin, NT, Hong Kong
{shil, lxu}@cse.cuhk.edu.hk

Abstract. A further investigation is made on an adaptive local factor analysis algorithm from Bayesian Ying-Yang (BYY) harmony learning, which makes parameter learning with automatic determination of both the component number and the factor number in each component. A comparative study has been conducted on simulated data sets and several real problem data sets. The algorithm has been compared with not only a recent approach called Incremental Mixture of Factor Analysers (IMoFA) but also the conventional two-stage implementation of maximum likelihood (ML) plus model selection, namely, using the EM algorithm for parameter learning on a series candidate models, and selecting one best candidate by AIC, CAIC, and BIC. Experiments have shown that IMoFA and ML-BIC outperform ML-AIC or ML-CAIC while the BYY harmony learning considerably outperforms IMoFA and ML-BIC. Furthermore, this BYY learning algorithm has been applied to the popular MNIST database for digits recognition with a promising performance.

1 Introduction

Clustering and dimension reduction have been considered as two of the fundamental problems in the literature of unsupervised learning. It is well known that Gaussian mixture model (GMM) with full covariance matrices requires sufficient training data to guarantee the reliability of the estimated model parameters, while GMM with diagonal covariance matrices requires a relatively large number of Gaussians to provide high recognition performance. Local factor analysis (LFA) (also called mixture of factor analyzers (MFA)) combines the widely-used GMM model with one well known dimension reduction approach, namely factor analysis (FA). Via local structure analysis, LFA is able to reduce the freedom degree of covariance matrices to achieve a good generalization. Several efforts have been made on such a topic of local dimensionality reduction [2,3,13].

In the literature of LFA research, the conventional method performs the maximum likelihood (ML) learning in help of one of typical statistical criteria to select both component number and local dimensions of local factor analysis. However, it suffers a huge computing cost. Bayesian Ying-Yang (BYY) learning was proposed as a unified statistical learning framework firstly in 1994 and systematically developed in the past decade. Providing a general learning framework,

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 260–269, 2006.

BYY harmony learning consists of a general BYY system and a fundamental harmony learning principle as a unified guide for developing new regularization techniques, a new class of criteria for model selection, and a new family of algorithms that perform parameter learning with automatic model selection. Readers are referred to [15,17] for a recent systematical review. Applying the BYY harmony learning to local factor analysis, an adaptive learning algorithm has been developed that performs local factor analysis with both the local dimensions of each component and the number of components automatically determined during parameter learning [14,16].

This paper investigates the automatic BYY harmony learning based LFA, in comparison with the ML learning via criteria of AIC, CAIC, BIC, as well as a recently proposed approach called Incremental Mixture of Factor Analyzers (IMoFA)[11] that makes an increasing model selection during learning. A comparative study is conducted via experiments on not only simulated data but also several real problem data sets, as well as a popular digit recognition database, respectively. The rest of this paper is organized as follows. In Section 2, we review FA and LFA, together with typical statistical criteria and the recent proposed algorithm IMoFA. Section 3 will further introduce the BYY harmony learning based LFA. After a series of comparative experiments in Section 4, we apply the BYY-LFA to the popular MNIST database of digit recognition in Section 5. Finally, we conclude in Section 6 and make further discussion in Section 7.

2 FA and LFA

2.1 Factor Analysis Model

Factor analysis (FA) is a classical dimension reduction technique aiming to find the hidden causes and sources [8]. Provided a d-dimensional vector of observable variables $\mathbf{x}$, the FA model is given by $\mathbf{x} = \mathbf{A}\mathbf{y} + \mathbf{e}$, where $\mathbf{A}$ is a $d \times k$ loading matrix, $\mathbf{y}$ is a m-dimensional unobservable latent vector assumed from Gaussian $G(\mathbf{y}|\mathbf{0}, \mathbf{I_k})$ with $m < d$ generally, $\mathbf{e}$ is a d-dimensional random noise vector assumed from Gaussian $G(\mathbf{e}|\mathbf{0}, \mathbf{\Psi})$ with $\mathbf{\Psi}$ being a diagonal matrix. Moreover, $\mathbf{y}$ and $\mathbf{e}$ are mutually independent. Therefore, $\mathbf{x}$ is distributed with zero mean and covariance $\mathbf{A}\mathbf{A}^T + \mathbf{\Psi}$. The goal of FA is to find $\theta = \{\mathbf{A}, \mathbf{\Psi}\}$ that best models the structure of $\mathbf{x}$. One widely used method to estimate θ is the maximum likelihood (ML) learning that maximizes the log-likelihood function, usually implemented by the expectation-maximization (EM) algorithm [1,8].

2.2 Local Factor Analysis

Local factor analysis (LFA) (or also called mixture of factor analyzers (MFA)), is a useful multivariate analysis tool to explore not only clusters but also local subspaces with wide applications including pattern recognition, bioinformatics, and financial engineering [14,10]. LFA performs clustering analysis and dimension reduction in each cluster (component) simultaneously. Provided $\mathbf{x}$ as a d-dimensional random vector of observable variables, the mixture model assumes

that $\mathbf{x}$ is distributed according to a mixture of k underlying probability distributions $p(\mathbf{x}) = \sum_{l=1}^{k} \alpha_l p_l(\mathbf{x})$, where $p_l(\mathbf{x})$ is the density of the lth component in the mixture, and α_l is the probability that an observation belongs to the lth component with $\alpha_l \geq 0, l = 1, \ldots, k$, and $\sum_{l=1}^{k} \alpha_l = 1$. For LFA, it is further assumed that each $p_l(\mathbf{x})$ is modelled by a single FA [14]. That is, we have

$$p_l(\mathbf{x}|\mathbf{y}) = G(\mathbf{x}|\mathbf{A}_l\mathbf{y} + \mathbf{c}_l, \mathbf{\Psi}_l), \quad p_l(\mathbf{y}) = G(\mathbf{y}|\mathbf{0}, \mathbf{I}_{m_l}), \tag{1}$$

$$p_l(\mathbf{x}) = \int p_l(\mathbf{x}|\mathbf{y}) p_l(\mathbf{y}) d\mathbf{y} = G(\mathbf{x}|\mathbf{c}_l, \mathbf{A}_l\mathbf{A}_l^T + \mathbf{\Psi}_l), \tag{2}$$

where $\mathbf{y}$ is a m_l-dimensional unobservable latent vector, $\mathbf{A}_l$ is a $d \times m_l$ loading matrix, $\mathbf{c}_l$ is a d-dimensional mean vector, $\mathbf{\Psi}_l$ is a diagonal matrix, $l = 1, 2, \ldots, k$.

For a set of observations $\{\mathbf{x}_t\}_{t=1}^{n}$, supposing that the number of components k and the numbers of local factors $\{m_l\}$ are given, one widely used method to estimate the unknown parameters $\theta = \{\alpha_l, \mathbf{A}_l, \mathbf{c}_l, \mathbf{\Psi}_l\}_{l=1}^{k}$ is the maximum likelihood (ML) learning, which can be effectively implemented by expectation-maximization (EM) algorithm [1,3].

2.3 Conventional Statistical Criteria

Two important problems for LFA are how to select the number of Gaussian components k and how to decide the numbers of sub-factors $\{m_l\}_{l=1}^{k}$. They can be addressed in a *two-phase* procedure in help of typical statistical model selection criteria such as Akaike's information criterion (AIC) [4], Bozdogan's consistent Akaike's information criterion (CAIC) [6], Schwarz's Bayesian inference criterion (BIC) [9] which coincides with Rissanen's minimum description length (MDL) criterion [5]. These criteria are based on the maximum likelihood (ML) estimates of parameters which can be obtained by the EM algorithm [3,8], summarized into the following general form:

$$J(\hat{\theta}, k) = -2L(\hat{\theta}) + C(n)D(k) \tag{3}$$

where $L(\hat{\theta})$ is the log likelihood based on the ML estimate $\hat{\theta}$ under a given k, $D(k)$ is the number of the independent parameters in a corresponding model, and $C(n)$ is a function with respect to the number of observations as follows:

$$C(n) = \begin{cases} 2, & \text{for AIC;} \\ ln(n) + 1, & \text{for CAIC;} \\ ln(n), & \text{for BIC and MDL;} \end{cases} \tag{4}$$

For LFA, the number of free parameters is

$$D(k, \{m_l\}) = k - 1 + kd + kd + \sum_{l=1}^{k} (dm_l - m_l(m_l - 1)/2).$$

In the first phase, two ranges of $k \in [k_{min}, k_{max}]$ and $m_l \in [m_{min}, m_{max}]$ are selected to form a domain $\mathcal{M}$, assumed to contain the optimal $k^*, \{m_l^*\}_{l=1}^{k^*}$. At each specific choice of $k, \{m_l\}$ in $\mathcal{M}$, the parameters are estimated θ via the

ML learning. In the second phase, selection is made among all candidate models obtained in the first phase according to their criterion values, that is:

$$\hat{k}, \{\hat{m_l}\} = \arg \min_{k,\{m_l\}} \{J(\hat{\theta}, k, \{m_l\}), \{k, \{m_l\}\} \in \mathcal{M}\}, \tag{5}$$

However, in this domain $\mathcal{M}$, we have to implement EM algorithm at least $\sum_{k=k_{min}}^{k_{max}} (m_{max} - m_{min} + 1)^k$ times, which is usually too time-consuming without any knowledge or assumption about the underlying model structure.

2.4 Incremental Mixture of Factor Analysers

Recently, an adaptive algorithm referred as *incremental mixture of factor analysers (IMoFA)* was proposed in [11]. Starting with a 1-factor, 1-component mixture model, in process, IMoFA either splits component or adding local factors according to the validation likelihood, which is terminated when there is no improvement on the validation likelihood. There are two variants IMoFA-L and IMoFA-A for unsupervised and supervised approaches, respectively. In this paper, we consider the unsupervised learning with IMoFA-L, shortly denoted by IMoFA. The detailed procedure and algorithm is referred to [11].

3 BYY Harmony Learning for LFA

Bayesian Ying-Yang (BYY) harmony learning provides a promising tool for local factor analysis with an ability of determining the number of components as well as the number of local factors during parameters learning [14,16,17], which considers the following alternative but equivalent probabilistic FA model:

$$\begin{aligned} p_l(\mathbf{x}|\mathbf{y}) &= G(\mathbf{x}|\mathbf{U}_l\mathbf{y} + \mathbf{c}_l, \mathbf{\Psi}_l),\ \mathbf{U}_l^T\mathbf{U}_l = \mathbf{I}_{m_l},\ p_l(\mathbf{y}) = G(\mathbf{y}|\mathbf{0}, \mathbf{\Lambda}_l), \\ p_l(\mathbf{x}) &= \int p_l(\mathbf{x}|\mathbf{y}) p_l(\mathbf{y}) d\mathbf{y} = G(\mathbf{x}|\mathbf{c}_l, \mathbf{U}_l\mathbf{\Lambda}_l\mathbf{U}_l^T + \mathbf{\Psi}_l), \end{aligned} \tag{6}$$

where $\mathbf{y}$ is still a m_l-dimensional unobservable latent vector, $\mathbf{c}_l$ is a d-dimensional mean vector, $\mathbf{\Lambda}_l$ and $\mathbf{\Psi}_l$ are both diagonal matrices.

Parameters $\theta = \{\alpha_l, \mathbf{U}_l, \mathbf{\Lambda}_l, \mathbf{c}_l, \mathbf{\Psi}_l\}_{l=1}^k$ can be estimated by BYY harmony learning, which may be implemented in several ways. Here, we consider the B-architecture without regularization [15,17], given as follows.

$$\begin{aligned} &\hat{\theta} = \arg\max_{\theta} H(\theta, k), \quad H(\theta, k) = \sum_{t=1}^{n} \sum_{l=1}^{k} P(l|\mathbf{x}_t) \ln[\alpha_l p_l(\mathbf{x}_t|\mathbf{y}_{l,t}) p_l(\mathbf{y}_{l,t})] \\ &\text{subject to } \mathbf{U}_l^T\mathbf{U}_l = \mathbf{I}_{m_l},\ \alpha_l \geq 0,\ \text{and } \sum_{l=1}^{k} \alpha_l = 1,\ l = 1, ..., k. \end{aligned} \tag{7}$$

In a B-architecture, $P(l|\mathbf{x}_t)$ is free and thus it follows from the above maximization that we have

$$\begin{aligned} &P(l|\mathbf{x}_t) = \begin{cases} 1, l = l_t; \\ 0, \text{otherwise.} \end{cases} \\ &l_t = \arg\max_{l} \ln[\alpha_l p_l(\mathbf{x}_t|\mathbf{y}_{l,t}) p_l(\mathbf{y}_{l,t})], \quad \mathbf{y}_{l,t} = \arg\max_{\mathbf{y}} \ln(p_l(\mathbf{x}_t|\mathbf{y}) p_l(\mathbf{y})). \end{aligned} \tag{8}$$

Performing (7) results in maximizing $\ln \alpha_l$, $\ln p_l(\mathbf{x}|\mathbf{y})$ and $\ln p_l(\mathbf{y})$, which will push α_l or $\mathbf{\Psi}_l$ towards zero if component l is extra. Thus we can delete component l if its corresponding α_l or $\mathbf{\Psi}_l$ is approaching to zero. Also, if the latent dimension $\mathbf{y}^{(j)}$ is extra, maximizing $\ln p_l(\mathbf{y})$ will push the variance $\mathbf{\Lambda}_l^{(j)}$ towards zero, thus factor j can be deleted. As long as k and $\{m_l\}$ are initialized at values large enough, they will be determined appropriately and automatically during parameter learning, with details referred to [14,15].

We can estimate $\hat{\theta}$ by an adaptive algorithm obtained from Eq.(7) on Eq.(6). One example is given by Eq.(24) in Section 3.1.3 of [16], which is actually a non-temporal degeneration of the general algorithms given in [14] by its Sec. IV, especially its Table 2 and Eq.(72).

To compare with the EM algorithm in a batch way, here we also consider a batch algorithm to implement Eq.(7), which iterates the following steps:

Yang-step: Get $\mathbf{y}_{l,t}$ by (8) and $P(l|\mathbf{x}_t)$ by (8) for $l = 1, ..., k$ and $t = 1, ..., n$.
Ying-step: Delete the lth component if $\alpha_l \to 0$.

By using a Lagrange multiplier λ and letting the derivatives of the Lagrangian $H(\theta) + \lambda(\sum_{l=1}^{k} \alpha_l - 1)$ respect to λ, α_l, $\mathbf{c}_l$, $\mathbf{\Lambda}_l$, and $\mathbf{\Psi}_l$ equal zero, we get to update

$$\alpha_l^{new} = \frac{1}{n}\sum_{t=1}^{n} P(l|\mathbf{x}_t), \quad c_l^{new} = \frac{1}{n\alpha_l^{new}}\sum_{t=1}^{n}[P(l|\mathbf{x}_t)(\mathbf{x}_t - \mathbf{U}_l\mathbf{y}_{l,t})],$$
$$\mathbf{\Lambda}_l^{new} = diag\{\frac{1}{n\alpha_l^{new}}\sum_{t=1}^{n}[P(l|\mathbf{x}_t)\mathbf{y}_{l,t}\mathbf{y}_{l,t}^T]\},$$
$$\mathbf{\Psi}_l^{new} = diag\{\frac{1}{n\alpha_l^{new}}\sum_{t=1}^{n}[P(l|\mathbf{x}_t)(\mathbf{x}_t - \mathbf{U}_l\mathbf{y}_{l,t} - \mathbf{c}_l)(\mathbf{x}_t - \mathbf{U}_l\mathbf{y}_{l,t} - \mathbf{c}_l)^T]\}.$$

Update $\mathbf{U}_l$ by using gradient ascending on the Stiefel manifold, that is,

$$G_{\mathbf{U}_l} = \frac{1}{n}\mathbf{\Psi}_l^{-1}\{\sum_{t=1}^{n}[P(l|\mathbf{x}_t)(\mathbf{x}_t - \mathbf{c}_l)\mathbf{y}_{l,t}^T] - \mathbf{U}_l\sum_{t=1}^{n}[P(l|\mathbf{x}_t)\mathbf{y}_{l,t}\mathbf{y}_{l,t}^T]\},$$
$$\mathbf{U}_l^{new} = \mathbf{U}_l + \eta_0(G_{\mathbf{U}_l} - \mathbf{U}_l G_{\mathbf{U}_l}^T\mathbf{U}_l). \tag{9}$$

Discard the j-th factor of the lth component if the jth element of $\mathbf{\Lambda}_l$ approximately equals zero.

For classification, we first obtain $M_{lj}, l = 1, \cdots, k_j$ by BYY-LFA for each class $j = 1, \cdots, C$. As a test data $\mathbf{y}_i$ comes, we compute the the likelihoods $p(\mathbf{y}_i|M_{lj}), l = 1, \cdots, k_j, j = 1, \cdots, C$ and find the κ largest ones. Then, we classify $\mathbf{y}_i$ to the class $j^* = arg\max_j \kappa_j$, where κ_j is the account that the κ largest ones share the class label j. This decision rule actually shares the idea of the well known k-NN approach, shortly denoted by a BYY-LFA Rank-κ rule.

4 Empirical Comparative Experiments

In all following experiments, we compare the performances for LFA, including not only the conventional two stage implementation of maximum likelihood (ML)

plus AIC, CAIC, BIC (namely ML-AIC, ML-CAIC, ML-BIC, respectivly), but also IMoFA and BYY. To avoid local optima caused by initialization, we implement both the IMoFA and BYY harmony algorithm for 10 times on each simulation, as well as the EM algorithm for 10 times on each candidate model. Among the ten rounds' learned model, we choose the one with the best likelihood as the result. In order to compare these algorithms in average and facilitate our observing on statistical behaviors, each simulation is repeated 100 times.

4.1 Simulated Data

We arbitrarily generate simulated data sets from Gaussian components with the same $k = 3$ and the same $m = 2$ for each component. We investigate the performances of each method, including ML-AIC, ML-CAIC, ML-BIC, IMoFA, and BYY-LFA, on simulated data sets. The noise variances ψ_l^2 are selected based on ζ_l, which denotes the smallest value in $\mathbf{\Lambda}_l$. After running 100 times for each situation, the experimental results are shown in Fig. 1, where the two rows (A) and (B) represent two situations generated differently. We can observe that BIC, IMoFA, and BYY have the highest correct rates, while AIC has a risk of overestimating both the number of Gaussian components and the number of local factors, but CAIC has a risk of underestimating the number of components.

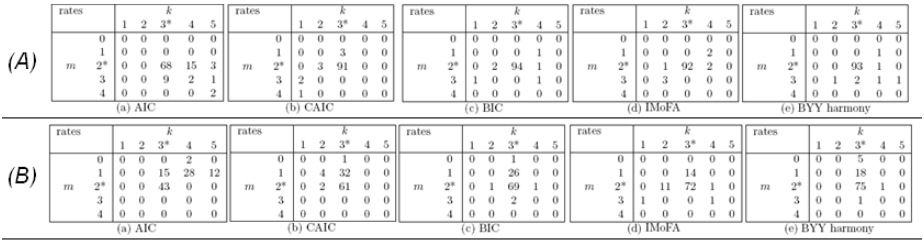


Fig. 1. Comparisons on simulated data sets. Both the two situations are implemented for 100 experiments with $k = 3$ and a same $m_l = 2$. Each row in the figure expressing one situation: (A) $n = 1000$, $d = 5$, $\psi_l^2 = 0.2\zeta_l$; (B) $n = 1000$, $d = 10$, $\psi_l^2 = 0.5\zeta_l$.

Furthermore, we design a 10-dimensional data set, with data generated from $k = 5$ different 200-sample Gaussian components. The numbers of factors are 3, 3, 4, 5, 6, respectively. For the two-phase approaches, we set $k_{max} = 8$, $k_{min} = 2$, and $2 \leq m_l \leq 8$ for each component, while for BYY-LFA we initially set $k_{init} = k_{max} = 8$ and each $m_l = m_{max} = 8$. After 100 repititions, the correct selection frequencies for ML-AIC, ML-CAIC, ML-BIC, IMoFA and BYY-LFA are 78, 86, 91, 94, 96, respectively. The conventional two-phase approaches tends to over-select or under-select the components number, while IMoFA and BYY-LFA can obtain desired results automatically.

4.2 Real World Data

We further test all these LFA implementations with eight real world data sets, in comparison with [11]. As shown in in Fig. 2(a), we consider Pendigits, Optdigits,

Segment and Waveform from UCI repository of machine learning databases[1], ORL from the Olivetti Research Laboratory[2], Vistex from MIT Media Lab[3], Yeast[4], and LVQ from Helsinki Univ. of Technology[5]. We use 10-fold cross-validation on ORL and Yeast to generate the test sets. As noted previously, we repeat EM, IMoFA and BYY 10 times for each simulation, and then the results with the highest likelihood are selected.

Dataset	Training	Test	Dimensions	Classes
PEN	7,494	3,498	16	10
OPT	2,880	1,797	64	10
SEG	700	1,610	14	7
WAVE	300	4,700	21	3
ORL	400	cv10	256	2
VIS	2,700	910	169	10
YEAST	208	cv10	79	5
LVQ	1,929	1,929	20	16

(a) Datasets description.

	Dataset Accuracy							
Methods	PEN	OPT	SEG	WAVE	ORL	VIS	YEAST	LVQ
ML-AIC	94.11	92.73	72.34	71.29	98.37	63.68	88.96	88.13
ML-CAIC	95.17	96.96	84.05	75.84	98.69	62.55	92.41	87.59
ML-BIC	97.73	97.80	78.62	82.71	99.12	68.66	92.36	90.23
ML-CV	95.20	96.91	82.10	75.86	99.01	62.53	92.35	87.56
IMoFA	97.91	92.94	86.13	82.61	98.53	70.52	91.85	89.56
BYY	98.78	97.61	85.44	83.12	98.34	70.80	94.16	90.03

(b) Experimental results for real world data sets.

	CPU Time(in minutes)							
Methods	PEN	OPT	SEG	WAVE	ORL	VIS	YEAST	LVQ
Criteria	171	246	232	154	98	255	192	288
IMoFA	26	49	45	26	14	61	38	65
BYY	24	27	37	21	19	36	34	28

(c) Time cost about comparative LFA on real world databases.

Fig. 2. Comparisons on real world data sets

The average classification accuracy of the 10 repetitions on the test sets are shown in Fig. 2(b), which indicates that the BYY harmony learning algorithm can automatically select not only the proper number of components but also the proper local dimension for each component to fit the data. The computing time are shown in Fig. 2(c), where we report the average time of the three criteria's implementations because they take very similar CPU time. All the above experiments were conducted via MATLAB 7.0.1(R14) on a P4 3.2GHz 512MB DRAM PC. We observe that ML learning by AIC, CAIC or BIC costs much more than those of IMoFA and BYY, because they require to compute a whole set of candidate models. BYY harmony learning is the most favorable one. For IMoFA, it has to compute several different choices' likelihood and judge functions in order to decide whether to add a component or to add a factor to one component step by step. This problem turns more serious when either the number of components or some local factor numbers are large.

5 BYY-LFA for Digits Recognition

Handwritten digits recognition is a convenient and important subproblem in optical character recognition (OCR) and has also been regarded as a typical test

[1] http://www.ics.uci.edu/ mlearn/MLRepository.html.
[2] http://www.cam-orl.co.uk/facedatabase.html.
[3] http://www.white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html.
[4] http://www.soe.ucsc.edu/research/compbio/genex/expressdata.html.
[5] http://www.cis.hut.fi/research/lvq_pak/.

case for pattern recognition theories. In this section, we further implement the BYY-LFA in comparison with IMoFA by the widely used handwritten digits database MNIST[6], in which there are 60,000 training images, i.e. 6,000 for each from "0" to "9", and 10,000 test images that are drawn from the same distribution as the training set. Shown in the first three rows of Fig. 3(a), where images are size-normalized and translated, with each represented by 28×28 pixels.

A LFA model is used for each digit from "0" to "9". To avoid local maxima, we still repeat IMoFA and BYY-LFA for 10 rounds, and then pick the best result for each approach. The fourth row of Fig. 3(a) describes several component means

(a) Digits illustration.

Approach	Test Error(%)
IMoFA	2.13
BYY-LFA Rank-1	1.18
BYY-LFA Rank-2	1.09
BYY-LFA Rank-3	0.89
Approach*	**Test Error(%)**
linear classifier (1-layer perceptron)	12.0
pairwise linear classifier	7.6
K-nearest-neighbors, Euclidean	5.0
1000 RBF + linear classifier	3.6
40 PCA + quadratic classifier	3.3
K-NN, Tangent Distance, 16×16 **	1.1
SVM deg 4 polynomial	1.1
Reduced Set SVM deg 5 polynomial	1.0
2-layer MLP, 1000 hidden units	4.5
3-layer MLP, 500+150 hidden units	2.95
LetNet-5, (no distortions)	0.95
Boosted LeNet-4, (distortions)	0.7
Conv CE, (affine distortions)	0.6
Conv CE, (elastic distortions)	0.4

* Performance data are collected from http://yann.lecun.com/exdb/mnist. All these results are noted to be regarded as reference for different methods by different designers.

** The system in that experiment used 16 by 16 pixel images.

(b) Comparing testing misclassification rates.

Fig. 3. Digits recognition via the BYY-LFA in MNIST database. The first three rows are some samples drawn from MNIST of "2". The 4th row describes component means for "2" by BYY-LFA. The last three rows include some misclassified digits.

of digit "2" obtained by the BYY-LFA learning. To observe clearly, we represent them into 8×8 grey-scale-reverse images.

Given a test image y_i, for IMoFA we calculate the likelihood $p(y_i|M_l)$ for each mixture M_l and determine $l^* = \arg max_l\ p(y_i|M_l), l = 0, \ldots, 9$. For classification by BYY-LFA, we implement BYY-LFA Rank-1, 2, and 3 as described at the end of Sec.3. Shown in Fig. 3(b) are comparisons with many known algorithms registered in the MNIST database. Here, one result of 0.8 by *V-SVM deg 9 poly (distortion)* is not included because it is referred in personal communication [12]

[6] Freely available at http://yann.lecun.com/exdb/mnist/.

and lacks further information. Some of the misclassified digits has been provided in the last three rows of Fig. 3(a).

From the experiments, we find that BYY-LFA outperforms IMoFA and achieves obviously better results than those models with a large number of parameters such as MLP and RBF. Compared with the currently best results, BYY-LFA is much favorable in training time and storage. BYY-LFA is trained only in around one hour in our PC described before, while the multi-layer net and convolutional net need much longer time, e.g., 2 weeks for LeNet-5 and a month for boosted LeNet-4 on a Sparc 10 machine [12]. Although there is no time cost available on the *Conv CE* methods, i.e., the methods providing the best two results in Fig.3(b)), it is expected to be comparable to that by LeNet because they share similar nature of the convolutional networks. Furthermore, compared with the multi-layer net and SVM method, BYY-LFA requires far less memory. Consequently, viewing from both the efficiency and the computational cost, BYY-LFA is more preferable, with the results comparable to the best ones.

6 Conclusion

A comparative study has been conducted on Bayesian Ying-Yang (BYY) harmony learning for local factor analysis (LFA) with automatic determination of both the component number and the local factors in each component, in comparison with ML-AIC, ML-CAIC, ML-BIC as well as a recent approach called Incremental Mixture of Factor Analyzers (IMoFA). A series of comparative experiments on simulated data sets, real world data sets, and the popular digits recognition database MNIST have shown that the BYY-LFA is the best in terms of both the performances and the computing time, while IMoFA and ML-BIC are better ML-AIC and ML-CAIC.

7 Further Discussion

We also find that, when the sample size is small, these discussed local factor analysis methods all face a risk of mis-selection, not only for the automatic methods including BYY harmony learning and IMoFA, but also for the typical criteria including AIC, CAIC and BIC. A better BYY model selection criterion considering the small-sample-size problems with the help of the two-phase implementation has been proposed in [16,17]. It has shown its advantages of producing much more accurate selection for small-sample-size cases compared to above discussed methods. However, to save space, the details are not covered in this paper.

Acknowledgement

The work described in this paper was fully supported by a grant from the Research Grant Council of the Hong Kong SAR (Project No: CUHK4173/06E).

References

1. Redner, R.A. and Walker, H.F.: Mixture densities, maximum likelihood, and the EM algorithm. SIAM Review, vol.26, pp.195-239, 1984.
2. Kambhatla, N. and Leen, T.K.: Fast non-linear dimension reduction. Advances in NIPS 6, Morgan Kaufmann, San Francisco, 1994.
3. Hinton G.E., Revow, M. and Dayan, P.: Recognizing handwritten digits using mixtures of Linear models. Advances in NIPS 7, MIT Press, Cambridge, MA, 1995.
4. Akaike, H.: A new look at statistical model identification. IEEE Trans. Automatic Control, vol.19, pp.716-723, 1974.
5. Barron, A. and Rissanen, J.: The minimum description length priciple in coding and modeling. IEEE Trans. Information Theory, vol.44, pp.2743-2760, 1998.
6. Bozdogan, H.: Model selection and Akaike's information criterion (AIC): the general theory and its analytical extensions. Psychometrika, vol.52(3), pp.345-370, 1987.
7. Figueiredo, M.A.T. and Jain, A.K.: Unsupervised learning of finite mixture models. IEEE Trans. Pattern Analysis and Machine Intelligence, vol.24(3), pp.381-396, 2002.
8. Rubin, D. and Thayer, D.: EM algorithms for ML factor analysis. Psychometrika, vol.47(1), pp.69-76, 1982.
9. Schwarz, G.: Estimating the dimension of a model. The Annals of Statistics, vol.6(2), pp.461-464, 1978.
10. Ghahramani Z. and Beal M.: Variational inference for Bayesian mixture of factor analysers. Advances in NIPS, vol.12, pp.449-455, 2000.
11. Albert Ali Salah and Ethem Alpaydin: Incremental Mixtures of Factor Analysers. Proc. 17th Intl Conf. on Pattern Recognition, vol.1, 276-279, 2004.
12. LeCun, Y. et al.: Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE, vol.86(11), pp.2278-2324, 1998.
13. Xu, L, : Multisets Modeling Learning: An Unified Theory for Supervised and Unsupervised Learning, Invited Talk, *Proc. IEEE ICNN94*, June 26-July 2, 1994, Orlando, Florida, Vol.I, pp.315-320.
14. Xu, L.: Temporal BYY Encoding, Markovian State Spaces, and Space Dimension Determination. IEEE Trans. Neural Networks, vol.15(5), pp.1276-1295, 2004.
15. Xu, L: Advances on BYY harmony learning: information theoretic perspective, generalized projection geometry, and independent factor auto-determination, *IEEE Trans on Neural Networks, Vol. 15, No. 4*, pp885-902, 2004.
16. Xu, L.: A Unified Perspective and New Results on RHT Computing, Mixture Based Learning, and Multi-learner Based Problem Solving. To appear in a special issue of Pattern Recognition, 2006.
17. Xu, L.: Trends on Regularization and Model Selection in Statistical Learning: A Perspective from Bayesian Ying Yang Learning. Challenges to Computational Intelligence (in press), Duch, W., Mandziuk, J. and Zurada, J.M. eds, the Springers series - Studies in Computational Intelligence, Springer-Verlag, 2006.

Interpolating Support Information Granules

B. Apolloni[1], S. Bassis[1], D. Malchiodi[1], and W. Pedrycz[2]

[1] Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano
Via Comelico 39/41, 20135 Milano, Italy
{apolloni, bassis, malchiodi}@dsi.unimi.it
[2] Department of Electrical and Computer Engineering, University of Alberta
ECERF, 9107 - 116 Street, Edmonton, Alberta, Canada T6G 2V4
pedrycz@ee.ualberta.ca

Abstract. We develop a hybrid strategy combing thruth-functionality, kernel, support vectors and regression to construct highly informative regression curves. The idea is to use statistical methods to form a confidence region for the line and then exploit the structure of the sample data falling in this region for identifying the most fitting curve. The fitness function is related to the fuzziness of the sampled points and is regarded as a natural extension of the statistical criterion ruling the identification of the confidence region within the Algorithmic Inference approach. Its optimization on a non-linear curve passes through kernel methods implemented via a smart variant of support vector machine techniques. The performance of the approach is demonstrated for three well-known benchmarks.

1 Introductory Comments

This work concerns the use of techniques of granular computing [1] in the refinement of standard regression models [2]. The underlying concept and the design rationale can be concisely outlined in the following manner (see Fig. 1). Given the experimental data, we commonly confine to the linear regression model as the first possible alternative worth exploring. Once accepted, we then focus on the refinement of the model. From the functional standpoint, there are several essential phases reflecting the rationale. First, the confidence region of the preliminary linear model (formed through the use of the confidence curves for some predefined confidence level) eliminates data points falling outside this region. The remaining data are subject to further usage in model building by endowing them with some properties of information granules. We consider the surroundings of those selected points as true information granules and equip them with bell-shaped membership functions similar to those encountered e.g. in radial basis functions (RBF) [3] (see Fig. 2(a)). In own turn, we connect the shape of the bells around points to the mutual relations between these points as it emerges from a suitable clustering of them. Considering the landscape constituted by a norm on the bells, we may look for a regression curve maximizing the integral of this norm along the curve (see Fig. 2(b)).

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 270–281, 2006.

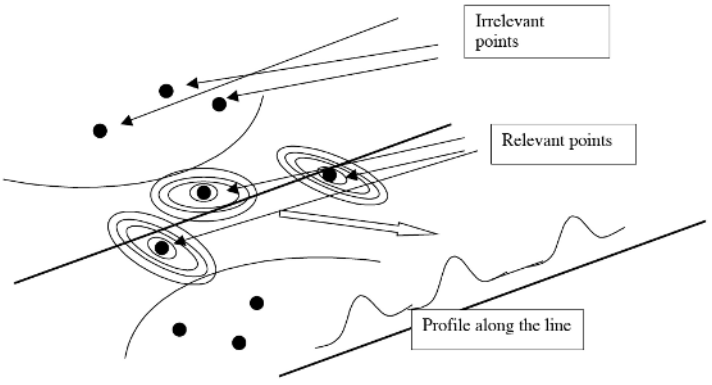


Fig. 1. A synopsis of the proposed method

We consider all this to form a dual objective in respect to support vector machines (SVM) [4]. With the latter we try to draw a line passing along the valleys, with the former along the crests of the fitness landscape. In both cases we manage more complex curves by making use of kernel techniques. The main idea is to exploit well established SVM techniques in order to develop an efficient solution. As it is, however, our dual objective has the drawback of not presenting a saddle point as identifier of the optimal solution. This makes harmless the SVM search for the null value of the duality gap. In this study we overcome this drawback by adopting a proper shift trick. Finally, we use the kernel mathematics to deal with non-linear curves as well.

Clear advantage of this procedure relies on the formation of a unifying processing framework that exploits both types of information, namely granular and statistical. This is particularly beneficial as these two are generally viewed to be mutually exclusive. In the literature, indeed we have a huge vein of works on statistical regression theory (refer to [5] and [6] as some representative examples). Also fuzzy regression has gained some visibility, where the drifts of the model with respect to the observed data come within the fuzziness with which the whole the data generation system (the coefficients of the regression line included) can be defined [7,8]. Both approaches start, however, from the general assumption of the existence of *the true* model, concealed to the humans apart some air-holes releasing sample observations alternatively framed into either an exact though indeterminate framework or into a context not susceptible of sharp computations. On the contrary our starting point is the sample data that we try to organize into operationally suitable descriptions, distinguishing between local information – in the fuzzy sets realm – and global information – in the realm of statistics – that are jointly owned by them. The benefit of the approach is the substantial easiness with which we may integrate many tools separately assessed in the single frameworks.

The paper is organized as follows: Section 2 describes how the regression model is determined, while Section 3 covers some preliminary numerical experiments.

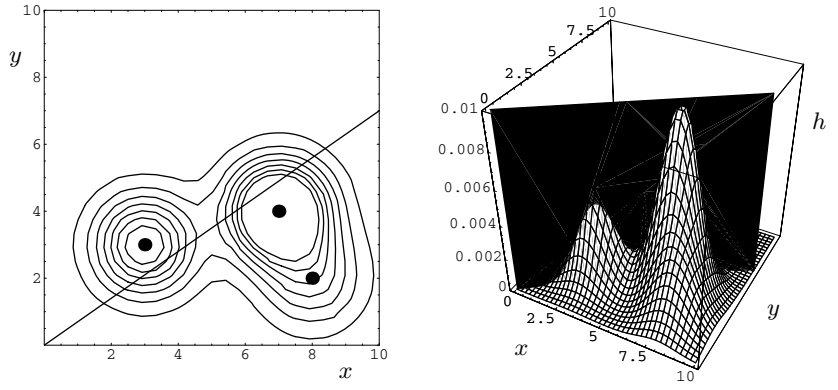


Fig. 2. Fitting the granules' information with a line. (a) Fitness contour lines; x independent variable, y dependent variable. (b) Crossing landscape with the regression line; h: fitness

Finally, we offer some conclusions and elaborate on future developments of the proposed approach.

2 The Design Method of the Model

Let us use as leading workbench the SMSA dataset [9] (see Fig. 3(a)) listing age adjusted mortality specifications (M) as a function of a demographic index (%NW, the non-white percentage). After reading it as a sample $\mathbf{z} = \{(x_i, y_i)\ i = 1, \ldots, n\}$, the proposed method works through a sequence of steps: i) identifying the information granules, ii) endowing each granule with a relevance measure, iii) determining a regression line on the basis of the data selected in i) and ii), iv) revisiting the linear granular regression problem in terms of solving a dual problem, v) moving to non-linear curves via kernel methods, and vi) reconsidering different bell heights, as detailed in the following subsections.

2.1 Identifying Information Granules

The first step is devoted to the selection, among the sample points, of the information granules upon which the rest of the procedure will be based. For the fixed value of δ, we identified these granules with the m sample points included in a $1 - \delta$ confidence region Ω for the regression line describing the relation among the sample points' coordinates. Namely, we explain the coordinates (x_i, y_i) of each sample point through

$$y_i = a + bx_i + \epsilon_i \tag{1}$$

with ϵ_i representing a random (for instance Gaussian) noise, and compute a confidence region where the regression line entirely falls with probability $1 - \delta$

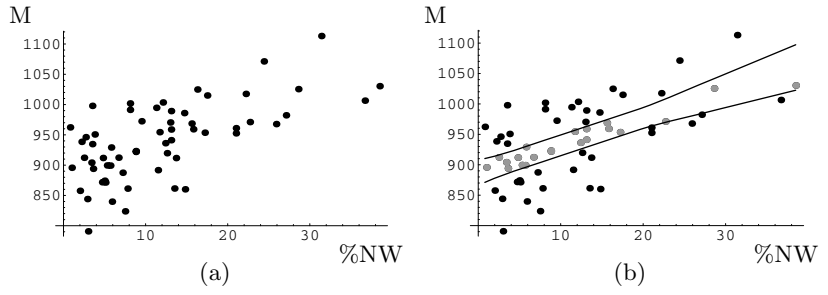


Fig. 3. (a) A sample of features extracted from the SMSA dataset where %NW refers to a demographic index (the percentage of non-white persons) and M to the age adjusted mortality; (b) 0.90 confidence regions for a regression line assuming a Gaussian noise. Gray points refer to the items in the SMSA dataset shown in (a) contained in the confidence region.

according to the Algorithmic Inference approach [10]. Hence, discarding points in $\mathbf{z}$ not belonging to Ω we obtain a *pruned version* $\mathbf{z}^* = \{(x_i^*, y_i^*)\ i = 1, \ldots, m\}$ of the sample. For instance, Fig. 3(b) illustrates the 0.90-confidence Ω corresponding to the sample in Fig. 3(a) as obtained through the Algorithmic Inference regression method [11], and the afterwards pruned sample.

2.2 Assigning Relevance to the Granules

The points in $\mathbf{z}^*$ constitute the statistically drawn base of knowledge, while the remaining ones are essentially assumed to be outliers. We also assume the former to be *information granules*, namely the centers of m fuzzy sets described by bell-shaped membership functions μ_i defined as follows:

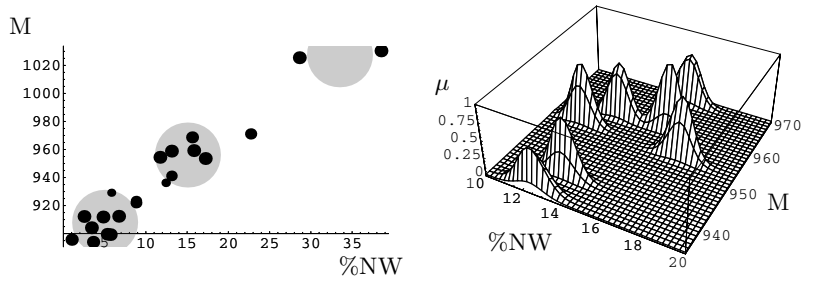
$$\mu_i(x, y) = h_i \mathrm{e}^{-\pi h_i((x-x_i^*)^2+(y-y_i^*)^2)}. \tag{2}$$

Each of these functions resembles a Gaussian symmetric bell centered around the point $\mathbf{z}_i^* = (x_i^*, y_i^*)$, i.e. a bidimensional normal density function, whose variates' coordinates have the same variance $\sigma^2 = (2\pi h_i)^{-1}$ and covariance $\rho = 0$.

Determining the set $\{h_i, i = 1, \ldots, m\}$ is the operational way of making the model definite. This corresponds to embedding in the i-th granule some information about its *relevance* h_i. Indeed, the higher this value, the smaller the variance of the corresponding density.

A possible way of determination of h_i would consider the topology of the pruned samples by some clustering mechanisms, say Fuzzy C-Means (FCM) [12]. Having fixed the number of clusters to be equal to c, once their centroids $\{\mathbf{v}_1, \ldots \mathbf{v}_c\}$ has been identified, we compute the relevance h_i as the maximal value of its membership grades to the various clusters, that is:

$$h_i = \max_{1 \le k \le c} \left\{ \left(\sum_{j=1}^{c} \left(\frac{||\mathbf{z}_i^* - \mathbf{v}_k||}{||\mathbf{z}_i^* - \mathbf{v}_j||} \right)^{\frac{2}{\alpha-1}} \right)^{-1} \right\}, \tag{3}$$



(a) (b)

Fig. 4. (a) Output of the Fuzzy C-Means procedure applied to the points shown in Fig. 3 starting with a number of centroids $c = 3$. Gray disks: 0.90 cuts. Black circles have a radius proportional to the relevance of the points. (b) Bells membership functions obtained by applying (2) to the points shown in (a) where the height of each bell depends on the relevance of the points.

where $\alpha \in \mathbb{N}$ is a fuzzification factor (> 1) whose original value has been selected when running the clustering procedure. The typical value of this factor is taken as 2.0.

For $c = 3$, Fig. 4(a) shows the output of such a procedure applied to our leading example together with the fuzzy centroids (gray disks), while Fig. 4(b) shows the bells membership functions corresponding to points located near the second centroid [1].

2.3 Finding the Optimal Regression Line

Among all the possible lines entirely contained in Ω, we will look now for the *optimal regression line*, i.e. the line r maximizing the sum of the integrals I_i^* of the curves obtained intersecting the membership functions $\mu_i^*(x, y)$ with the plane which contains r and in addition is orthogonal to the plane $X \times Y$ to which both r and the sample points belong. If we refer to the points of r through the equation $a + bx + y = 0$, i.e. r has slope and intercept respectively equal to $-b$ and $-a$, the above integral will depend on the latter quantities, thus we write $I_i^*(a, b)$.

In the plane having as axes r and any line orthogonal to it, say having coordinates ξ and ψ, given the radial symmetry of the bell membership function, we may express the latter again as a bidimensional Gaussian density function

$$\mu_i^*(\xi, \psi) = h_i \mathrm{e}^{-\pi h_i((\xi-\xi_i^*)^2+(\psi-\psi_i^*)^2)} \tag{4}$$

where ξ_i^* and ψ_i^* are the analogous of x_i^* and y_i^* in the new space.

Summing up, the integral $I_i^*(a, b)$ corresponding to the i-th granule is

$$I_i^*(a, b) = \int_{-\infty}^{\infty} \mu_i^*(\xi, \psi_i)\mathrm{d}\xi = \mu_i^*(\psi_i) \int_{-\infty}^{\infty} \mu_i^*(\xi|\psi_i)\mathrm{d}\xi = \mu_i^*(\psi_i), \tag{5}$$

[1] We focus on this subset of points to facilitate visualization.

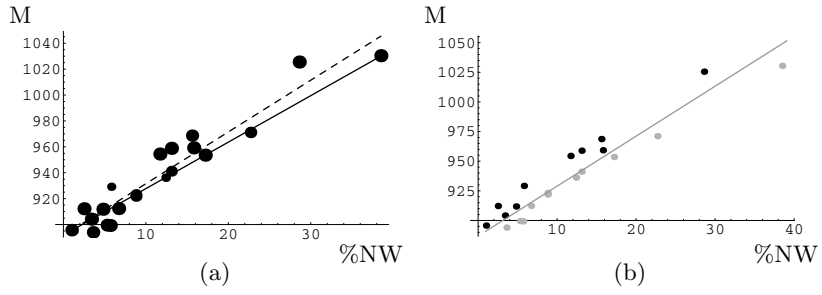


Fig. 5. Comparison between: (a) optimal regression line (9) (black line) and maximum likelihood estimator (MLE) line (dashed line), and (b) the line minimizing the farthest points from itself.

where $\mu_i^*(\xi|\psi_i)$ has the shape and mathematical properties of a conditional density function of a random variable Ξ given the value of the companion variable $\Psi = \psi_i$ and, analogously, $\mu_i^*(\psi_i)$ is the marginal distribution of Ψ evaluated on ψ_i. Hence

$$\mu_i^*(\psi_i) = h_i^{1/2} \mathrm{e}^{-\pi h_i \psi_i^2} \tag{6}$$

and $\int_{-\infty}^{\infty} \mu_i^*(\xi|\psi_i)\mathrm{d}\xi = 1$ by definition.

Finally, as ψ_i is the distance of the point (x_i^*, y_i^*) from r, we have

$$\psi_i = \frac{|bx_i^* + y_i^* + a|}{\sqrt{1+b^2}}, \tag{7}$$

so that the integral value is

$$I_i^*(a,b) = h_i^{1/2} \mathrm{e}^{-\pi h_i \frac{(bx_i^* + y_i^* + a)^2}{1+b^2}}. \tag{8}$$

Therefore, the optimal regression line has parameters

$$(a^*, b^*) = \arg\max_{a,b} \sum_{i=1}^{m} h_i^{1/2} \mathrm{e}^{-\pi h_i \frac{(bx_i^* + y_i^* + a)^2}{1+b^2}}. \tag{9}$$

In order to solve the related optimization problem, we can turn to an incremental algorithm, like a simple gradient descent or simulated annealing [13], exploiting the easy form of the derivatives of the integrals w.r.t. the parameters a and b of the regression line. The sole constraint we put is that the final line must not trespass the borders of the confidence region Ω. In our leading example, after some thousands iterations of the gradient descent algorithm we obtained the results shown in Fig. 5(a).

2.4 Identifying a Suitable Dual Problem

First of all, in order to draw on the SVM literature, we focus on the problem of minimizing the distances of the farthest points from the line. This is a definitely

relevant change in respect to the usual mean square minimization target used to identify a regression line. This change proves not so disrupting, however, in consideration of the fact that: i) on the one hand we identify a confidence region for the regression line on the basis of the regression lines' distribution law, and ii) within this region we are looking for a meaningful curve. Now, as we have assumed relevant all points in the δ-cut represented by the confidence region, preserving the influence of the farthest ones looks like a worth target to pursue. Rather, for expository reasons we will start with equal bells around each point, so that what counts is their topological distance from the regression line. We will remove this constraint later on.

In order to fit the standard SVM notation, let us move from the (x, y) reference framework to the (x_1, x_2); grouping these variables in the vector $\mathbf{x}$ the regression line equation can be written as $\mathbf{w}\cdot\mathbf{x}+b = 0$. With these specifications, the primal form of the problem is the following:

Definition 1 (Primal problem). *Given a set of points $S = \{\mathbf{x}_i, i = 1, \ldots, m\}$, maximize the norm of $\mathbf{w}$ under the constraint that all points have functional distance $|\mathbf{w} \cdot \mathbf{x}_i + b|$ less or equal to 1 from the line. In formulas*

$$\max_{\mathbf{w},b} \left\{ \frac{1}{2}||\mathbf{w}||^2 \quad \textit{such that } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \leq 1 \quad \forall i \right\} \tag{10}$$

where $y_i = \mathrm{Sign}(\mathbf{w} \cdot \mathbf{x}_i + b)$.

In terms of Lagrangian multipliers $\alpha_i \geq 0$, (10) reads:

$$\max_{\mathbf{w},b} \left\{ \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{m} \alpha_i \left(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1\right) \right\} \tag{11}$$

The drawback of this problem is that the function we want to minimize has not a saddle point in the space $\mathbf{w} \times \boldsymbol{\alpha}$. Hence, to fulfill this condition and work with a dual problem in $\boldsymbol{\alpha}$ we consider the equivalent problem:

Definition 2 (Dual problem). *For line, points and labels as in Definition 1 and a suitable instantiation of the line, map S into S' by translating under the line the points that are over it and vice versa, along a direction normal to the line by a fixed quantity that is sufficient to swap the positions w.r.t. the line of the farthest points. Then find solution to:*

$$\min_{\mathbf{w},b} \left\{ \frac{1}{2}||\mathbf{w}||^2 \quad \textit{such that } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad \forall i \right\} \tag{12}$$

i.e.

$$\min_{\mathbf{w},b} \left\{ \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{m} \alpha_i \left(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1\right) \right\}. \tag{13}$$

Of this problem we have the dual formulation

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j \text{ such that } \sum_{1=1}^{m} y_i \alpha_i = 0; \alpha_i \geq 0, \forall i \tag{14}$$

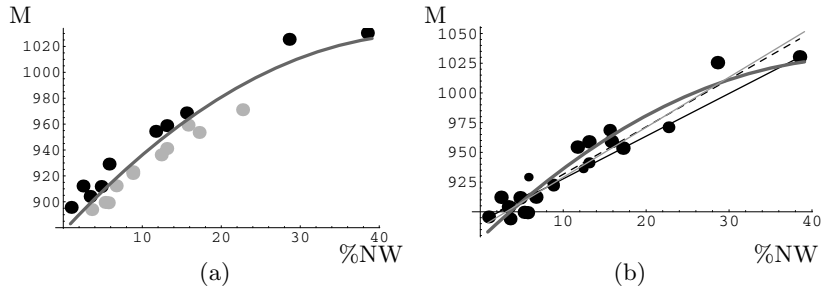


Fig. 6. (a) Optimal regression parabola (gray bold line), and (b) MLE (dashed line), optimal regression line (plain line), straight line minimizing the distance between the farthest points (gray line) and the optimal regression parabola (gray bold line) computed from the SMSA dataset

It is trivial to show that the procedure that computes y_i's, translates points according to the running line and update the latter on the basis of the above operations has a fixed point in the solution of problem in Definition 1. In particular, continuing our illustrative example we obtain the line in Fig. 5(b).

2.5 Toward More Complex Regression Curves

As can be seen in Figs. 5(b) and 6(b) the line computed on the basis of the support vectors lies in an opposite position than the optimal regression line w.r.t. the MLE curve, thus denoting some lack of information brought by the remaining points. This may suggest that SMSA dataset points could be better fitted through a parabola. A proper introduction of kernels allows us to solve the related regression problem in the same way as for linear curves. As it is well known, this boils down to the fact that the optimization object in (14) depends on the points only through the inner product $\mathbf{x}_i \cdot \mathbf{x}_j$. Assume it as a special issue of a symmetric function $k(\mathbf{x}_i, \mathbf{x}_j)$ – the kernel – and repeat the computation for any other issue of this function intended as the inner product $\mathbf{z}_i \cdot \mathbf{z}_j$ with $\mathbf{z}_i = \phi(\mathbf{x}_i)$ ranging in a suitable feature space and you obtain a fitting of the point according to a linear function on $\mathbf{z}$, hence a possibly non-linear function on $\mathbf{x}$.

The vector $\mathbf{z}$ has typically higher dimension than $\mathbf{x}$ (actually the additional components take into account the nonlinearities of the fitting function). We will come back to our leading example after having introduced the last point of the procedure.

2.6 Freeing the Shapes of the Granular Bells

In principle, having different bells around each sample point locates them at virtual distances that are different from the topological ones. As it emerges from (4), the more is relevant a point so farther it must be considered from the hyperplane

in the extended space. We may induce this virtual metric by simply multiplying the distance of a point $\mathbf{x}_i$ times its relevance h_i, i.e. by pushing or pulling consequently the points along the orthogonal direction to the hyperplane. The problem is that the virtual distances depend now not only on the versor but also on the position of the hyperplane (i.e. on the b coefficient). Thus we must find both parameters in the fixed point of the whole procedure, and this may require some dumping operator, such as exponential smoothing, to converge to a fixed point. This happens for instance in Fig. 6 with our example, where we subsituted the dot product in the last procedure with an ad hoc polynomial kernel computing the class of parabolas. Fig. 6(a) shows such a curve minimizing the distance (in the feature space) between the farthest points according to the relevance correction, while Fig. 6(b) summarizes the types of forms obtained so far.

3 Numerical Experiments

We ran a number of experimental studies making use of a number of well known benchmarks, from which we show the Swiss dataset [14], describing the dependence between fertility and socio-economic quantities in 47 provinces of Switzerland during 1888. In particular, Fig. 7(a) shows the 0.90 confidence region for the fertility percentage (F) as a function of the percentage of infant mortality (IM). Fig. 7(b) compares the optimal regression line (plain line) with the line minimizing the distance between the farthest points (gray line) both computed on the points lying in the confidence region and enriched with relevance information (obtained by applying the Fuzzy C-Means algorithm with again $c = 3$). Looking at the two lines, we recognize a different behavior on this dataset, coming from the different objective the two procedures aim to achieve. Finally, Fig. 8(a) compares the MLE parabola (dashed curve) with the one obtained through the optimization procedure keeping into account the additional relevance information (gray bold line).

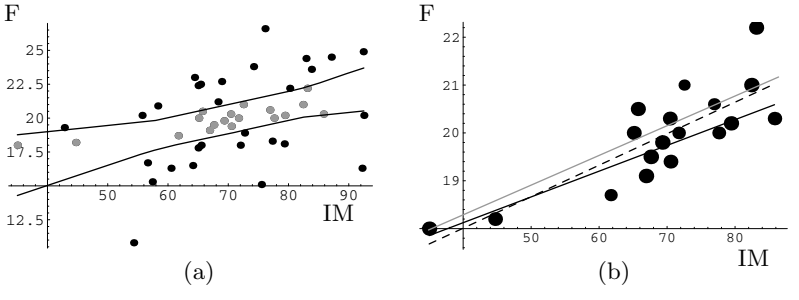
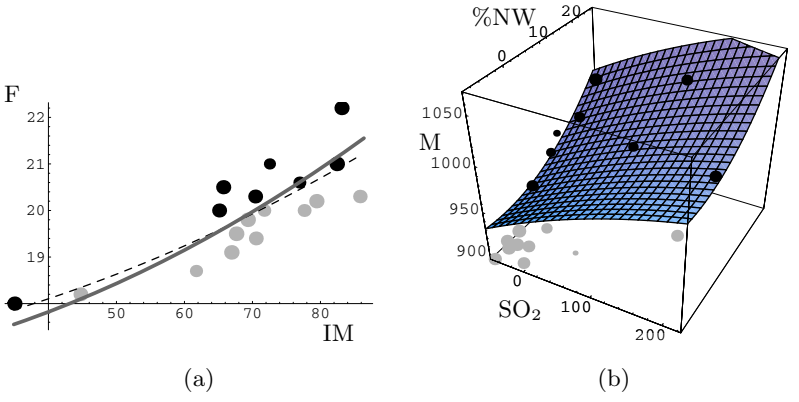
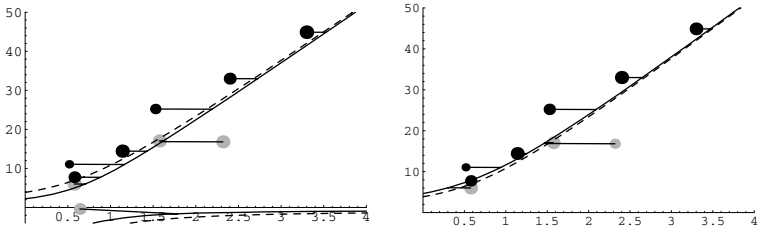


Fig. 7. (a) 0.90 confidence regions computed on the Swiss dataset. (b) Optimal regression line (plain line), line minimizing the farthest points (gray line) and MLE line (dashed line).



(a) (b)

Fig. 8. (a) Comparison between MLE parabola (dashed line) and the curve obtained considering the additional relevance information (gray bold curve). (b) 3D paraboloid fitting the points drawn from the SMSA dataset and enriched with relevance information obtained by applying the iterative procedure aiming at minimizing the distance between farthest points



(a) (b)

Fig. 9. Non-linear regression curves. Dashed curves: regression curves found by minimizing the distances of the farthest points from the curve using kernels; black curves: curves obtained by minimizing the weighted distance of points from curves. Points' size is proportional to membership function. (a) all the points, and (b) only those lying in the first quadrant are considered.

We also tried other kernel operators. For instance in Fig. 9 we used the typical polynomial kernel with degree 2 $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i, \mathbf{x}_j + c)^2$ for any constant c [2] on a specially featured dataset [15]. Note that the equidistance of the farthest points is obtained in the Z space, hence from the hyperplane fitting the points in this space. Thus, what we really exploit of this hyperplane is its versor (the angular coefficients) while the constant term must be renegotiated in the X space. At this point we are free to add more stringent requirements, for instance that the weighted sum of the quadratic distances from the fitting curve is minimized as in

[2] Actually a slight variant embedding also the shifts of points to pivot the curves on farthest points, as explained in Definition 2.

the original goal. In this way we obtain an approximate solution of the original problem in a reasonable time thanks to the use of kernels in the dual optimization problem. Note that, thanks to the quadratic shape of the membership functions the curve is very close to the one obtained with a b minimizing the distances of farthest points (gray curve) like in all previous examples. Moreover, assuming the point lying in the second quadrant as outlier and therefore deleting it, we obtain the different scenario depicted in Fig. 9(b).

The procedure can be applied with no further variation to multidimensional data points. For the sake of visualization we focus on a three-dimensional dataset, constituted by the above SMSA dataset, where the age adjusted mortality now depends on both the non-white percentage and the solfure dioxide concentration SO_2. Fig. 8(b) shows a paraboloid surface solving the dual problem (14).

4 Conclusions

In the perspective of probability as a way of organizing available information about a phenomenon rather than a *physical* property of the phenomenon, we consider additional information which is local, hence not gathered through a measure summing to 1 over a population. In particular respect to the linear regression problem, we focus on: i) $1 - \delta$-cuts identified through statistical methods, and ii) a local density of clusters of points that reverberates in a membership function of population points to the information granules represented by the sample points. In the perspective that still the representation of these informations has to be negotiated with the suitability of their exploitation, we used an augmented kernel trick to have the possibility of locating the information granules in the virtual space we feel most proper, and the dual formulation of the SVM problem to get results quickly. The proposed method is very general and the implementation is available at the url `http://laren.dsi.unimi.it/GranularRegression`. This could help move forward toward full exploitation of information available within data.

References

1. Pedrycz, W.: Granular computing in data mining. In Last, M., Kandel, A., eds.: Data Mining & Computational Intelligence. Springer-Verlag (2001)
2. Morrison, D.F.: Multivariate statistical methods. 2nd edn. McGraw-Hill, New York (1989)
3. Poggio, T., Girosi, F.: Networks for approximation and learning. In Lau, C., ed.: Foundations of Neural Networks. IEEE Press, Piscataway, NJ (1992) 91–106
4. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines. Cambridge University Press (2000)
5. Seber, G.A.F., Alan, L.J.: Linear Regression Analysis. Second edn. Hoboken Wiley-Interscience (2003)
6. Douglas, B.M., Watts, D.J.: Nonlinear regression analysis and its applications. John Wiley & Sons, New York (1988)

7. Tanaka, H., Uejima, S., Asai, K.: Linear regression analysis with fuzzy model. IEEE Transactions Systems, Man and Cybernetic (1982) 903–907
8. Savic, D., Pedrycz, W.: Evaluation of fuzzy regression models. Fuzzy Sets and Systems **39** (1991) 51–63
9. U.S. Department Labor Statistics: SMSA dataset. Air pollution and mortality, `http://lib.stat.cmu.edu/DASL/Datafiles/SMSA.html` (accessed January 2006)
10. Apolloni, B., Malchiodi, D., Gaito, S.: Algorithmic Inference in Machine Learning. Advanced Knowledge International, Magill (2003)
11. Apolloni, B., Bassis, S., Gaito, S., Iannizzi, D., Malchiodi, D.: Learning continuous functions through a new linear regression method. In Apolloni, B., Marinaro, M., Tagliaferri, R., eds.: Biological and Artificial Intelligence Environments, Springer (2005) 235–243
12. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algoritms. Plenum Press, New York (1981)
13. Aarts, E., Korst, J.: Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing. John Wiley, Chichester (1989)
14. Mosteller, F., Tukey, J.: Data Analysis and Regression: A Second Course in Statistics. Addison Wesley, Reading, Mass. (1977)
15. Apolloni, B., Iannizzi, D., Malchiodi, D., Pedrycz, W.: Granular regression. In Apolloni, B., Marinaro, M., Tagliaferri, R., eds.: Proceedings of WIRN 2005. LNCS, Springer (2005) in press.

Feature Selection Based on Kernel Discriminant Analysis

Masamichi Ashihara and Shigeo Abe

Graduate School of Science and Technology
Kobe University
Rokkodai, Nada, Kobe, Japan
abe@eedept.kobe-u.ac.jp
http://www2.eedept.kobe-u.ac.jp/~abe

Abstract. For two-class problems we propose two feature selection criteria based on kernel discriminant analysis. The first one is the objective function of kernel discriminant analysis (KDA) and the second one is the KDA-based exception ratio. We show that the objective function of KDA is monotonic for the deletion of features, which ensures stable feature selection. The KDA-based exception ratio defines the overlap between classes in the one-dimensional space obtained by KDA. The computer experiments show that the both criteria work well to select features but the former is more stable.

1 Introduction

Feature selection, i.e., deletion of irrelevant or redundant input variables from the given input variables, is one of the important steps in constructing a pattern classification system with high generalization ability [1,2]. And many selection methods for kernel-based methods have been proposed [2,3,4,5,6,7]. The margin [5,8,9] is often used for feature selection for support vector machines. Instead of the margin, in [7], block deletion of features in backward feature selection is proposed using the generalization ability by cross-validation as the selection criterion.

Feature selection has a long history of research and many methods have been developed. In [10], an exception ratio is defined based on the overlap of class regions approximated by hyperboxes. This exception ratio is monotonic for the deletion of input variables. By this monotonicity, we can terminate feature selection when the exception ratio exceeds a predefined value.

In this paper we propose two feature-selection criteria based on kernel discriminant analysis (KDA) for two-class problems. The first criterion uses the objective function of KDA. Namely the ratio of the between-class scatter and within-class scatter. We prove that this criterion is monotonic for the deletion of input variables. The second criterion is the exception ratio defined on the one-dimensional space generated by KDA according to [10].

The feature selection is done by backward selection. We start from all the input variables. We temporally delete one input variable, calculate the selection

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 282–291, 2006.

criterion, and delete the input variable that improves the selection criterion the most. This process is iterated until the stopping condition is satisfied.

In Section 2, we summarize KDA and in Section 3, we discuss two selection criteria and their monotonicity. In Section 4, we explain backward feature selection used and in Section 5 we demonstrate the validity of the proposed methods by computer experiments.

2 Kernel Discriminant Analysis

In this section we summarize kernel discriminant analysis, which finds the component that maximally separates two classes in the feature space [11,12], [13, pp. 457–468].

Let the sets of m-dimensional data belong to Class i $(i = 1, 2)$ be $\{\mathbf{x}_1^i, \ldots, \mathbf{x}_{M_i}^i\}$, where M_i is the number of data belonging to Class i, and data $\mathbf{x}$ be mapped into the l-dimensional feature space by the mapping function $\mathbf{g}(\mathbf{x})$. Now we find the l-dimensional vector $\mathbf{w}$, in which the two classes are separated maximally in the direction of $\mathbf{w}$ in the feature space.

The projection of $\mathbf{g}(\mathbf{x})$ on $\mathbf{w}$ is $\mathbf{w}^T\,\mathbf{g}(\mathbf{x})/\|\mathbf{w}\|$. We find such $\mathbf{w}$ that maximizes the difference of the centers, and minimizes the variances, of the projected data.

The square difference of the centers of the projected data, d^2, is

$$d^2 = (\mathbf{w}^T(\mathbf{c}_1 - \mathbf{c}_2))^2 = \mathbf{w}^T(\mathbf{c}_1 - \mathbf{c}_2)\,(\mathbf{c}_1 - \mathbf{c}_2)^T\mathbf{w}, \tag{1}$$

where $\mathbf{c}_i$ are the centers of class i data:

$$\mathbf{c}_i = \frac{1}{M_i}\sum_{j=1}^{M_i}\mathbf{g}(\mathbf{x}_j^i) \quad \text{for} \quad i = 1, 2. \tag{2}$$

We define

$$Q_B = (\mathbf{c}_1 - \mathbf{c}_2)\,(\mathbf{c}_1 - \mathbf{c}_2)^T \tag{3}$$

and call Q_B the *between-class scatter matrix*.

The variances of the projected data, s_i^2, are

$$s_i^2 = \mathbf{w}^T\,Q_i\,\mathbf{w} \qquad \text{for} \quad i = 1, 2, \tag{4}$$

where

$$Q_i = \frac{1}{M_i}\,(\mathbf{g}(\mathbf{x}_1^i), \ldots, \mathbf{g}(\mathbf{x}_{M_i}^i))\,(I_{M_i} - \mathbf{1}_{M_i})\begin{pmatrix}\mathbf{g}^T(\mathbf{x}_1^i)\\ \vdots \\ \mathbf{g}^T(\mathbf{x}_{M_i}^i)\end{pmatrix} \quad \text{for} \quad i = 1, 2. \tag{5}$$

Here, I_{M_i} is the $M_i \times M_i$ unit matrix and $\mathbf{1}_{M_i}$ is the $M_i \times M_i$ matrix with all elements being $1/M_i$. We define

$$Q_W = Q_1 + Q_2 \tag{6}$$

and call Q_W the *within-class scatter matrix*.

Now, we want to maximize

$$J(\mathbf{w}) = \frac{d^2}{s_1^2 + s_2^2} = \frac{\mathbf{w}^T Q_B \mathbf{w}}{\mathbf{w}^T Q_W \mathbf{w}}, \tag{7}$$

but since $\mathbf{w}$, Q_B, and Q_W are defined in the feature space, we need to use kernel tricks. Assume that a set of M' vectors $\{\mathbf{g}(\mathbf{y}_1), \ldots, \mathbf{g}(\mathbf{y}_{M'})\}$ spans the space generated by $\{\mathbf{g}(\mathbf{x}_1^1), \ldots, \mathbf{g}(\mathbf{x}_{M_1}^1), \mathbf{g}(\mathbf{x}_1^2), \ldots, \mathbf{g}(\mathbf{x}_{M_2}^2)\}$, where $\{\mathbf{y}_1, \ldots, \mathbf{y}_{M'}\} \subset \{\mathbf{x}_1^1, \ldots, \mathbf{x}_{M_1}^1, \mathbf{x}_1^2, \ldots, \mathbf{x}_{M_2}^2\}$ and $M' \leq M_1 + M_2$. Then $\mathbf{w}$ is expressed as

$$\mathbf{w} = (\mathbf{g}(\mathbf{y}_1), \ldots, \mathbf{g}(\mathbf{y}_{M'}))\, \boldsymbol{\alpha}, \tag{8}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{M'})^T$ and $\alpha_1, \ldots, \alpha_{M'}$ are scalars. Substituting (8) into (7), we obtain

$$J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^T K_B \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T K_W \boldsymbol{\alpha}}, \tag{9}$$

where

$$K_B = (\mathbf{k}_{B_1} - \mathbf{k}_{B_2})\, (\mathbf{k}_{B_1} - \mathbf{k}_{B_2})^T, \tag{10}$$

$$\mathbf{k}_{Bi} = \begin{pmatrix} \dfrac{1}{M_i} \displaystyle\sum_{j=1}^{M_i} H(\mathbf{y}_1, \mathbf{x}_j^i) \\ \cdots \\ \dfrac{1}{M_i} \displaystyle\sum_{j=1}^{M_i} H(\mathbf{y}_{M'}, \mathbf{x}_j^i) \end{pmatrix} \quad \text{for} \quad i = 1, 2, \tag{11}$$

$$K_W = K_{W_1} + K_{W_2}, \tag{12}$$

$$K_{W_i} = \frac{1}{M_i} \begin{pmatrix} H(\mathbf{y}_1, \mathbf{x}_1^i) \cdots H(\mathbf{y}_1, \mathbf{x}_{M_i}^i) \\ \cdots \\ H(\mathbf{y}_{M'}, \mathbf{x}_1^i) \cdots H(\mathbf{y}_{M'}, \mathbf{x}_{M_i}^i) \end{pmatrix} (I_{M_i} - \mathbf{1}_{M_i}) \times \begin{pmatrix} H(\mathbf{y}_1, \mathbf{x}_1^i) \cdots H(\mathbf{y}_1, \mathbf{x}_{M_i}^i) \\ \cdots \\ H(\mathbf{y}_{M'}, \mathbf{x}_1^i) \cdots H(\mathbf{y}_{M'}, \mathbf{x}_{M_i}^i) \end{pmatrix}^T \quad \text{for} \quad i = 1, 2. \tag{13}$$

Taking a partial derivative of (9) with respect to $\mathbf{w}$ and equating the resulting equation to zero, we obtain the following generalized eigenvalue problem:

$$K_B \boldsymbol{\alpha} = \lambda K_W \boldsymbol{\alpha}, \tag{14}$$

where λ is a generalized eigenvalue.

Substituting

$$K_W \boldsymbol{\alpha} = \mathbf{k}_{B_1} - \mathbf{k}_{B_2} \tag{15}$$

into the left-hand side of (14), we obtain

$$(\boldsymbol{\alpha}^T K_W \boldsymbol{\alpha})\, K_W \boldsymbol{\alpha}. \tag{16}$$

Thus, by letting $\lambda = \boldsymbol{\alpha}^T K_W \boldsymbol{\alpha}$, (15) is a solution of (14).

Since K_{W_1} and K_{W_2} are positive semi-definite, K_W is positive semi-definite. If K_W is positive definite, $\boldsymbol{\alpha}$ is given by

$$\boldsymbol{\alpha} = K_W^{-1} (\mathbf{k}_{B_1} - \mathbf{k}_{B_2}). \tag{17}$$

Even if we choose independent vectors $\mathbf{y}_1, \ldots, \mathbf{y}_{M'}$, for non-linear kernels, K_W may be positive semi-definite, i.e., singular. One way to overcome singularity is to add positive values to the diagonal elements [11]:

$$\boldsymbol{\alpha} = (K_W + \varepsilon I)^{-1} (\mathbf{k}_{B_1} - \mathbf{k}_{B_2}), \tag{18}$$

where ε is a small positive parameter.

3 Selection Criteria and Their Monotonicity

3.1 KDA Criterion

The first selection criterion is the value of (7) for optimum $\mathbf{w}$. We call this KDA criterion. The KDA criterion with linear kernels, i.e., the LDA criterion is often used for a feature selection criterion but its monotonicity for deletion of features is not known.

We can easily prove that the KDA criterion is monotonic for the deletion of input variables. Let $\mathbf{x}^i$ be the m-dimensional vector, in which the ith element of $\mathbf{x}$ is replaced with 0 and other elements are the same with those of $\mathbf{x}$. Then the resulting feature space $S^i = \{\mathbf{g}(\mathbf{x}^i) \,|\, \mathbf{x}^i \in R^m\}$ is the subspace of $S = \{\mathbf{g}(\mathbf{x}) \,|\, \mathbf{x} \in R^m\}$, where the feature space variables in S^i that include the ith element of $\mathbf{x}^i$ are zero for polynomial and RBF kernels.

Let the coefficient vectors obtained by KDA in S and S^i be $\mathbf{w}_{\mathrm{opt}}$ and $\mathbf{w}^i_{\mathrm{opt}}$, respectively. Then

$$J(\mathbf{w}_{\mathrm{opt}}) \geq J(\mathbf{w}^i_{\mathrm{opt}}) \tag{19}$$

is satisfied. This is proved as follows. Assume that the above relation does not hold. Namely, $J(\mathbf{w}_{\mathrm{opt}}) < J(\mathbf{w}^i_{\mathrm{opt}})$ is satisfied. Then $\mathbf{w}_{\mathrm{opt}}$ is not optimal in S since $\mathbf{w}^i_{\mathrm{opt}} \in S$.

Monotonicity of the selection criterion is very important because we can terminate the selection procedure by setting a threshold, or we can use optimization techniques such as branch and bound for feature selection.

3.2 KDA-Based Exception Ratio

In this section, we discuss the exception ratio defined in the one-dimensional space, $\mathbf{w}^T \mathbf{g}(\mathbf{x})/\|\mathbf{w}\|$, obtained by KDA, which is an extension of the exception ratio [10] defined in the input space. We call the space obtained by KDA *KDA space*. We define the class overlap by the overlap of class data in the KDA space. Namely, for class i ($i = 1, 2$), we define the activation regions with level 1, $A_{ii}(1)$, calculating the maximum $V_{ii}(1)$ and minimum $v_{ii}(1)$ of class i data in the KDA

space. If the activation regions $A_{11}(1)$ and $A_{22}(2)$ overlap we define the overlapping regions as the inhibition region $I_{12}(1)$ with the interval $[W_{12}(1), w_{12}(1)]$. If there are data in the inhibition region, we define the activation regions with level 2, $A_{12}(2)$ and $A_{21}(2)$. If there is an overlap between $A_{12}(2)$ and $A_{21}(2)$, we define the inhibition region $I_{12}(2)$. We repeat the above procedure until there are no data in the inhibition region.

The ratio of activation regions and inhibition regions indicates the difficulty of classification. Therefore, we define the exception ratio o_{ij} for classes i and j as the sum of the ratios of the activation and inhibition regions as follows:

$$o_{ij} = \sum_{l=1,\cdots,l_{ij}} p_{ij}(l) \frac{b_{I_{ij}}(l)}{b_{A_{ij'}}(l)}, \tag{20}$$

where $j' = i$ for $l = 1$, $j' = j$ for $l \geq 2$,

$$b_{I_{ij}} = \begin{cases} W_{ij}(l) - w_{ij}(l) & \text{for } W_{ij}(l) - w_{ij}(l) > \varepsilon, \\ \varepsilon & \text{otherwise,} \end{cases}$$

$$b_{A_{ij'}} = \begin{cases} V_{ij'}(l) - v_{ij'}(l) & \text{for } V_{ij'}(l) - v_{ij'}(l) > \varepsilon, \\ \varepsilon & \text{otherwise,} \end{cases}$$

$$p_{ij}(l) = \frac{\text{number of class } i \text{ training data in } I_{ij}(l)}{\text{total number of training data}}.$$

Here, ε is a small positive parameter. If there is no data in the inhibition region, the region does not affect separability of classes. Thus, in (20), we add $p_{ij}(l)$ to reflect this fact. We call the exception ratio given by (20) *KDA-based exception ratio.*

The exception ratio is zero if there is no overlap between classes. Thus, by this criterion, separability is considered to be the same even if the margins between classes are different. The exception ratio defined in the input space is monotonic for the deletion of input features [10], but unfortunately the KDA-based exception ratio is not monotonic as the computer experiments discussed later show.

4 Backward Feature Selection

We select features using backward feature selection. In the backward feature selection, first we calculate the value of the selection criterion using all the features. Then starting from the initial set of features we temporally delete each feature, calculate the value of the selection criterion, and delete the feature with the highest value of the selection criterion from the set. We iterate feature deletion so long as class separability is higher than the prescribed level.

Let the initial set of selected features be F^m, where m is the number of input variables, and the value of the selection criterion be T^m. We delete the ith $(i = 1, \ldots, m)$ feature temporally from F^m and calculate the selection criterion. Let the selection criterion be T_i^m. We iterate this procedure for all i

$(i = 1, \ldots, m)$. Then we delete the feature $\arg\max_{i \in F^m} T_i^m$ from F^m: $F^{m-1} = F^m - \{\arg\max_{i \in F^m} T_i^m\}$, if $T_i^m / T^m > \delta_{\mathrm{KDA}}$ or $T_i^m / T^m < \delta_{\mathrm{EXT}}$, where the first inequality is for the KDA criterion, the second inequality is for the KDA-based exception ratio, and δ_{KDA} and δ_{EXT} are thresholds for the KDA criterion and KDA-based exception ratio, respectively.

We iterate the above feature selection procedure so long as the above inequality is satisfied.

5 Performance Evaluation

We evaluated performance of the selection criteria using the two-class problems listed in Table 1 [11].[1] Each problem has 100 or 20 training and test data sets.

For the features selected by backward feature selection, we trained the L1 support vector machines, scaling the input range into $[0, 1]$, calculated the means and standard deviations of the recognition rates, and statistically analyzed the results with the significance level of 0.05. We used an AthlonMP2000+ personal computer running on Linux.

Table 1. Two-class benchmark data sets

Data	Inputs	Train.	Test	Sets
B. cancer	9	200	77	100
Diabetes	8	468	300	100
German	20	700	300	100
Heart	13	170	100	100
Image	18	1300	1010	20
Ringnorm	20	400	7000	100
F. solar	9	666	400	100
Thyroid	5	140	75	100
Titanic	3	150	2051	100
Twonorm	20	400	7000	100
Waveform	21	400	4600	100

Table 2. Parameter setting

Data	Kernel	ε	η
B. cancer	$\gamma 10$	10^{-8}	10^{-8}
Diabetes	$\gamma 10$	10^{-8}	10^{-6}
German	$\gamma 10$	10^{-8}	10^{-8}
Heart	$\gamma 10$	10^{-8}	10^{-8}
Image	$\gamma 10$	10^{-8}	10^{-8}
Ringnorm	$\gamma 10$	10^{-8}	10^{-4}
F. solar	$\gamma 10$	10^{-8}	10^{-7}
Thyroid	$\gamma 10$	10^{-8}	10^{-8}
Titanic	$\gamma 10$	10^{-8}	10^{-4}
Twonorm	$\gamma 10$	10^{-8}	10^{-6}
Waveform	$\gamma 10$	10^{-8}	10^{-3}

We selected the kernel and its parameter, from among polynomial kernels with $d = [2, 3, 4]$ and RBF kernels with $\gamma = [0.1, 1, 10]$, so that the maximum value of the objective function of KDA [14] is realized. We selected the value of ε, which is used to avoid matrix singularity in KDA and the threshold value of Cholesky factorization, η, from among $\varepsilon = [10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}]$, $\eta = [10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}]$ so that the KDA criterion is maximized as follows:

1. Calculate the KDA criterion, using all the features, for the first five training data sets. Thus we obtain 5 values of the objective function.
2. Select the values of ε and η that correspond to the maximum value of the KDA criterion.

[1] http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm

Table 2 lists the parameter values obtained by the above procedure. For all the problems, RBF kernels with $\gamma = 10\,(\gamma = 10)$ were selected.

In evaluating the selected features by the support vector machine, we determined the kernel and parameter values by 5-fold cross-validation; for the original set of features, we used the same kernel types and parameter ranges as those for KDA and determined the value of the margin parameter C from $C = [1, 10, 50, 100, 500, 1000, 2000, 3000, 5000, 8000, 10000, 50000, 100000]$. For the selected feature set, we used the same kernel and kernel parameter as those for the initial set of features and determined the value of C by cross-validation.

Since each problem consists of 100 or 20 data sets, we combined the first 5 training data sets into one and selected features by backward feature selection for the two selection criteria with $\delta_{\mathrm{KDA}} = 0.5$ and $\delta_{\mathrm{EXT}} = 1.5$.

Figures 1 and 2 show the recognition rates of the thyroid data set when features were deleted using the KDA criterion and KDA-based exception ratio criterion, respectively. The horizontal axis shows the deleted features at each selection step and the vertical axis shows the recognition rates of the training data set in the right and test data sets in the left for each selection step. The vertical axis also shows the value of the selection criterion with the initial value normalized to 1.

In Fig. 1, the selection criterion is monotonic for the deletion of features. Since $\delta_{\mathrm{KDA}} = 0.5$, three features: 4th, 3rd, and 1st features were deleted and 2nd and 5th features were left. In Fig. 2 the deletion sequence of features is the same with that by the KDA criterion. But the selected features are different. From the figure, for $\delta_{\mathrm{EXT}} = 1.5$ only the 4th feature was deleted compared with three features by the KDA criterion. Since the exception ratio decreased when the fourth feature was deleted, the exception ratio was not monotonic.

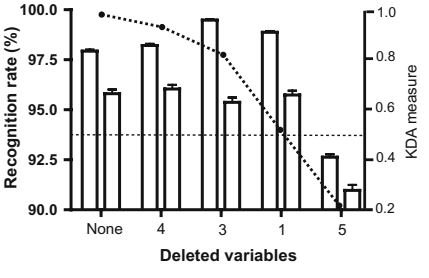


Fig. 1. Feature deletion for the thyroid data set by KDA criterion

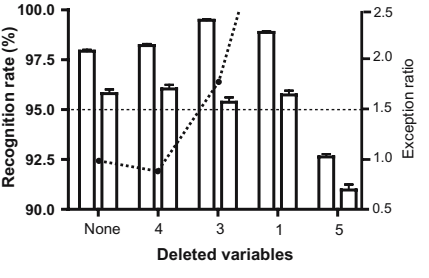


Fig. 2. Feature deletion for the thyroid data set by exception ratio

Tables 3 and 4 show the feature selection results using the KDA and KDA-based exception ratio criteria, respectively. In the tables, the "Deleted" column lists the features deleted. If for a classification problem two feature strings are shown, the numbers of features selected by the two criteria are different. The second feature string shows the features that are deleted after the first feature string is deleted. And the asterisk shows that the number of features deleted

Table 3. Recognition performance for feature selection using the KDA criterion

Data	Deleted	Parm	C	Train.	Test	KDA	#F
B. cancer	None	γ0.1	500	*77.57*±**1.87**	72.36±4.67	12.3	4
	5,9		2000	**78.68±1.83**	72.94±4.56	8.3	
	2,7*		100	74.56±4.04	72.77±5.39	3.7	
Diabetes	None	*d*3	100	**78.95**±1.27	76.42±1.79	3.31	5
	4,3*		50	78.44±1.05	76.67±1.76	2.50	
	5,1		100	78.35±1.11	**77.00**±1.67	1.78	
German	None	γ0.1	50	*77.80*±1.03	76.19±2.27	676	9
	4,20,16,5,18,15,10,17		500	**78.71**±0.90	75.82±2.14	359	
	19*		100	76.99±1.00	75.77±2.17	137	
Heart	None	γ0.1	50	**85.96**±1.91	83.69±3.41	1081	5
	6,11,9		100	**86.15**±1.93	83.76±3.52	694	
	4,1*		100	85.17±2.07	83.43±3.53	65	
Image	None	γ10	1000	98.60±0.17	97.13±0.47	18.9	6
	8,6,12,9,10,3		2000	**99.28±0.09**	97.37±0.37	22.2	
Ringnorm	None	γ10	10	**99.51±0.33**	**97.67**±0.33	27.6	0
	18*		10	**99.38±0.35**	*97.41*±0.37	25.8	
	20, 15, 11, 5, 17, 14		10	98.33±0.54	95.50±0.39	13.9	
F. solar	None	*d*2	10	67.50±1.05	67.61±1.72	0.730	1
	9,6,8,3,7,2,1		100000	67.46±1.09	67.67±1.81	0.436	
Thyroid	None	γ10	10	97.93±0.78	95.80±2.09	26.1	3
	4*		10	*98.21*±0.82	96.04±2.08	25.2	
	3,1		8000	**98.87±0.64**	95.75±2.16	14.2	
Titanic	None	*d*3	100	**79.49**±3.66	77.47±1.43	0.839	2
	2,1		100000	78.09±3.60	77.57±**0.26**	0.542	
Twonorm	None	*d*3	10	**98.09±0.59**	**97.59±0.12**	42.7	0
	18,7*		10	*97.62*±0.71	*96.95*±**0.14**	35.3	
	12,5,2		50	96.86±0.82	95.67±0.19	23.8	
Waveform	None	γ10	1	**93.53**±1.36	**90.00**±0.44	22.8	1
	3,16*		1	**93.18**±1.28	*89.77*±0.45	19.1	
	6,15,19,8		1	91.63±1.43	88.41±0.39	12.5	

is the same with that deleted by the other criterion not used in deleting the features. For example, in Table 3 for b. cancer the 5th and the 9th features are deleted using the KDA criterion and since four features are deleted by the KDA-based exception ratio criteria as shown in Table 4, we delete two more features: the 2nd and 7th. The best average recognition rate and standard deviation are shown in boldface and the second best italic. If there is no statistical difference they are shown in Roman.

In Table 3, "Parm" and "C" columns list the kernels and the values of C selected by 5-fold cross validation. For example, γ0.1 means the RBF kernels with $\gamma = 0.1$ and d3 means the polynomial kernels with degree 3. (In Table 4, the "Parm" column is not included because it is the same with that in Table 3.) The "Train." and "Test" columns list the average recognition rates with the standard deviations. The "KDA (EXT)" column lists the values of the selection criterion. The "#F" column lists the number of features that are successively

Table 4. Recognition performance for feature selection using the exception ratio

Data	Deleted	C	Train.	Test	EXT	#F
B. cancer	None	500	77.57±1.87	**72.36**±4.67	0.288	1
	1,9*	100000	**82.73**±1.92	70.55±4.73	0.170	
	4,2	100000	*78.45*±1.69	**72.72**±4.73	0.358	
Diabetes	None	100	**78.95**±1.27	76.42±1.79	14.6	5
	5,3	10	77.51±1.04	76.10±1.83	19.4	
	1,6*	100	77.59±1.26	75.90±1.82	27.2	
German	None	50	*77.80*±1.03	**76.19**±2.27	0	8
	4,20,16,5,18,15,10,17*	500	**78.71**±0.90	**75.82**±2.14	0	
	2	500	76.98±1.11	73.91±2.21	0	
Heart	None	50	**85.96**±1.91	83.69±3.41	0	6
	6,11,9*	100	**86.15**±1.93	83.76±3.52	0	
	4,1	100	85.17±2.07	83.43±3.53	0	
Image	None	1000	98.60±0.17	97.13±0.47	1.43	6
	3,10,6,8,9,14	2000	**99.23±0.13**	**97.40±0.37**	0.79	
Ringnorm	None	10	**99.51±0.33**	**97.67±0.33**	0.0993	0
	17	10	**99.40±0.34**	*97.52*±0.32	0.131	
	20,5,6,12,2,8*	50	99.25±0.42	94.80±0.38	1.26	
F. solar	None	10	**67.50±1.05**	**67.61±1.72**	2.97	0
	4,7,2,1,6,5,3*	50000	49.69±6.65	48.76±6.64	0.0158	
Thyroid	None	10	97.93±0.78	95.80±2.09	0.000257	3
	4	10	*98.21*± 0.82	96.04±2.08	0.000218	
	3,1*	8000	**98.87±0.64**	95.75±2.16	0.113	
Titanic	None	100	**79.49±3.66**	**77.47±1.43**	0.894	1
	1,3*	100000	46.6±30.0	45.92±29.5	0.0520	
Twonorm	None	10	**98.09±0.59**	**97.59±0.12**	0.00805	0
	8,3	50	**97.96±0.65**	*96.91*±0.17	0.0536	
	10,4,2*	50	96.65±0.89	95.46±0.19	0.154	
Waveform	None	1	**93.53**±1.36	**90.00**±0.44	0.264	0
	9,7	1	*92.88*±1.27	*89.41*±0.39	0.301	
	6,21,4,8*	1	91.72±1.29	88.54±0.41	1.28	

deleted without deteriorating the generalization ability in each deletion step. For example, in Table 3, according to the KDA criterion the four features are deleted for diabetes but by statistical analysis, additional one feature can be deleted.

From Table 3, except for the image data, the KDA criterion is monotonic for the deletion of features. For the image data, because of the memory overflow, we could not delete more than 6 features. Except for the ringnorm, twonorm, and waveform data sets, the selected features by the KDA criterion show comparable performance for the test data with the original features.

In Table 4, for german and heart data sets, since the exception ratio was 0, we deleted the features using KDA criterion until the exception ratio became non-zero. The exception ratio was monotonic for b. cancer, diabetes, ringnorm, twonorm, and waveform. For f. solar and titanic data sets, since the exception ratio monotonically decreased for the deletion of features, we could not stop the deletion procedure. The selected features by the exception ratio show comparable

performance for the test data with the original features for b. cancer, diabetes, heart, image, and thyroid data sets. The "#F" for the KDA criterion is in most cases better than that for the exception ratio. And the feature selection is more stable.

6 Conclusions

In this paper, we proposed two measures for feature selection: the KDA criterion which is the objective function of KDA and the KDA-based exception ratio, which defines the overlap of classes in the one-dimensional space obtained by KDA. We show that the KDA criterion is monotonic for the deletion of features. According to the computer experiments for two-class problems, we showed that both criteria work well to selects features but the KDA criterion was more stable.

References

1. S. Abe. *Support Vector Machines for Pattern Classification*. Springer, 2005.
2. I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1–3):389–422, 2002.
3. P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. *Proc. ICML '98*, 82–90, 1998.
4. J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping. Use of the zero-norm with linear models and kernel methods. *J. Machine Learning Research*, 3:1439–1461, 2003.
5. S. Perkins, K. Lacker, and J. Theiler. Grafting: Fast, incremental feature selection by gradient descent in function space. *J. Machine Learning Research*, 3:1333–1356, 2003.
6. Y. Liu and Y. F. Zheng. FS_SFS: A novel feature selection method for support vector machines. *Pattern Recognition* (to appear).
7. S. Abe. Modified backward feature selection by cross validation. In *Proc. ESANN 2005*, 163–168, 2005.
8. J. Bi, K. P. Bennett, M. Embrechts, C. Breneman, and M. Song. Dimensionality reduction via sparse support vector machines. *J. Machine Learning Research*, 3:1229–1243, 2003.
9. A. Rakotomamonjy. Variable selection using SVM-based criteria. *J. Machine Learning Research*, 3:1357–1370, 2003.
10. R. Thawonmas and S. Abe. A novel approach to feature selection based on analysis of class regions. *IEEE Trans. SMC–B*, 27(2):196–207, 1997.
11. S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. *NNSP 99*, 41–48, 1999.
12. G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, 2000.
13. B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.
14. S. Kita, S. Maekawa, S. Ozawa, and S. Abe. Boosting kernel discriminant analysis with adaptive kernel selection. In *Proc. ICANCA 05*, CD-ROM, 2005.

Local Selection of Model Parameters in Probability Density Function Estimation

Ezequiel López-Rubio[1], Juan Miguel Ortiz-de-Lazcano-Lobato[1], Domingo López-Rodríguez[2], Enrique Mérida-Casermeiro[2], and María del Carmen Vargas-González[1]

[1] Department of Computer Science and Artificial Intelligence, University of Málaga, Campus Teatinos, s/n, 29071 Málaga, Spain
{ezeqlr, jmortiz}@lcc.uma.es
http://www.lcc.uma.es

[2] Department of Applied Mathematics, University of Málaga, Campus Teatinos, s/n, 29071 Málaga, Spain
{dlopez, merida}@ctima.uma.es
http://www.satd.uma.es/matap

Abstract. Here we present a novel probability density estimation model. The classical Parzen window approach builds a spherical Gaussian density around every input sample. Our proposal selects a Gaussian specifically tuned for each sample, with an automated estimation of the local intrinsic dimensionality of the embedded manifold and the local noise variance. This leads to outperform other proposals where local parameter selection is not allowed, like the manifold Parzen windows.

1 Introduction

The estimation of the unknown probability density function (PDF) of a continuous distribution from a set of data points forming a representative sample drawn from the underlying density is a problem of fundamental importance to all aspects of machine learning and pattern recognition (see [1], [2] and [3]).

Parametric approaches make assumptions about the unknown distribution. They consider, a priori, a particular functional form for the PDF and reduce the problem to the estimation of the required functional parameters. On the other hand, nonparametric methods make less rigid assumptions. Thus they are more flexible and they usually provide better results. Popular nonparametric methods include the histogram, kernel estimation, nearest neighbor methods and restricted maximum likelihood methods, as can be found in [4], [5], [6] and [7].

The kernel density estimator, also commonly referred as the Parzen window estimator, [9], places a Gaussian kernel on each data point of the training set. Then, the PDF is approximated by summing all the kernels, which are multiplied by a normalizing factor. Thus, this model can be viewed as a finite mixture model (see [8]) where the number of mixture components will equal the number of points in the data sample. The parameter which defines the shape of those components, i.e. the covariance of the Gaussian kernel, is the same for all of them and the estimation of the arbitrary distribution is, therefore, penalized because of the poor adaptation to local structures of the

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 292 – 301, 2006.

data. Besides, most of the time, Parzen windows estimates are built using a "spherical Gaussian" with a single scalar variance parameter σ^2, which spreads the density mass equally along all input space directions and gives too much probability to irrelevant regions of space and too little along the principal directions of variance of the distribution. This drawback is partially solved in Manifold Parzen Windows algorithm, [10], where a different covariance matrix is calculated for each component. On the other hand, this model considers that the true density mass of the dataset is concentrated in a non-linear lower dimensional manifold embedded in the higher dimensional input space. In this sense, only information about directions of the lower dimensional manifold will be preserved in order to reduce the memory cost of the model. There is also a unique regularization parameter which is used to represent the variance in the discarded directions of the components, as it will be explained more detailed in section 2.

We present, in section 3, a model that selects automatically the adequate values for some parameters of the Manifold Parzen Windows model. Our method chooses the right dimensionality of the manifold according to a quality criterion specified by the user, which is the percentage of neighbourhood variance we want to be retained in each component. In a similar way, the regularization variance parameter will be selected by the method itself without the aid of human knowledge. Therefore the time invested in tuning the parameters to obtain good density estimations will be diminished. We show some experimental results, in section 4, where the selection achieved by our method produces more precise estimations that the Manifold Parzen Windows one.

2 The Manifold Parzen Windows Method

Let X be an n-dimensional random variable and $p_X()$ an arbitrary probability density function over X which is unknown and we want to estimate. The training set of the algorithm is formed by l samples of the random variable and the density estimator has the form of a mixture of Gaussians, whose covariances C_i may be identical or not:

$$\hat{p}_{mp}(x) = \frac{1}{l}\sum_{i=1}^{l} N_{x_i,C_i}(x) \,. \tag{1}$$

with $N_{\mu,C}(x)$ the multivariate Gaussian density:

$$N_{\mu,C}(x) = \frac{1}{\sqrt{(2\pi)^n |C|}} e^{-\frac{1}{2}(x-\mu)'C^{-1}(x-\mu)} \tag{2}$$

where μ is the mean vector, C is the covariance matrix and $|C|$ the determinant of C.

The density mass is expected to concentrate close to an underlying non-linear lower dimensional manifold and, thus, the Gaussians would be "pancakes" aligned with the plane locally tangent to that manifold. Without prior knowledge about the distribution $p_X()$ the information about the tangent plane is provided by the samples of the training set. Thus the principal directions of the samples in the neighbourhood of each sample x_i will be computed. The local knowledge about the principal directions will be obtained when we calculate the weighted covariance matrix C_{κ_i} for each sample:

$$C_{\kappa_i} = \frac{\sum_{j=1..l,\, j\neq i} \kappa(x_j; x_i)(x_j - x_i)'(x_j - x_i)}{\sum_{j=1..l,\, j\neq i} \kappa(x_j; x_i)} \tag{3}$$

where $(x_j\text{-}x_i)'$ $(x_j\text{-}x_i)$ denotes the outer product and $K(x, x_i)$ is a neighbourhood kernel centered in x_i which will associate an influence weight to any point x in the vicinity of x_i.

Vincent and Bengio propose in [10] the utilization of a hard k-neighbourhood which assigns a weight of 1 to any point no further than the k-th nearest neighbour of the sample x_i among the training set, according to some metric such as the Euclidean distance in input space, and setting the weight to 0 to those points further than the k-neighbour. This approach usually involves C_{κ_i} to be ill-conditioned so it is slightly modified by adding a small isotropic Gaussian noise of variance σ^2

$$C_i = C_{\kappa_i} + \sigma^2 I \tag{4}$$

When we deal with high dimensional training datasets it would be prohibitive in computation time and storage to keep and use each full covariance matrix C_i. Therefore, a compacted representation of them is preserved, storing only the eigenvectors associated with the first d largest eigenvalues of them, where d is chosen by the user of the algorithm and is fixed for each covariance matrix. The eigenvectors related to the largest eigenvalues of the covariance matrix correspond to the principal directions of the local neighbourhood, i.e. the high variance local directions of the supposed underlying d-dimensional manifold. The last few eigenvalues and eigenvectors are but noise directions with a small variance and a same low noise level, which is also the same σ^2 it was used before, is employed for them.

Once the model has been trained any sample of the distribution may be tested. The probability density estimation for the sample will be computed by the average of the probability density provided by the l local Gaussians as was mentioned in (**1**).

3 Dynamic Parameter Selection in Manifold Parzen Windows Algorithm

We extend the training of Vincent and Bengio's method [10], by providing a more automatic way to estimate density functions.

First we incorporate the capacity of estimating the intrinsic dimensionality, i.e. the needed number of principal directions d, of the underlying manifold for each neighbourhood. The cause is that we use a qualitative parameter, α, which represents the explained variance by the principal directions of the local manifold. Then, the method will be able to choose by itself the minimum number of eigenvectors which retain a particular amount of the variance presented in the vicinity of each training sample. This method has been employed in [11] and [12] with good results.

A second level of automated adaptation to the data will be added by means of a parameter γ. This parameter will enable the method to select the right noise level for discarded directions. So, a better adaptation of the model to the unknown distribution will be achieved.

3.1 The Explained Variance Method

The explained variance method considers a variable number, D_i, of eigenvalues and their corresponding eigenvectors to be kept which is computed independently for each sample x_i. This number reflects the intrinsic dimensionality of the lower dimensional manifold where the data lies for the neighbourhood of x_i. Through the training process the method ensures that a minimum amount of variance is conserved in order to satisfy the level of accuracy, $\alpha \in [0,1]$, chosen by the user. The number of principal directions which are preserved is set consequently to the minimum value which allows us to reach that level at least.

The most precise estimation of the data in the neighbourhood of a sample can be achieved if we conserve the full covariance matrix, i. e. we keep information about every direction, because it will be more likely to discover the right dimensionality of the underlying manifold. On the other hand, the worst estimation will be obtained when all the directions are ignored and the sample x_i is the only statistical information which is kept, i.e. when we lose all the variance relative to the directions of the embedded manifold. Thus, the lost variance when no directions are kept, V_0, can be defined as:

$$V_0 = \sum_{p=1}^{D} \lambda_i^p \tag{5}$$

with λ_i^p the p eigenvalue of the covariance matrix C_{κ_i}, which are supposed to be sorted in decreasing order, and D the dimension of the training samples.

In any other situation the discarded variance, V_Z, depends on the number, Z, of principal directions conserved:

$$V_Z = \sum_{p=Z+1}^{D} \lambda_i^p \tag{6}$$

Our goal is to obtain the most compressed representation of the covariance C_{κ_i}, while the model maintains a minimum level of quality α. with respect to the maximum accuracy the method can achieve. Let $V_0 - V_Z$ be the amount of error (we must remember that the more variance is lost the less precise the estimation will be) eliminated when we conserve information about the Z principal directions. Then

$$D_i = \min\left\{Z \in \{0,1,...,D\} \mid V_0 - V_Z \geq \alpha \;\; V_0\right\} \tag{7}$$

Substitution of (**5**) and (**6**) into (**7**) yields

$$D_i = \min\left\{Z \in \{0,1,...,D\} \mid \sum_{p=1}^{D} \lambda_i^p - \sum_{p=Z+1}^{D} \lambda_i^p \geq \alpha \;\; \sum_{p=1}^{D} \lambda_i^p\right\} \tag{8}$$

It is well known that the sum of variances of a dataset equals the trace of the covariance matrix for this dataset, therefore equation **(8)** can be simplified as follows:

$$D_i = \min\left\{Z \in \{0,1,...,D\} \mid \sum_{p=1}^{Z} \lambda_i^p \geq \alpha \;\; trace(C_{\kappa_i})\right\} \tag{9}$$

The quotient between λ_i^p and $trace(C_{\kappa_i})$ is the amount of variance explained by the pth principal direction of the estimated manifold. Thus, if we sum these quotients for all the retained directions, we can see the parameter α as the amount of variance which we want to be retained in each neighbourhood. Hence, we select D_i so that the amount of variance explained by the directions associated to the D_i largest eigenvalues is at least α.

3.2 The Qualitative Parameter γ

With the variance explained parameter our aim is to add the model the ability to adapt by itself to the local properties of the distribution. Thus, it saves memory space which is not required, i. e. only the necessary information of the covariance of each neighbourhood will be stored.

The same idea was applied to deal with the parameter σ^2, which controls the width of the Gaussians in Manifold Parzen Windows method. In order to take into consideration the local structure and to obtain better estimators, the noise variance for each neighbourhood is determined by

$$\sigma_i^2 = \gamma \cdot \lambda_i^{D_i} \tag{10}$$

where $\gamma \in [0,1]$ and $\lambda_i^{D_i}$ is the last of the preserved eigenvalues, i.e. the smallest of the first D_i largest eigenvalues.

As can be noticed there is a close relation between α and γ. If we use a value for α near to 0 then we likely retain only the first eigenvalue, which is associated to the first principal direction of the data. Therefore, it encompasses a great percentage of the total variance of the distribution. This means that $\lambda_i^{D_i}$ will be large and the noise variance will be set to a relatively large value. This implies that the Gaussian for the i neighbourhood will be widened along the discarded directions. In the opposite case, if a value near to 1 is assigned to α, then we store nearly all the eigenvalues and eigenvectors of the covariance matrix. The last retained eigenvalue will be very small and independently of the value of γ the noise variance will be set to a value near 0. This is in consonance with the fact that if we conserve all the information about the directions of change then there is not noise variance, because there is not any discarded dimension. In subsection 4.2, we present some plots where the fact just commented can be observed.

3.3 Parzen Manifold Windows with Qualitative Parameters

The proposed algorithm is designed to estimate an unknown density distribution $p_X()$ which the l samples of the training dataset are generated from. The generated estimator will be formed by a mixture of l Gaussians, one for each sample. Their shapes are adapted to the adequate local structure of the neighbourhoods through the training process and rely on the user specified qualitative parameters. The user chooses both the quality of the estimation, expressed by the explained variance parameter α; and γ, which means the width of the Gaussians in the discarded directions relative to the width in the last conserved direction.

The training method can be summarized as follows:

1. Take the training sample x_i with $i \in \{1,2,...,l\}$. Initially the first sample x_1 is selected.
2. Compute the covariance matrix C_{κ_i} following (**3**) where only the k nearest neighbours x_j of x_i are considered.
3. Extract the eigenvalues and eigenvectors from C_{κ_i} and estimate the dimensionality of the underlying manifold D_i, by means of (**9**)
4. Use (**10**) to calculate σ_i^2, the noise variance for the discarded directions.
5. Store the local model, i. e., the first D_i eigenvectors and eigenvalues, the local noise level σ_i^2, the l samples and the number of neighbours k.
6. Go to step 1, and continue the training process for the next sample. If there are not more samples to process, the algorithm finishes.

4 Experimental Results

This section shows some experiments we have designed in order to compare quality of density estimation presented by our method, we term MparzenQuality throughout this whole section and by the Vincent and Bengio's one, which will be referred as MParzen. For this purpose the measure used was the average negative log likelihood

$$ANLL = -\frac{1}{m}\sum_{i=1}^{m} \log \hat{p}(x_i) \tag{11}$$

where $\hat{p}(x)$ is the estimator, and the training dataset is formed by m examples x_i.

4.1 Experiment on 2D Artificial Data

A training set of 300 points, a validation set of 300 points and a test set of 10000 points were generated from the following distribution of two dimensional *(x,y)* points:

$$x = 0.04t\sin(t) + \varepsilon_x, \quad y = 0.04t\cos(t) + \varepsilon_y$$

where $t \sim U(3,15), \varepsilon_x \sim N(0,0.01), \varepsilon_y \sim N(0,0.01), U(a,b)$ is uniform in the interval *(a,b)* and $N(\mu,\sigma)$ is a normal density.

We trained a MParzenQuality model with explained variance 0.1 and 0.9 on the training set. The parameters k and γ were tuned to achieve the best performance on the validation test. On the other hand, MParzen with $d = 1$ and $d = 2$ was trained and the rest of its parameters were also tuned.

Quantitative comparative results of the two models are reported en Table 1, where it can be seen that our model outperforms the previous one in density distribution estimation. Figure 1 shows the results obtained when we applied the models on the test set. Darker areas represent zones with high density mass and lighter ones indicates the estimator has detected a low density area.

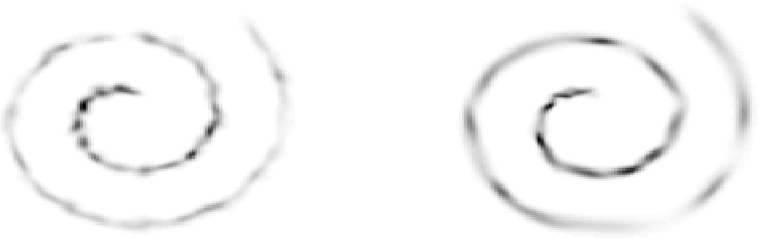


Fig. 1. Density estimation for the 2D artificial dataset, MParzen model (left) and MParzenQuality (right)

We can see in the plots that our model has less density holes (light areas) and less 'bumpiness'. This means that our model represents more accurately the true distribution, which has no holes and is completely smooth. We can see that the quantitative ANLL results agree with the plots. So, our model outperforms clearly the MParzen approach.

Table 1. Comparative results on the espiral dataset

Algorithm	Parameters used	ANLL on test-set
MParzen	$d = 1$, $k = 11$, $\sigma^2 = 0.009$	-1.466
MParzen	$d = 2$, $k = 10$, $\sigma^2 = 0.00001$	-1.419
MParzenQuality	$\alpha = 0.1$, $k = 10$, $\gamma = 0.1$	-2.204
MParzenQuality	$\alpha = 0.9$, $k = 10$, $\gamma = 0.1$	-2.116

4.2 Density Estimation on Astronomical Data

The dataset comes from the VizieR service [13], which is an information system for astronomical data. In particular, we have selected the Table 6 of the Complete near-infrared and optical photometric CDFS Catalog from Las Campanas Infrared Survey [14]. We have extracted 22 numerical features from 10,000 stars. Hence, we have 10,000 sample vectors. These data have been normalized in order to cope with the variability and the very heterogeneous scaling of the original data. This dataset has been split randomly in a training set (10% of the dataset), validation set (10%) and test set (the remaining 80%).

We have carried out simulation runs for MParzen with the number of dimensions retained from 1 to 6. For each of those values we have tried the following noise levels: σ^2 = 0.09, 0.1, 0.11, 0.13, 0.15, 0.17, 0.19, 0.3 and 0.5 (values near 0.11, which generates good performance). The simulations with the MParzenQuality model have been carried out with the following parameter values: α = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9; γ = 0.09, 0.11, 0.13, 0.15, 0.17, 0.19, 0.1, 0.2, 0.3, 0.4 and 0.5. In both models we have tried the following numbers of neighbours: 10, 15 and 20.

In Figure 2 the ANLL of the models is plotted versus the number of retained principal directions. For each value of d or α, only the best performing combination of the

rest of the parameters is shown in the plot. Please note that for the MParzenQuality model the average of the principal directions retained is averaged over all the samples, so fractional values of dimensionality are shown. It can be observed that our proposal is clearly superior in all conditions.

It should also be noted that with the MParzen model we have detected serious problems with the outliers. The original VizieR dataset is fairly uniform, but there are 3 outliers. These data samples caused the MParzen model to completely fail the ANLL performance test, because the model assigned a zero probability to these samples, up to double precision calculations, yielding a plus infinite ANLL. Our MParzenQuality model did not suffer from this problem, showing a better probability density allocation. These outliers have been removed in order to perform the tests corresponding to Figure 2.

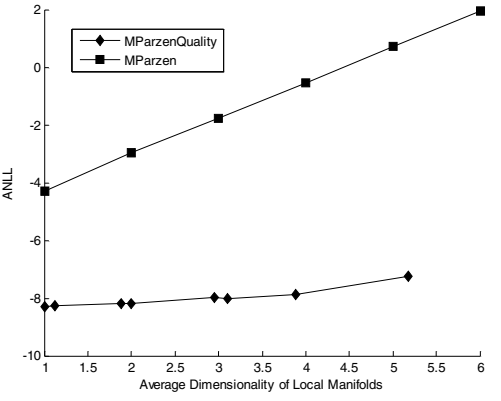


Fig. 2. Results with the VizieR astronomical dataset

A set of curves which represents the contribution of the qualitative parameters when we employ 15 neighbours for each data sample is presented in Figure 3.

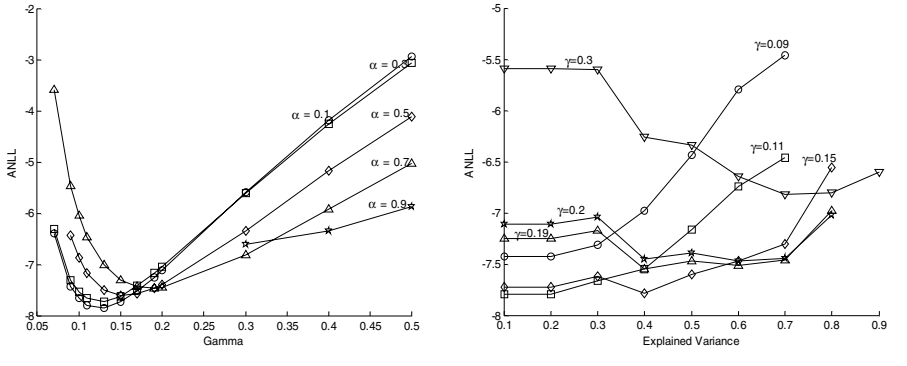


Fig. 3. Relationship of the qualitative parameters and the quality of the results

Similar conclusions may be extracted for both plots. First when the explained variance is fixed to a small percentage, then smaller values for parameter γ produces more adequate width for the "pancakes" and, thus, better results (see the minimum values for the curves of the left plot). On the other hand, if parameter α is greater than 0.5 then the last preserved eigenvalue is small, and the width of the Gaussians will be too narrow if the value assigned to γ is not chosen high enough. A compromise value γ is 0.2, which maintains an average performance, although it does not achieve the best results.

5 Conclusions

We have presented a probability density estimation model. It is based in the Parzen window approach. Our proposal builds a local Gaussian density by selecting independently for each training sample the best number of retained dimensions and the best estimation of noise variance. This allows our method to represent input distributions more faithfully than the manifold Parzen window model, which is an improvement of the original Parzen window method. Computational results show the superior performance of our method, and its robustness against outliers in the test set.

References

1. Bishop, C., Neural Networks for Pattern Recognition, Oxford University Press (1995)
2. Silverman, B., Density Estimation for Statistics and Data Analysis, Chapman and Hall, New York (1986)
3. Vapnik, V. N., Statistical Learning Theory, John Wiley & Sons, New York (1998)
4. Izenman, A. J., Recent developments in nonparametric density estimation. Journal of the American Statistical Association, 86(413) (1991) 205-224
5. Lejeune, M., Sarda, P., Smooth estimators of distribution and density functions. Computational Statistics & Data Analysis, 14 (1992) 457-471.
6. Hjort, N.L., Jones, M.C., Locally Parametric Nonparametric Density Estimation, Annals of Statistics, 24, 4 (1996) 1619-1647
7. Hastie, T., Loader, C., Local regression: Automatic kernel carpentry, Statistical Science, 8 (1993) 120-143
8. McLachlan, G., Peel, D., Finite Mixture Models, Wiley, 2000)
9. Parzen, E., On the Estimation of a Probability Density Function and Mode, Annals of Mathematical Statistics, 33 (1962) 1065-1076
10. Vincent, P., Bengio, Y., Manifold Parzen Windows, Advances in Neural Information Processing Systems, 15 (2003) 825-832
11. López-Rubio, E., Ortiz-de-Lazcano-Lobato, J. M., Vargas-González, M. C., López-Rubio, J. M., Dynamic Selection of Model Parameters in Principal Components Analysis Neural Networks, Proceedings fo the 16th European Conference on Artificial Intelligence (ECAI 2004) 618-622
12. López-Rubio, E., Ortiz-de-Lazcano-Lobato, J. M., Vargas-González, M. C., Competitive Networks of Probabilistic Principal Components Analysis Neurons, 9th IASTED International Conference on Artificial Intelligence and Soft Computing (ASC 2005) 141-146

13. VizieR service [online], Available at: http://vizier.cfa.harvard.edu/viz-bin/VizieR/ (March 29, 2004)
14. Chen, H.-W., et al., Early-type galaxy progenitors beyond z=1, *Astrophysical Journal, 560*, 2001, L131

The Sphere-Concatenate Method for Gaussian Process Canonical Correlation Analysis

Pei Ling Lai[1], Gayle Leen[2], and Colin Fyfe[2]

[1] Southern Taiwan Institute of Technology, Taiwan
pllai@mail.stut.edu.tw
[2] Applied Computational Intelligence Research Unit,
The University of Paisley, Scotland
{gayle.leen, colin.fyfe}@paisley.ac.uk

Abstract. We have recently developed several ways of using Gaussian Processes to perform Canonical Correlation Analysis. We review several of these methods, introduce a new way to perform Canonical Correlation Analysis with Gaussian Processes which involves sphering each data stream separately with probabilistic principal component analysis (PCA), concatenating the sphered data and re-performing probabilistic PCA. We also investigate the effect of sparsifying this last method. We perform a comparative study of these methods.

1 Introduction

A stochastic process $Y(\mathbf{x})$ is a collection of random variables indexed by $\mathbf{x} \in X$ such that values at any finite subset of X form a consistent distribution. A Gaussian Process (GP) therefore is a stochastic process on a function space which is totally specified by its mean and covariance function [11,8,6].

We have recently investigated several ways of using GPs to perform Canonical Correlation Analysis (CCA). Canonical Correlation Analysis is used when we have two data sets which we believe have some underlying correlation. Consider two sets of input data, $\mathbf{x}_1 \in X_1$ and $\mathbf{x}_2 \in X_2$. Then in classical CCA, we attempt to find the linear combination of the variables which gives us maximum correlation between the combinations. Let $y_1 = \mathbf{w}_1^T \mathbf{x}_1$ and $y_2 = \mathbf{w}_2^T \mathbf{x}_2$. Then, for the first canonical correlaton, we find those values of $\mathbf{w}_1$ and $\mathbf{w}_2$ which maximises $E(y_1 y_2)$ under the constraint that $E(y_1^2) = E(y_2^2) = 1$.

2 The Semi-parametric Method

Consider a stochastic process which defines a distribution, $P(f)$, over functions, f, where f maps some input space, χ to $\Re$. If e.g. $\chi = \Re$, f is infinite dimensional but the $\mathbf{x}$ values index the function, $f(\mathbf{x})$, at a countable number of points and so we use the data at these points to determine $P(f)$ in function space. If $P(f)$ is Gaussian for every finite subset of X, the process is a GP and is then determined by a mean function $\theta(\mathbf{x})$ and covariance function $\Sigma(\mathbf{x})$. These are often defined

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 302–310, 2006.

by hyperparameters, expressing our prior beliefs on the nature of θ and Σ, whose values are learned from the data.

A commonly used covariance function is $\Sigma : \Sigma_{ij} = \sigma_y^2 \exp(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{2l^2}) + \sigma_n^2 \delta_{ij}$ which enforces smoothing via the l parameter. The σ_y parameter determines the magnitude of the covariances and σ_n enables the model to explain the data, $y = f(\mathbf{x}) + n$, with $n \sim N(0, \sigma_n^2)$.

2.1 GP for Canonical Correlation Analysis

In [2], we used a GP to perform CCA in the following manner. Let the input data be $\mathbf{x}_1$ and $\mathbf{x}_2$. Then we define two sets of parameters for the Gaussian Process: let $\theta_i(\mathbf{x}_i), i = 1, 2$, define the mean function of the estimate for CCA and let $\Sigma_i, i = 1, 2$, be the corresponding covariance function. For example, in our first, expository example, we let $\mathbf{x}_1$ and $\mathbf{x}_2$ have a linear relationship so that $\theta_i(\mathbf{x}_i) = b_i \mathbf{x}_i + c_i, i = 1, 2$, with b_i, c_i being the parameters of the process, and $\Sigma^i_{kj} = \sigma^2_{i,y} \exp(-\frac{\|\mathbf{x}_{1,k} - \mathbf{x}_{1,j}\|^2 + \|\mathbf{x}_{2,k} - \mathbf{x}_{2,j}\|^2}{2l_i^2}) + \sigma^2_{i,n} I_N, k, j = 1, ..., N, i = 1, 2$ where N is the number of samples, $\mathbf{x}_{1,j}$ (resp. $\mathbf{x}_{2,j}$) is the j^{th} sample from the first (resp. second) data stream and l_i determines the degree of interaction between the samples. Note that we have continued to index the data stream by i so that $\Sigma^1 \neq \Sigma^2$ since $l_1, \sigma_{1,y}, \sigma_{1,n}$ may evolve differently from $l_2, \sigma_{2,y}, \sigma_{2,n}$.

Then we wish to maximise the covariance in function space of $(\theta_1(\mathbf{x}_1) - \mu_1)(\theta_2(\mathbf{x}_2) - \mu_2)$ under the constraint that $E(\theta_1(\mathbf{x}_1) - \mu_1)^2 = E(\theta_2(\mathbf{x}_2) - \mu_2)^2 = 1$. This is done by maximising the likelihood of the model given the two data sets: we assume that the current estimate of $\theta_2(\mathbf{x}_2)$ is the target for training $\theta_1(\mathbf{x}_1)$ but is corrupted by Gaussian noise. This gives a log likelihood function of

$$\begin{aligned} L &= \log p(\theta_1(\mathbf{x}_1) | \mathbf{x}_1, \theta_2) \\ &= -\frac{1}{2} \log |\Sigma_1| - \frac{1}{2} (\theta_2(\mathbf{x}_2) - \theta_1(\mathbf{x}_1))^T (\Sigma^1)^{-1} (\theta_2(\mathbf{x}_2) - \theta_1(\mathbf{x}_1)) - \frac{n}{2} \log(2\pi) \end{aligned}$$

Let γ_i be a generic parameter of the covariance matrix, Σ^i. Then we use the standard method of gradient descent on the log likelihood with $\theta_2(.)$ as the target for training $\theta_1(.)$,

$$\frac{\partial L}{\partial b_1} = (\theta_2(\mathbf{x}_2) - \theta_1(\mathbf{x}_1))^T (\Sigma^1)^{-1} \mathbf{x}_1; \qquad \frac{\partial L}{\partial c_1} = (\theta_2(\mathbf{x}_2) - \theta_1(\mathbf{x}_1))^T (\Sigma^1)^{-1} \quad (1)$$

$$\begin{aligned} \frac{\partial L}{\partial \gamma_1} &= -0.5 trace((\Sigma^1)^{-1} \frac{\partial \Sigma^1}{\partial \gamma_1}) \\ &\quad + 0.5 (\theta_2(\mathbf{x}_2) - \theta_1(\mathbf{x}_1))^T (\Sigma^1)^{-1} \frac{\partial \Sigma^1}{\partial \gamma_1} (\Sigma^1)^{-1} (\theta_2(\mathbf{x}_2) - \theta_1(\mathbf{x}_1)) \end{aligned} \quad (2)$$

where

$$\frac{\partial \Sigma_1}{\partial l_1} = 2\Sigma_1 T^1 / (2l_1^2); \quad \frac{\partial \Sigma_1}{\partial \sigma_{1,y}} = 2\sigma_{1,y} \exp(-T^1/(2l_1^2)); \quad \frac{\partial \Sigma_1}{\partial \sigma_{1,n}} = 2\sigma_{1,n} I_N \quad (3)$$

where $T^1_{kj} = \frac{\|\mathbf{x}_{1,k} - \mathbf{x}_{1,j}\|^2 + \|\mathbf{x}_{2,k} - \mathbf{x}_{2,j}\|^2}{2l_1^2}$. Thus we are using the current estimates given by $\theta_2(\mathbf{x}_2)$ as targets for the training of the mean and covariance functions

for the estimated functions on $\mathbf{x}_1$. We alternate this training with the equivalent rules for for the estimated functions on $\mathbf{x}_2$ when $\theta_1(\mathbf{x}_1)$ becomes the target. We can view the covariance matrix as the local product of the covariance matrices of X_i, thus creating a covariance matrix[6] for the product space $X_1 \times X_2$. An alternative would be to use the sum of the individual covariances.

We must also heed the constraint that $E(\theta_1(\mathbf{x}_1) - \mu_1)^2 = E(\theta_2(\mathbf{x}_2) - \mu_2)^2 = 1$ during training and so we scale the parameters of $\theta_i()$ after each update to satisfy this constraint.

In [2], we showed that this method works very well on artificial data. However, crucially the artificial data had only a single correlation in *one dimension* of each of the two data streams. Subsequently, [5], we showed that this method is not as accurate as our previous artificial neural networks methods [4,3] on real data sets. We find that the fields which exhibit the greatest correlations between the two data sets are identified accurately but the other fields are not identified with any accuracy.

In Table 1, we show converged weights with this method on a standard data set from [7]; this data set is composed of 88 students' exam marks in 5 exams. The marks of two exams in which books were allowed form one data set and the marks on the other 3 exams in which books were not allowed form the other data set. We see that the fields with the largest correlation in each data set is identified with the above method (the line 'Gaussian Process CCA - one l') but the fields with smaller contributions to the correlation are not accurately identified. It might be thought that this would be remedied by using a diagonal covariance matrix so that

$$\Sigma^i_{kj} = \sigma^2_{i,y} \exp(-\frac{1}{2}\{(\mathbf{x}_{1,k} - \mathbf{x}_{1,j})^T M_1 (\mathbf{x}_{1,k} - \mathbf{x}_{1,j}) + (\mathbf{x}_{2,k} - \mathbf{x}_{2,j})^T M_2 (\mathbf{x}_{2,k} - \mathbf{x}_{2,j})\}) + \sigma^2_{i,n} I_N, k, j = 1, ..., N, i = 1, 2 \quad (4)$$

where both M_1 and M_2 are diagonal matrices of width parameters which are separately updated. We see from Table 1 however, ('GP CCA with diag. cov.') that little improvement is achieved.

Remarkably, however, we can actually get a very good approximation to the CCA filters by simply modeling the noise with this method i.e. with $\Sigma^i_{kj} = \sigma^2_{i,n} I_N, k, j = 1, ..., N, i = 1, 2$. We show results with our standard data set (denoted 'GP CCA - noise cov.') in Table 1. Clearly we can use this approximation to initialise our parameters if we wish to use GP CCA on a large data set.

3 The Sphere-Concatenate Method

Now it may be shown [7] that a method of finding the canonical correlation directions is to solve the generalised eigenvalue problem

$$\begin{bmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = \rho \begin{bmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} \quad (5)$$

where Σ_{ij} is the covariance matrix between the i^{th} and j^{th} data streams. Note that this is equivalent to the standard eigenproblem

$$\begin{bmatrix} 0 & \Sigma_{11}^{-\frac{1}{2}}\Sigma_{12}\Sigma_{22}^{-\frac{1}{2}} \\ \Sigma_{22}^{-\frac{1}{2}}\Sigma_{21}\Sigma_{11}^{-\frac{1}{2}} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} = \rho \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} \tag{6}$$

where

$$\mathbf{v}_1 = \Sigma_{11}^{\frac{1}{2}}\mathbf{w}_1$$
$$\mathbf{v}_2 = \Sigma_{22}^{\frac{1}{2}}\mathbf{w}_2$$

This standard eigenproblem can now be seen to be equivalent to a decomposition of a cross-covariance matrix of sphered data.

This suggests the following algorithm:

1. Use Probabilistic PCA[1] on both data streams, independently. This gives us eigenvector matrices, $\mathbf{V}_1, \mathbf{V}_2$ and eigenvalues on the main diagonal of Λ_1, Λ_2.
2. Project each data stream onto their respective eigenvectors and divide by the square root of the eigenvalues. This gives us sphered data.
3. Concatenate these two sphered data streams.
4. Perform PPCA on this data, to get eigenvectors $\mathbf{V}_3$ and eigenvalues Λ_3.
5. To recover the CCA directions, $W_1 = \mathbf{V}_1\Lambda_1^{-\frac{1}{2}}\mathbf{V}_{3,1}$, $W_2 = \mathbf{V}_2\Lambda_2^{-\frac{1}{2}}\mathbf{V}_{3,2}$, where we have used the notation $\mathbf{V}_{3,i}$ to denote the appropriate part of $\mathbf{V}_3$.

This algorithm is easily shown to perform well on both artificial and real data (see Table 1, 'Sphere-concatenate' for a comparison on our standard data set).

3.1 Sparsifying the Data

The computational intensity of Gaussian process methods are very dependent on the number of data samples we have and so one criticism of the method of this section might be that we are now performing PPCA in the Step 4 of the algorithm on a data set which is twice as long as previously.

We may address this problem with the Sparse Kernel Principal Component method [10]. Tipping proposes to sparsify Kernel PCA by specifying the covariance matrix of the data as

$$C = \sigma_n^2 I + \sum_{i=1}^{N} a_i \phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T = \sigma_n^2 I + \Phi^T A \Phi \tag{7}$$

where the weights a_i are adjustable parameters which are positioned on the main diagonal of diagonal matrix , A, in the last equation and he has performed a nonlinear mapping of the data using the function, $\phi(.)$. Kernel PCA [9] utilises the "kernel trick": provided you can find the scalar product $K_{ij} = \phi(\mathbf{x}_i).\phi(\mathbf{x}_j)$, you need never actually require the individual functions $\phi(.)$. By maximising

[1] Note that we retain all eigenvalues,vectors at this stage.

the likelihood of the data under this model, he shows that we are performing a reduced PCA. Tipping derives an algorithm based on Kernel PCA [9] to find these weights by iterating

$$\Sigma = (A^{-1} + K)^{-1} \tag{8}$$

$$\mu_n = \sigma^{-2} \Sigma \mathbf{k}_n \tag{9}$$

$$a_i^{new} = \frac{\sum_{n=1}^{N} \mu_{ni}^2}{N(1 - \Sigma_{ii}/a_i} \tag{10}$$

where K is the positive definite kernel matrix. We first calculate appropriate Kernel matrices of the two data streams separately giving K_1 and K_2. We may use Tipping's method in our algorithm replacing Step 1 with the iteration

$$\Sigma_1 = (A^{-1} + K_1)^{-1} \tag{11}$$

$$\mu_{1,n} = \sigma^{-2} \Sigma_1 \mathbf{k}_{1,n} \tag{12}$$

$$\Sigma_2 = (A^{-1} + K_2)^{-1} \tag{13}$$

$$\mu_{2,n} = \sigma^{-2} \Sigma_2 \mathbf{k}_{2,n} \tag{14}$$

$$a_i^{new} = \frac{\sum_{n=1}^{N} \mu_{1,ni}^2}{N(1 - \Sigma_{1,ii}/a_i)} + \frac{\sum_{n=1}^{N} \mu_{2,ni}^2}{N(1 - \Sigma_{2,ii}/a_i)} \tag{15}$$

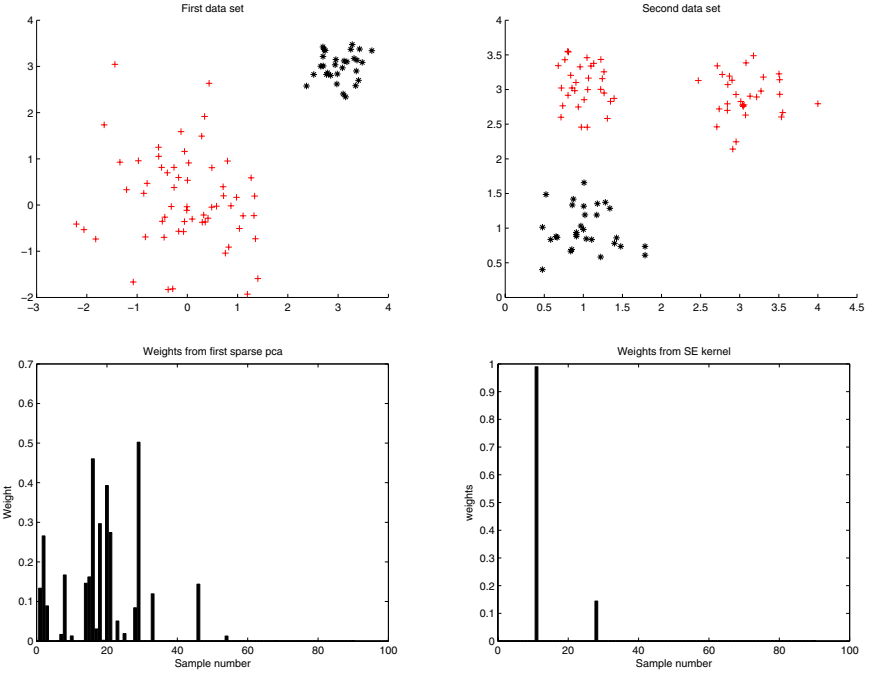


Fig. 1. The top row shows the two data sets. The bottom row the weights found with the linear kernel (left) and the squared exponential kernel (right).

Note that while we are sparsifying two data streams with different covariance matrices, we are utilising a single A matrix. We require this since we wish to identify pairs of important data points simultaneously.

We illustrate the effects of this algorithm in Figure 1: we create 90 samples of two data sets, the first 30 of which are such that the corresponding elements come from related clusters (the black '*'s in Figure 1) while the last 60 samples contain no such relationship - $\frac{1}{2}$ of these samples in one data set come from one cluster while the other $\frac{1}{2}$ come from another cluster; in the other data set these samples are drawn from a widely dispersed cluster. We show the weights from both the linear and the squared exponential covariance matrices in that figure. The degree of sparsification can be controlled by the σ parameter. Even if we set it low (and thereby do not get an appropriate degree of sparsification) we

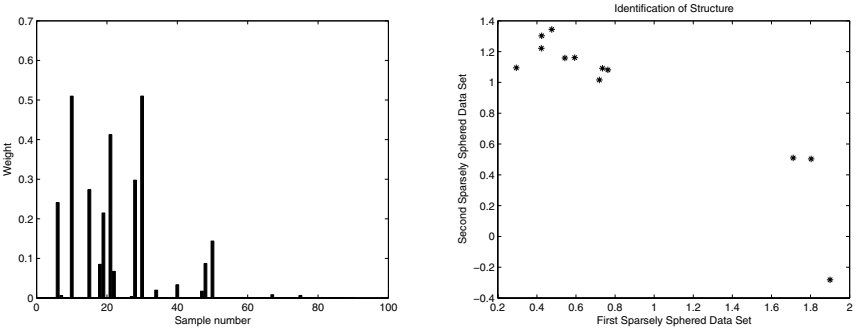


Fig. 2. Left: weights found by linear covariance method. Plots of first sphered coordinate in both data sets. The wrongly included elements are clearly identified.

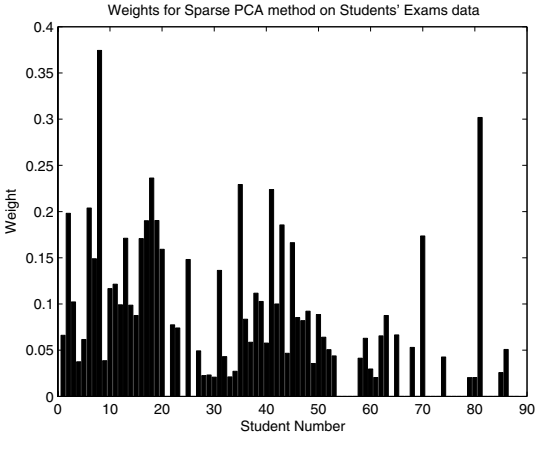


Fig. 3. The weights given to each student's marks after the first stage of the sparse kernel PCA method

may identify samples which have been wrongly included by plotting the sphered data from the first data set against the sphered data in the second data set. This is illustrated in Figure 2 in which we show a simulation in which three samples from the non-matching clusters have been identified. We see that they are easily identified if we plot the first elements of the sphered data from each data stream.

For comparison, we have used this method (with the linear kernel) on the students' exams data; results are shown in Table 1 from a simulation which used $\sigma = 20$. The level of sparsity may be gauged from Figure 3 in which we show the weights, a_i after the first stage of the algorithm. We see from Table 1 that the sparsity has been achieved with the loss of some accuracy.

Table 1. In each section, the middle line gives the weight vector for the open book exams, the bottom line gives the weight vector for the closed book exams. The last column shows the cosine of the angle between the standard statistical method and the other methods.

Standard Statistics				cosine
$\mathbf{w}_1$	0.0260	0.0518		1
$\mathbf{w}_2$	0.0824	0.0081	0.0035	1
Neural Network - Lagrangian Method				
$\mathbf{w}_1$	0.0264	0.0526		1.0000
$\mathbf{w}_2$	0.0829	0.0098	0.0041	0.9998
Neural Network - Gen. Eigenproblem				
$\mathbf{w}_1$	0.0270	0.0512		0.9998
$\mathbf{w}_2$	0.0810	0.0090	0.0040	0.9999
Gaussian Process CCA - one l				
$\mathbf{w}_1$	0.0272	0.0499		0.9994
$\mathbf{w}_2$	0.1163	0.0063	-0.0035	0.9964
Gaussian Process CCA -diag. cov.				
$\mathbf{w}_1$	0.0260	0.0513		1.0000
$\mathbf{w}_2$	0.0896	0.0143	-0.0103	0.9862
GP CCA - noise cov				
$\mathbf{w}_1$	0.0161	0.0620		0.9778
$\mathbf{w}_2$	0.1047	-0.0541	0.242	0.8299
Sphere-Concatenate				
$\mathbf{w}_1$	0.0183	0.0364		1.0000
$\mathbf{w}_2$	0.0579	0.0057	0.0024	1.0000
Sparse Sphere-Concatenate				
$\mathbf{w}_1$	0.0199	0.0547		0.9933
$\mathbf{w}_2$	0.0449	0.0253	0.0039	0.9149
Probabilistic CCA[1]				
$\mathbf{w}_1$	0.0211	0.0420		1.0000
$\mathbf{w}_2$	0.0668	0.0065	0.0028	1.0000

4 Discussion

Comparative results from the 5 methods are given in Table 1.

We require some method for comparing these results. We have chosen to accept the standard statistical method as the ground truth since all other methods are developed as neural or probabilistic implementations of the standard statistical method. Therefore we compare in the last column of that table the cosine of the angles between the weights found by the other methods and those of the standard statistical technique. We also include in this table the Probabilistic CCA method of [1] and two artificial neural network methods which we have previously investigated [4,3]. We see that generally the methods are reasonably accurate other than the Gaussian Process method of [2]. Sparsification of the sphere-concatenate method also diminishes accuracy of the result. However both probabilistic methods, the Sphere-Concatenate method and the Probabilistic CCA method of [1] lose amplitude which is caused by their noise models which captures some of the amplitude of the correlations. This aspect requires further study.

5 Conclusion

We have shown how canonical correlation analysis can be performed using Gaussian Processes in several different manners. We have shown how a previous method gives results which are not as accurate as either our previous neural methods or a new sphere-concatenate method. We have illustrated a method for sparsifying this last method. However both probabilistic methods which perform accurate CCA also lose amplitude compared to the standard statistical method. Clearly this may be remedied by enforcing $E(y_1^2) = E(y_2^2) = 1$, however the need for another step is less than satisfactory. This aspect will form the basis for further study.

References

1. F. R. Bach and M. I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Dept. of Statistics, University of California, 2005.
2. C. Fyfe and G. Leen. Stochastic processes for canonical correlation analysis. In *14th European Symposium on Artificial Neural Networks*, 2006.
3. Z. K. Gou and C. Fyfe. A canonical correlation neural network for multicollinearity and functional data. *Neural Networks*, 2003.
4. P. L. Lai and C. Fyfe. A neural network implementation of canonical correlation analysis. *Neural Networks*, 12(10):1391–1397, Dec. 1999.
5. P. L. Lai, G. Leen, and C. Fyfe. A comparison of stochastic processes and artificial neural networks for canonical correlation analysis. In *International Joint Conference on Neural Networks*, 2006.
6. D. J. C. MacKay. Introduction to gaussian processes. Technical report, University of Cambridge, http://www.inference.phy.cam.uk/mackay/gpB.pdf, 1997.

7. K. V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis*. Academic Press, 1979.
8. C. E. Rasmussen. *Advanced Lectures on Machine Learning*, chapter Gaussian Processes in Machine Learning, pages 63–71. 2003.
9. B. Scholkopf, A. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
10. M. Tipping. Sparse kernel principal component analysis. In *NIPS*, 2003.
11. C. K. I. Williams. Prediction with gaussian processes: from linear regression to linear prediction and beyond. Technical report, Aston University, 1997.

Theory of a Probabilistic-Dependence Measure of Dissimilarity Among Multiple Clusters

Kazunori Iwata and Akira Hayashi

Faculty of Information Sciences, Hiroshima City University, Hiroshima, 731-3194, Japan
{kiwata, akira}@im.hiroshima-cu.ac.jp

Abstract. We introduce novel dissimilarity to properly measure dissimilarity among multiple clusters when each cluster is characterized by a probability distribution. This measure of dissimilarity is called redundancy-based dissimilarity among probability distributions. From aspects of source coding, a statistical hypothesis test and a connection with Ward's method, we shed light on the theoretical reasons that the redundancy-based dissimilarity among probability distributions is a reasonable measure of dissimilarity among clusters.

1 Introduction

In clustering tasks, dissimilarity among multiple clusters is a fundamental measure to evaluate how different clusters are over a sample space [1,2]. It is sometimes referred to as the clustering evaluation function or clustering validity index [3, 4] [5, Ch. 10.6]. A cluster means a group of samples that obey the same (probabilistic) law. If the measure of dissimilarity (or similarity) is well defined, it is useful for effectively constructing clusters from samples without a supervisor [6, 7, 8]. Some effective applications based on dissimilarity (or similarity) have recently been studied [9,10,11,12]. The aims of this paper are to derive a novel measure of dissimilarity among multiple clusters when each cluster is characterized by a probability distribution (PD), as well as to theoretically justify it. Accordingly, our dissimilarity is one of the probabilistic-dependence measures. In this paper, we formulate a clustering task from a probabilistic point of view, and we then introduce a novel measure of dissimilarity called the redundancy-based dissimilarity. The measures of redundancy-based dissimilarity are defined for samples and for PDs. They are connected with a law that the redundancy-based dissimilarity among samples (RDSS) asymptotically coincides with the redundancy-based dissimilarity among PDs (RDSP). From aspects of source coding, a statistical hypothesis test and a connection with Ward's method, we shed light on the theoretical reasons for the RDSP being a reasonable measure of dissimilarity among clusters. This is because clarifying reasons could play a role as a guide for practical applications, especially in selecting an appropriate measure in a given clustering task.

The organization of this paper is as follows. In Section 2, we present some notations and describe definitions of the RDSS and the RDSP. We show main results in Section 3. The theorems related to the main results are summarized in Section 4. Finally, we give some conclusions in Section 5.

This work was supported in part by Grant-in-Aids 18700157 and 18500116 for scientific research from the Ministry of Education, Culture, Sports, Science, and Technology, Japan.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 311–320, 2006.

2 Redundancy-Based Dissimilarity

Without loss of generality, we formulate a clustering task from a probabilistic point of view. The clustering task deals with a sequence of samples where each sample is classified into one of some clusters. The probabilistic law of information sources varies at each time-step according to a point probability for the choice of a cluster. At each time-step, one of the clusters is independently chosen with the point probability, and then it generates a sample according to the underlying structure of the cluster that is characterized by a PD. To facilitate its exposition, we number the clusters by what we call the label number. The label number of each sample in a sequence means that the sample was generated from the cluster having the label number. In general, the label number of each sample is unknown. We use X to express a stochastic variable (SV) over an arbitrary sample space $\mathcal{X}$ and also use X_i to denote X at the time-step $i \in \mathbb{N}$. Let $\mathcal{L} \triangleq \{1, \ldots, M\}$ be the entire set of label numbers. Let $\mathcal{P}(\mathcal{L})$ be the set of cluster probability density functions (PDFs), that is,

$$\mathcal{P}(\mathcal{L}) \triangleq \{P_m | m \in \mathcal{L}\}, \tag{1}$$

where P_m denotes the PDF of cluster m over $\mathcal{X}$. If the label number of a sample $x \in \mathcal{X}$ is $m \in \mathcal{L}$, then henceforth we express it by $x \sim P_m$. For every time-step i and every $m \in \mathcal{L}$, we define the point probability as

$$\omega(m) \triangleq \Pr(X_i \sim P_m). \tag{2}$$

This implies that a cluster is chosen at each time-step to generate each sample independently and according to an identical PDF ω. For simplicity, we assume that $\omega(m) > 0$ for every $m \in \mathcal{L}$. Throughout this paper, this is an underlying assumption. Hence, it satisfies

$$\sum_{m \in \mathcal{L}} \omega(m) = 1. \tag{3}$$

For any positive number n let $\boldsymbol{X}_n = (X_1, X_2, \ldots, X_n)$ be an information source that consists of the clusters, and is sometimes written as $\boldsymbol{X}$ for brevity when we do not need to indicate n explicitly. The expected value of any function $y(x)$ over $\mathcal{X}$ with respect to P_m is denoted by

$$E_{P_m}[y(x)] \triangleq \int_{x \in \mathcal{X}} P_m(x) y(x)\, dx, \tag{4}$$

for every $m \in \mathcal{L}$. We use I_C to denote an indicator function such that for any condition C,

$$I_C = \begin{cases} 1, & \text{if } C \text{ is true} \\ 0, & \text{otherwise} \end{cases}. \tag{5}$$

Fig. 1 draws an information source treated in the clustering task.

We now introduce novel measures that we call the redundancy-based dissimilarity in this paper. The measures are so called because they are closely related to an information-theoretic redundancy of codeword length, as is discussed in Section 3.2. They are defined as dissimilarity among samples and among PDs, respectively.

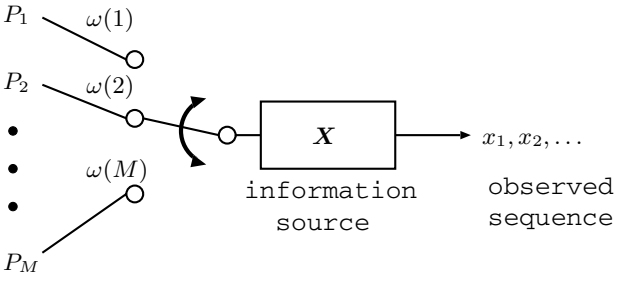


Fig. 1. Information source in the clustering task

Definition 1 (RDSS). *For any subset $\underline{\mathcal{L}} \subseteq \mathcal{L}$, when each sample obeys one element of $\mathcal{P}(\underline{\mathcal{L}})$, we define the squared dissimilarity measure among multiple samples, $(x_1, \ldots, x_n) \in \mathcal{X}^n$ where $n = \sum_{m \in \underline{\mathcal{L}}} \sum_{i=1}^{n} I_{x_i \sim P_m}$, by*

$$\left\{ rds_{\mathcal{P}(\underline{\mathcal{L}})}(x_1, \ldots, x_n) \right\}^2 \triangleq \sum_{m \in \underline{\mathcal{L}}} \left| \sum_{i=1}^{n} I_{x_i \sim P_m} \log \frac{P_m(x_i)}{Q_{\underline{\mathcal{L}}}(x_i)} \right|, \tag{6}$$

where for any $x \in \mathcal{X}$ the PDF $Q_{\underline{\mathcal{L}}}$ is,

$$Q_{\underline{\mathcal{L}}}(x) = \sum_{m \in \underline{\mathcal{L}}} \lambda_{\underline{\mathcal{L}}}(m) P_m(x), \tag{7}$$

where $\lambda_{\underline{\mathcal{L}}}(m)$ is a normalized probability defined by

$$\lambda_{\underline{\mathcal{L}}}(m) \triangleq \frac{\omega(m)}{\sum_{m \in \underline{\mathcal{L}}} \omega(m)}. \tag{8}$$

This $rds_{\mathcal{P}(\underline{\mathcal{L}})}(x_1, \ldots, x_n)$ is referred to as redundancy-based dissimilarity among samples (RDSS).

Definition 2 (RDSP). *For any subset $\underline{\mathcal{L}} \subseteq \mathcal{L}$, we define the squared dissimilarity measure among multiple PDFs $\mathcal{P}(\underline{\mathcal{L}})$ by*

$$\{RDS(\mathcal{P}(\underline{\mathcal{L}}))\}^2 \triangleq \sum_{m \in \underline{\mathcal{L}}} \lambda_{\underline{\mathcal{L}}}(m) \mathrm{D}(P_m \| Q_{\underline{\mathcal{L}}}), \tag{9}$$

where $\lambda_{\underline{\mathcal{L}}}$ is defined in (8) *and $\mathrm{D}(P_m \| Q_{\underline{\mathcal{L}}})$ denotes the information divergence given by*

$$\mathrm{D}(P_m \| Q_{\underline{\mathcal{L}}}) = E_{P_m} \left[\log \frac{P_m(x)}{Q_{\underline{\mathcal{L}}}(x)} \right], \tag{10}$$

where the PDF $Q_{\underline{\mathcal{L}}}$ is defined in (7)*. This $RDS(\mathcal{P}(\underline{\mathcal{L}}))$ is referred to as redundancy-based dissimilarity among PDs (RDSP).*

Note that the value of $RDS(\mathcal{P}(\underline{\mathcal{L}}))$ vanishes if and only if all the PDFs in $\mathcal{P}(\underline{\mathcal{L}})$ are equal. It is well-known that when $|\underline{\mathcal{L}}| = 2$, for any $m \in \mathcal{L}$ and $m' \in \mathcal{L}$ the distance $RDS(P_m, P_{m'})$ is a metric between P_m and $P_{m'}$. It is referred to as the Jensen-Shannon divergence. For the details and the proof, see [13, 14, 15].

3 Main Results

In Section 3.1 we consider an appropriate measure of dissimilarity among multiple clusters in terms of Sanov's Theorem, when each cluster is characterized by a PD. From aspects of source coding, a statistical hypothesis test and a connection with Ward's method, we explain that the RDSP is a reasonable measure of dissimilarity among clusters in Sections 3.2–3.4.

3.1 Sanov's Theorem in Clustering Task

We focus on any two clusters $m \in \mathcal{L}$ and $m' \in \mathcal{L}$ in the information source shown in Fig. 1. This is for simplicity of development and the discussion below holds even if we deal with more than two clusters. When clusters are characterized by PDs, a popular and straightforward way for measuring dissimilarity between the two clusters is to apply information divergence defined as

$$\mathrm{D}(P_m \| P_{m'}) = E_{P_m}\left[\log \frac{P_m(x)}{P_{m'}(x)}\right] \quad \text{for any } m \in \mathcal{L} \text{ and } m' \in \mathcal{L}, \tag{11}$$

where P_m and $P_{m'}$ denote the PDFs of the cluster m and the cluster m', respectively (An alternative way is to apply the symmetric sum, $\mathrm{D}(P_m \| P_{m'})/2 + \mathrm{D}(P_{m'} \| P_m)/2$, but it still does not satisfy the triangle inequality). This means that the dissimilarity between the two clusters is measured by the difference of empirical distributions between two sequences of samples. One sequence is generated from the PDF of the cluster m and the other is generated from the PDF of the cluster m', since the difference of the empirical distributions becomes the difference between the two PDFs as $n \to \infty$. In such cases, from a probabilistic point of view, the justification of applying $\mathrm{D}(P_m \| P_{m'})$ as a dissimilarity measure in clustering tasks might be based on Theorem 1 which is a variant of Sanov's theorem. For the original Sanov's theorem, see [16] or [17, Ch. 6.2].

Theorem 1 (Clustering Task Version of Sanov's Theorem). *Let $\boldsymbol{X} = (X_1, X_2, \ldots)$ be an information source, in which each SV is drawn independently according to an identical PDF $P_{m'}$ over $\mathcal{X}$. Let $\mathcal{M}(\mathcal{X})$ be the set of all possible PDs over a Polish space $\mathcal{X}$ [17, Appendix B.3]. For any subset $\underline{\mathcal{X}} \subseteq \mathcal{X}$, define an empirical PD in $\mathcal{M}(\mathcal{X})$ by*

$$\mathsf{R}_n(\underline{\mathcal{X}}) \triangleq \frac{1}{n}\sum_{i=1}^{n} I_{X_i \in \underline{\mathcal{X}}}, \tag{12}$$

where $X_i \sim P_{m'}$ holds for $i = 1, \ldots, n$. Also, for any subset $\underline{\mathcal{X}} \subseteq \mathcal{X}$, define a PD in $\mathcal{M}(\mathcal{X})$ as

$$\mathsf{R}(\underline{\mathcal{X}}) \triangleq \int_{x \in \underline{\mathcal{X}}} P_{m'}(x)\, dx. \tag{13}$$

It holds that $\mathsf{R}_n(\underline{\mathcal{X}}) \to \mathsf{R}(\underline{\mathcal{X}})$ in probability as $n \to \infty$. Then, for any PD S in $\mathcal{M}(\mathcal{X})$ the empirical PD R_n satisfies a large deviation principle whose rate function is given by

$$U(\mathsf{S} \,|\, \mathsf{R}) = \begin{cases} \int_{\mathcal{X}} d\mathsf{S}(\underline{\mathcal{X}}) \log \frac{d\mathsf{S}(\underline{\mathcal{X}})}{d\mathsf{R}(\underline{\mathcal{X}})} & \text{if } g = \frac{d\mathsf{S}}{d\mathsf{R}} \text{ exists,} \\ \infty & \text{otherwise,} \end{cases} \tag{14}$$

where g represents the Radon-Nikodym derivative of S *with respect to* R. *Also, the empirical PD* $\mathsf{R}_n(\underline{\mathcal{X}})$ *asymptotically goes away from*

$$\mathsf{S}(\underline{\mathcal{X}}) \triangleq \int_{x \in \underline{\mathcal{X}}} P_m(x)\, dx, \tag{15}$$

and the asymptotic speed with respect to n *is described as*

$$\lim_{n\to\infty} \frac{1}{n} \log \Pr\left(\mathsf{R}_n = \mathsf{S}\right) = -U(\mathsf{S} \,|\, \mathsf{R}), \tag{16}$$

$$= -\mathrm{D}(P_m \| P_{m'}). \tag{17}$$

Theorem 1 states that under the probabilistic law given by $P_{m'}$, the probability of all sequences of samples which obey the other law given by $d\mathsf{S} = P_m$ vanishes exponentially with n. In addition, the vanishing speed depends on $\mathrm{D}(P_m \| P_{m'})$ in the exponent. Hence, information divergence does not simply express some difference between two PDs. Also, applying $\mathrm{D}(P_m \| P_{m'})$ as a dissimilarity measure among clusters is indeed reasonable, but only when an information source is drawn according to an identical PDF $P_{m'}$, that is, when the PDF of the information source consists of a PDF of one cluster. However, it is not reasonable, at least in the clustering task treated in this paper, because here the information source consists of multiple cluster PDFs. Accordingly, this raises the question of what dissimilarity we should employ in the clustering task. In fact the RDSP defined in Definition 2 gives quite a reasonable answer to the question because the RDSP regards the multiple PDFs $\mathcal{P}(\underline{\mathcal{L}})$ as one unified PDF $Q_{\underline{\mathcal{L}}}$ of the information source.

3.2 Source Coding

In Fig. 1, the probabilistic law of the information source varies at each time-step according to a point probability ω for the choice of a cluster. Such a source is called an arbitrarily varying source [18, Ch. 3] in information theory. For brevity, we employ $\boldsymbol{x}_n \triangleq x_1, \ldots, x_n$ to denote a sequence of n samples from the information source. Suppose that a sequence $\boldsymbol{x}_n$ is observed from the information source $\boldsymbol{X}$ but the label number of each sample x_i in the sequence is unknown. When we use a predictive estimation to describe the information source, for any subset $\underline{\mathcal{L}} \subseteq \mathcal{L}$ the unified PDF of the information source is described as

$$Q_{\underline{\mathcal{L}}}(x_i) \triangleq \sum_{m \in \underline{\mathcal{L}}} \lambda_{\underline{\mathcal{L}}}(m) P_m(x_i), \tag{18}$$

where $\lambda_{\underline{\mathcal{L}}}$ is defined in (8). This is exactly equal to (7). Clearly, the redundancy caused by the label numbers being unknown is always non-negative,

$$\text{redundancy} = \mathrm{H}(Q_{\underline{\mathcal{L}}}) - \sum_{m \in \underline{\mathcal{L}}} \lambda_{\underline{\mathcal{L}}}(m) \mathrm{H}(P_m) \geq 0, \tag{19}$$

where the entropies are

$$\mathrm{H}(Q_{\underline{\mathcal{L}}}) = -E_{Q_{\underline{\mathcal{L}}}} \left[\log Q_{\underline{\mathcal{L}}}(x)\right], \tag{20}$$

$$\mathrm{H}(P_m) = -E_{P_m} \left[\log P_m(x)\right] \quad \text{for every } m \in \underline{\mathcal{L}}. \tag{21}$$

In the equation (19), the first term exhibits the entropy of a unified PDF given by $Q_{\underline{\mathcal{L}}}$, and the second term also exhibits the entropy of different PDFs given by $\mathcal{P}(\underline{\mathcal{L}})$. The second term is the limit of the achievable coding rate when the label number of each sample is known. The redundancy states that if all the PDFs in $\mathcal{P}(\underline{\mathcal{L}})$ are equal, then it vanishes and hence we can regard the different clusters as one unified cluster. In the clustering task, the redundancy gives an important meaning to the RDSP as a measure of dissimilarity among clusters, because for a given $\mathcal{P}(\underline{\mathcal{L}})$ and $Q_{\underline{\mathcal{L}}}$, transforming (19) yields

$$\text{redundancy} = \left\{RDS(\mathcal{P}(\underline{\mathcal{L}}))\right\}^2. \tag{22}$$

The RDSP is so called for this reason. Therefore, the RDSP is a reasonable measure of dissimilarity among clusters since it stands for a theoretical bound of the amount of loss of information when different clusters are regarded as one unified cluster.

3.3 The Statistical Hypothesis Test

We again consider the information source that is shown in Fig. 1. Recall that the unified PDF of the information source is described as (18). If we observe a sequence $\boldsymbol{x}_n$ from the information source and know the label number of each sample x_i, then according to the label number we divide $\boldsymbol{x}_n$ into subsequences $\left\{\boldsymbol{x}^{(m)} | m \in \underline{\mathcal{L}}\right\}$ where each subsequence is, for every $m \in \underline{\mathcal{L}}$, $\boldsymbol{x}^{(m)} \triangleq \{x_i \in \mathcal{X} | x_i \sim P_m, i = 1, \ldots, n\}$. For notational convenience, let $n_m = \left|\boldsymbol{x}^{(m)}\right|$ hereafter and hence it satisfies

$$\sum_{m \in \underline{\mathcal{L}}} n_m = n. \tag{23}$$

To measure dissimilarity among clusters, for a given n, $\mathcal{P}(\underline{\mathcal{L}})$, and $Q_{\underline{\mathcal{L}}}$, we reason which hypothesis is better: one is to regard $\boldsymbol{x}_n$ as an output from a unified cluster given by $Q_{\underline{\mathcal{L}}}$, and the other is to regard it as an output from different clusters given by $\mathcal{P}(\underline{\mathcal{L}})$. Accordingly, we take the corresponding two hypotheses designated by

$$\begin{aligned} H_0 : &\quad \boldsymbol{x}^{(m)} \sim Q_{\underline{\mathcal{L}}} \text{ for every } m \in \underline{\mathcal{L}}, \\ H_1 : &\quad \boldsymbol{x}^{(m)} \sim P_m \text{ for every } m \in \underline{\mathcal{L}}, \end{aligned} \tag{24}$$

respectively. We are now interested in accepting H_1 since it is indeed true in the information source. In fact it is easy to construct a test for increasing the probability that H_1 is accepted, but it simultaneously increases the probability of error in the test. For any test δ, let $\alpha_\delta(\boldsymbol{x}^{(m)})$ be the probability of the power so that the test δ accepts H_1 successfully when H_1 is true. On the other hand, let $\beta_\delta(\boldsymbol{x}^{(m)})$ be the probability of the error so that the test δ incorrectly accepts H_1 when H_0 is true. Needless to say, the result of the test depends on n, $\mathcal{P}(\underline{\mathcal{L}})$, and $Q_{\underline{\mathcal{L}}}$. From Neyman-Pearson's lemma [19, Theorem 12.7.1], in the sequel the most powerful test (MPT) δ^* is given as follows: for every $m \in \underline{\mathcal{L}}$, if

$$\frac{1}{n} \log \frac{P_m(\boldsymbol{x}^{(m)})}{Q_{\underline{\mathcal{L}}}(\boldsymbol{x}^{(m)})} = \frac{1}{n} \sum_{i=1}^{n} I_{x_i \sim P_m} \log \frac{P_m(x_i)}{Q_{\underline{\mathcal{L}}}(x_i)} \leq k_n^{(m)}, \tag{25}$$

then the hypothesis H_0 is accepted. On the other hand, if

$$\frac{1}{n}\log\frac{P_m(\boldsymbol{x}^{(m)})}{Q_{\underline{\mathcal{L}}}(\boldsymbol{x}^{(m)})}=\frac{1}{n}\sum_{i=1}^{n}I_{x_i\sim P_m}\log\frac{P_m(x_i)}{Q_{\underline{\mathcal{L}}}(x_i)}>k_n^{(m)}, \tag{26}$$

then the hypothesis H_1 is accepted. In these equations, the criterion $k_n^{(m)}$ is a positive number such that for a given $\alpha_{\delta^*}(\boldsymbol{x}^{(m)})$,

$$\alpha_{\delta^*}(\boldsymbol{x}^{(m)})=\Pr\left(\frac{1}{n}\log\frac{P_m(\boldsymbol{x}^{(m)})}{Q_{\underline{\mathcal{L}}}(\boldsymbol{x}^{(m)})}>k_n^{(m)}\,\middle|\,H_1\text{ is true.}\right), \tag{27}$$

when H_1 is true. Then, we have to decrease the probability of error with respect to n,

$$\beta_{\delta^*}(\boldsymbol{x}^{(m)})=\Pr\left(\frac{1}{n}\log\frac{P_m(\boldsymbol{x}^{(m)})}{Q_{\underline{\mathcal{L}}}(\boldsymbol{x}^{(m)})}>k_n^{(m)}\,\middle|\,H_0\text{ is true.}\right), \tag{28}$$

when H_0 is true. Notice that a dissimilarity measure among the clusters is characterized by the criterion, and by the decreasing speed of the probability of error with respect to n. In fact, for every $m\in\underline{\mathcal{L}}$ these satisfy

$$\lim_{n\to\infty}k_n^{(m)}=\lambda_{\underline{\mathcal{L}}}(m)\mathrm{D}(P_m\|Q_{\underline{\mathcal{L}}}), \tag{29}$$

$$\lim_{n\to\infty}\frac{1}{n}\log\beta_{\delta^*}(\boldsymbol{x}^{(m)})=-\lambda_{\underline{\mathcal{L}}}(m)\mathrm{D}(P_m\|Q_{\underline{\mathcal{L}}}). \tag{30}$$

These equations will be proved in Theorem 3. Hence, the criterion $k_n^{(m)}$ of the MPT converges to each term of the squared RDSP defined in (9). Also, the decreasing speed of the probability $\beta_{\delta^*}(\boldsymbol{x}^{(m)})$ of error with respect to n is determined by each term of the squared RDSP. In short, if the RDSP is large, then the MPT for the hypotheses works well even in the case where n is not sufficiently large. Therefore, the RDSP is a reasonable measure of dissimilarity among clusters since it determines the theoretical bound of the MPT for accepting the hypothesis H_1, that is, the true hypothesis in the information source.

3.4 A Connection Between the RDSP and Ward's Method

We explain the good ability of the RDSP via a connection between it and Ward's method [20]. Ward's method is one of the most popular methods in hierarchical cluster analysis and has been applied to many practical tasks (see [21], for example). It is well-known that Ward's method exhibits the ability to accurately construct the hierarchical structure of clusters. The key point of the method is to minimize the increase of the sum of squared differences from each cluster's mean whenever unifying any two clusters. That is, when we create a new cluster by unifying any two clusters, we always select two clusters m' and m'' that minimize the increase (Ward's measure),

$$\Delta_W(m',m'')=S_{m',m''}-S_{m'}-S_{m''}, \tag{31}$$

where S_m means the sum of squares within the cluster $m\in\{m',m''\}$ and $S_{m',m''}$ also means that within the unified cluster. Since a deviation of PDF over a sample space is

characterized by the entropy, replacing the sum of squares, which is a measure of the deviation of the samples, by the entropy yields

$$\Delta_R(m', m'') = \mathrm{H}(Q_{m',m''}) - \lambda_{m',m''}(m')\mathrm{H}(P_{m'}) - \lambda_{m',m''}(m'')\mathrm{H}(P_{m''}). \quad (32)$$

This is exactly equal to the squared RDSP in the case of $\underline{\mathcal{L}} = \{m', m''\}$. Intuitively, the redundancy expresses the increase of entropy in unifying PDs of clusters. Thus, the RDSP is a reasonable measure of dissimilarity among clusters since it is based on the same principle as Ward's measure in hierarchical cluster evaluations, though the two measures are essentially distinct measures because Ward's measure is a dissimilarity measure for samples and the RDSP is a dissimilarity measure for PDs.

4 Theorems

Theorem 2 implies that the RDSP is asymptotically approximated by the left side in (35). It is useful in practice because we can avoid serious computational complexity for the integration in respect to the information divergence of the RDSP.

Theorem 2. *Let* $n_m \triangleq \sum_{i=1}^n I_{x_i \sim P_m}$ *for every* $m \in \mathcal{L}$. *If*

$$E_{\lambda_{\underline{\mathcal{L}}}}[(n_m/n - \lambda_{\underline{\mathcal{L}}}(m))^2] < \infty, \quad (33)$$

$$E_{P_m}\left[\left(\frac{1}{n_m}\sum_{i=1}^n I_{x_i \sim P_m} \log \frac{P_m(x_i)}{Q_{\underline{\mathcal{L}}}(x_i)} - \mathrm{D}(P_m \| Q_{\underline{\mathcal{L}}})\right)^2\right] < \infty, \quad (34)$$

for any subset $\underline{\mathcal{L}} \subseteq \mathcal{L}$, *then*

$$\frac{1}{n}\left\{rds_{\mathcal{P}(\underline{\mathcal{L}})}(x_1, \ldots, x_n)\right\}^2 \to \left\{RDS(\mathcal{P}(\underline{\mathcal{L}}))\right\}^2, \quad (35)$$

in probability as $n \to \infty$.

Proof. One of the clusters is independently chosen at each time-step and according to an identical PDF λ. In addition, each sample is independently drawn by a chosen cluster. Hence, if (33) and (34) hold, then by the weak law of large numbers there exists a positive number ϵ such that

$$\lim_{n\to\infty} \Pr\left\{\left|\frac{1}{n}\sum_{m\in\underline{\mathcal{L}}}\sum_{i=1}^n I_{x_i\sim P_m} \log \frac{P_m(x_i)}{Q_{\underline{\mathcal{L}}}(x_i)} - \sum_{m\in\underline{\mathcal{L}}} \lambda_{\underline{\mathcal{L}}}(m)\mathrm{D}(P_m\|Q_{\underline{\mathcal{L}}})\right| > \epsilon\right\} = 0, \quad (36)$$

and hence we obtain (35). ∎

Theorem 3 is an analogy of Stein's lemma [17, Lemma 3.4.7].

Theorem 3. *We consider the hypotheses designated by* (24). *For any subset* $\underline{\mathcal{L}} \subseteq \mathcal{L}$, *if the hypothesis* H_1 *is true, then for every* $m \in \underline{\mathcal{L}}$ *the equation* (29) *holds. Also, if the hypothesis* H_0 *is true, then for every* $m \in \underline{\mathcal{L}}$ *the equation* (30) *holds.*

Proof. If the hypothesis H_1 is true, then by the weak law of large numbers the left side of (25) and (26) is,

$$\frac{1}{n}\log\frac{P_m(\boldsymbol{x}^{(m)})}{Q_{\underline{\mathcal{L}}}(\boldsymbol{x}^{(m)})} \to \lambda_{\underline{\mathcal{L}}}(m)\mathrm{D}(P_m\|Q_{\underline{\mathcal{L}}}), \tag{37}$$

in probability as $n \to \infty$. If the equation (29) does not hold, then we cannot set an arbitrary number in $\alpha_\delta(\boldsymbol{x}^{(m)})$ since $\alpha_\delta(\boldsymbol{x}^{(m)})$ must go to zero or one as $n \to \infty$ to satisfy (27). Therefore, the equation (29) holds. Next, we obtain (30) via Gärtner-Ellis theorem [22, 23]. For every $m \in \underline{\mathcal{L}}$, let

$$Y_n^{(m)} \triangleq \log\frac{P_m(\boldsymbol{x}^{(m)})}{Q_{\underline{\mathcal{L}}}(\boldsymbol{x}^{(m)})} - nk_n^{(m)}. \tag{38}$$

We examine the convergence speed that $Y_n^{(m)}/n$ goes into $(0,\infty)$ as $n \to \infty$. The moment generating function of $Y_n^{(m)}$ with respect to $Q_{\underline{\mathcal{L}}}^{n_m}$ is

$$M_{Q_{\underline{\mathcal{L}}}^{n_m}}(\theta) = E_{Q_{\underline{\mathcal{L}}}^{n_m}}\left[\exp(\theta Y_n^{(m)})\right]. \tag{39}$$

With the definition,

$$\phi^{(m)}(\theta) \triangleq \lim_{n\to\infty}\frac{1}{n}\log M_{Q_{\underline{\mathcal{L}}}^{n_m}}(\theta), \tag{40}$$

by (29) we have

$$\phi^{(m)}(\theta) = -\theta\lambda(m)\mathrm{D}(P_m\|Q_{\underline{\mathcal{L}}}) + \lambda(m)\log\left(\int_{\mathcal{X}} P_m(x)^\theta Q_{\underline{\mathcal{L}}}(x)^{1-\theta}\,dx\right). \tag{41}$$

Accordingly, the rate function is described as

$$U_m(\theta, y) = \sup_{\theta\in\mathbb{R}}\left(\theta y - \phi^{(m)}(\theta)\right). \tag{42}$$

From Gärtner-Ellis theorem (for example, see [17, Ch. 2]), we have

$$\lim_{n\to\infty}\frac{1}{n}\log\Pr\left\{\frac{Y_n^{(m)}}{n} \in (0,\infty)\right\} = -\inf_{y\in(0,\infty)} U_m(\theta, y), \tag{43}$$

$$= -\sup_{\theta\in\mathbb{R}}\left(-\phi^{(m)}(\theta)\right). \tag{44}$$

By solving $-d\phi^{(m)}(\theta)/d\theta = 0$ on the sup-function, we immediately obtain $\theta = 1$. Therefore, since the rate function is convex, substituting $\theta = 1$ into (44) yields (30). ∎

5 Conclusion

We have introduced the RDSP, a probabilistic-dependence measure among multiple PDs, for measuring dissimilarity among multiple clusters. Also, we have elucidated the theoretical reasons that the RDSP is a reasonable measure of dissimilarity among multiple clusters. These reasons could play a role as a guide for practical applications of the clustering task.

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. 2nd edn. John Wiley & Sons, New York (2001)
2. Xu, R., Wunsch-II, D.C.: Survey of clustering algorithms. IEEE Transactions on Neural Networks **16**(3) (2005) 645–678
3. Gokcay, E., Principle, J.C.: Information theoretic clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(2) (2002) 158–171
4. Maulik, U., Bandyopadhyay, S.: Performance evaluation of some clustering algorithms and validity indices. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(12) (2002) 1650–1654
5. Webb, A.R.: Statistical Pattern Recognition. 2nd edn. John Wiley & Sons, New York (2002)
6. Yeung, D., Wang, X.: Improving performance of similarity-based clustering by feature weight learning. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(4) (2002) 556–561
7. Fred, A.L., Leitão, J.M.: A new cluster isolation criterion based on dissimilarity increments. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(8) (2003) 944–958
8. Yang, M.S., Wu, K.L.: A similarity-based robust clustering method. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(4) (2004) 434–448
9. Tipping, M.E.: Deriving cluster analytic distance functions from gaussian mixture model. In: Proceedings of the 9th International Conference on Artificial Neural Networks. Volume 2., Edinburgh, UK, IEE (1999) 815–820
10. Prieto, M.S., Allen, A.R.: A similarity metric for edge images. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(10) (2003) 1265–1273
11. Wei, J.: Markov edit distance. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(3) (2004) 311–321
12. Srivastava, A., Joshi, S.H., Mio, W., Liu, X.: Statistical shape analysis: Clustering, learning, and testing. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(4) (2005) 590–602
13. Österreicher, F.: On a class of perimeter-type distances of probability distributions. Cybernetics **32**(4) (1996) 389–393
14. Topsøe, F.: Some inequalities for information divergence and related measures of discrimination. IEEE Transactions on Information Theory **46**(4) (2000) 1602–1609
15. Endres, D.M., Schindelin, J.E.: A new metric for probability distributions. IEEE Transactions on Information Theory **49**(7) (2003) 1858–1860
16. Sanov, I.N.: On the probability of large deviations of random variables. Selected Translations in Mathematical Statistics and Probability **1** (1961) 213–244
17. Dembo, A., Zeitouni, O.: Large Deviations Techniques and Applications. 2nd edn. Volume 38 of Applications of Mathematics. Springer, New York (1998)
18. Han, T.S., Kobayashi, K.: Mathematics of Information and Coding. Volume 203 of Translations of Mathematical Monographs. American Mathematical Society, Providence, RI (2002)
19. Cover, T.M., Thomas, J.A.: Elements of Information Theory. 1st edn. Wiley series in telecommunications. John Wiley & Sons, Inc., New York (1991)
20. Ward, J.H.: Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association **58**(301) (1963) 236–244
21. Ward, J.H., Hook, M.E.: Application of an hierarchical grouping procedure to a problem of grouping profiles. Educational Psychological Measurement **23**(1) (1963) 69–82
22. Gärtner, J.: On large deviations from the invariant measure. Theory of Probability and Its Applications **22**(1) (1977) 24–39
23. Ellis, R.S.: Large deviations for a general class of random vectors. The Annals of Probability **12**(5) (1984) 1–12

Kernel PCA as a Visualization Tools for Clusters Identifications

Alissar Nasser[1], Denis Hamad[1], and Chaiban Nasr[2]

[1] ULCO, LASL 50 rue F. Buisson, BP 699, 62228 Calais, France
{nasser, Hamad}@lasl.univ-littoral.fr
[2] LU, Faculty of Engineering, Rue Al-Arz, Tripoli, Lebanon
chnasr@ieee.org

Abstract. Kernel PCA has been proven to be a powerful technique as a nonlinear feature extractor and a pre-processing step for classification algorithms. KPCA can also be considered as a visualization tool; by looking at the scatter plot of the projected data, we can distinguish the different clusters within the original data. We propose to use visualization given by KPCA in order to decide the number of clusters. K-means clustering algorithm on both data and projected space is then applied using synthetic and real datasets. The number of clusters discovered by the user is compared to the Davies-Bouldin index originally used as a way of deciding the number of clusters.

Keywords: Clustering, visualization, Kernel PCA, K-means, DB index.

1 Introduction

Clustering has emerged as a popular technique for pattern recognition, image processing, and data mining. It is one of the well-studied techniques, which concerns the partitioning of similar objects into clusters such that objects in the same cluster share some unique properties.

For most clustering algorithms two crucial problems require to be solved: (1) determine the number of clusters K and (2) determine the similarity measure based on which patterns are assigned to corresponding clusters. Many clustering algorithms require that K to be provided as an input parameter. It is obvious that the quality of resulting clusters is largely dependent on the value of K. Many algorithms have been proposed for the estimation of the optimal number of partitions. In [4] a self-organizing feature map is trained. The final network structure allows visualizing high-dimensional data as a two dimensional scatter plot. The resulting representations allow a straightforward analysis of the inherent structure of clusters within the input data. [1] shows that minimization of partition entropy or maximization of partition certainty may be used to estimate the number of data generators, i.e. the number of clusters. A set of kernel functions are fitted to the data using the Expectation-maximization algorithm (EM) to model the probability density function PDF. Then the approach seeks the number of partitions whose linear combination yields the data PDF; densities and classification conditioned on this partition set can then easily obtained.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 321 – 329, 2006.

Girolami in [2] suggests estimating the number of clusters within the data by considering the most dominant terms $\lambda_i \left\{ 1_i^T u_i \right\}^2$ of the eigenvalue decomposition of the kernel matrix created by the dataset. Fig. 1 shows an example of the block structure of the Kernel matrix which distinguishes that there are three clusters in the data.

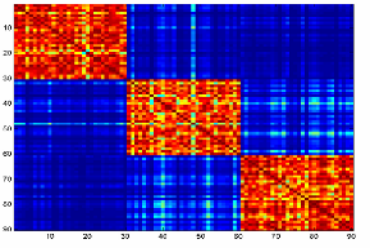


Fig. 1. Scatter plot of the kernel matrix of 3 clusters artificially generated according Gaussian density clearly showing the inherent block structure

In this paper, we propose using Kernel principal component analysis (KPCA) projection method as a visualization tool. KPCA can provide a means to decide the number of clusters in the data by looking at the scatter plot of the lower dimensional projected space. Therefore one can distinguish groups within the projected data and then initialize K-means clustering algorithm. The number of clusters used to initialize this algorithm is evaluated by the Davies and Bouldin index [8]. This index does not depend on either the number of clusters or the clustering algorithm.

The remaining part of the paper is organized as follows. Section 2 describes KPCA projection method. Section 3 presents K-means clustering algorithm, where section 4 shows how the number of clusters can be estimated from the KPCA visualization and describes the Davies and Bouldin index. Simulations on synthetic and real datasets are discussed in section 5.

2 Kernel Principal Component Analysis (KPCA)

Kernel principal component analysis (KPCA) first introduced in [3] has been shown to be an elegant way to extract nonlinear features from the data. KPCA can maintain and enhance those features of the input data which make distinct pattern classes separate from each other. It can provide a means to decide the number of clusters in the data by looking at the scatter plot of the lower dimensional space.

KPCA utilizes kernel trick to perform operation in a new feature space F where data samples are more separable. By using a nonlinear kernel function instead of the standard dot product, we implicitly perform PCA in a high-dimensional space F which is non-linearly related to the input space. Consequently, KPCA produces features which capture the nonlinear structure in the data better than linear PCA.

Given a dataset of L observations x_ℓ $\ell = 1 \ldots L$. $x_\ell \in R^N$. The mapping function is defined by:

$$\begin{aligned} \phi : \mathrm{R}^{\mathrm{N}} &\rightarrow \mathrm{F} \\ x_l &\rightarrow \phi(x_l) \end{aligned} . \tag{1}$$

The correlation matrix in the feature space F is:

$$\tilde{\mathrm{C}} = \frac{1}{\mathrm{L}} \sum_{l=1}^{\mathrm{L}} \phi(x_l)\phi(x_l)^{\mathrm{T}} . \tag{2}$$

KPCA method is based on solving eigenvector system on the transformed space:

$$\tilde{\mathrm{C}}\, \tilde{\mathrm{v}} = \tilde{\lambda}\, \tilde{\mathrm{v}} . \tag{3}$$

where $\tilde{\lambda}$ and $\tilde{\mathrm{v}}$ are the eigenvalue and eigenvector respectively of $\tilde{\mathrm{C}}$.
$\tilde{\mathrm{v}}$ lies in the span of $\phi(x_1),\ldots,\phi(x_{\mathrm{L}})$, thus it is a linear combination of $\phi(x_l)$ elements. Thus, $\tilde{\mathrm{v}}$ can be written as:

$$\tilde{\mathrm{v}} = \sum_{\mathrm{i}=1}^{\mathrm{L}} \mathrm{a}_i \phi(x_i) . \tag{4}$$

Defining the kernel function by:

$$\mathrm{K}(x_l, x_i) = (\phi(x_l)^{\mathrm{T}} \phi(x_i)) . \tag{5}$$

In order to extract principal components of any point x we have to project the image $\phi(x)$ of this point on the M obtained eigenvectors $\tilde{\mathrm{v}}_m$:

$$\tilde{\mathrm{y}}_{\mathrm{m}} = \tilde{\mathrm{v}}_m^{\mathrm{T}} \phi(x) = \sum_{j=1}^{\mathrm{L}} \mathrm{a}_{mj} \mathrm{K}(x_j, x) . \tag{6}$$

The Eigenvectors $\{\tilde{\mathrm{y}}_1, \cdots \tilde{\mathrm{y}}_m, \cdots, \tilde{\mathrm{y}}_M\}$ in F are called nonlinear principal components.

A number of different kernels have been used in many areas of Kernel Machines, such as polynomial, Gaussian, or sigmoid types [3]. We will consider the Gaussian kernel type in this paper.

3 K -Means Clustering Algorithm in Data Space

K-means [9] is an unsupervised clustering algorithm that partitions the dataset into a selected number K of clusters by minimizing a formal objective means-squared-error distortion MSE:

$$\mathrm{MSE} = \sum_{\mathrm{k}=1}^{\mathrm{K}} \sum_{\mathrm{x_i} \in \mathrm{C_k}} \left\| \mathrm{x_i} - \mathrm{c_k} \right\|^2 . \tag{7}$$

This objective means-squared-error represents the Euclidean distance between the samples and the centroids in the input dataset. K-means is an iterative simple, straightforward algorithm. It is based on the firm foundation of analysis of variances.

The result of K-means strongly depends on the initial guess of centroids. It is not obvious what a good K to use is.

4 Estimating the Number of Clusters by KPCA Projected Data Visualization

A plethora of new clustering methods has been proposed in recent years which give very impressive results with highly different data shapes. The important among them include Spectral clustering [5], KPCA which has been observed to perform clustering [6], Kernel Clustering in Feature Space [2] etc.; whilst, the problem of estimating the number of clusters within the dataset still remain. In this section, we will use the KPCA method for data projection and visualization in order to distinguish clusters within the data and therefore determine the cluster's number K. The approach proposed by Girolami in [2] to estimate the number of clusters, suggests considering the most dominant terms $\lambda_i\left\{1_i^T u_i\right\}^2$, where Nx1 dimensional vector 1_N has elements of value 1/N, λi and ui are the eigenvalues/eigenvectors decomposition of the Kernel matrix. In fact, these dominant terms depend highly on the choice of the kernel Gaussian width σ and a bad value will imply a bad estimation of the number of clusters. Fig. 2 shows an example of 2-spheres in R^3 we can see that small variation of the value of the kernel width lead to different estimation of number of clusters by the latter proposed method.

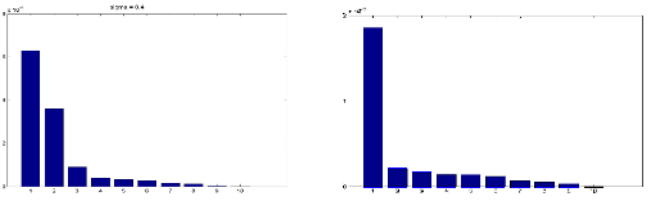


Fig. 2. The most dominant terms $\lambda_i\left\{1_i^T u_i\right\}^2$ for 3D 2-spheres dataset. A Gaussian kernel with width 0.1 (left) and 0.2 (right) were used. (Left) indicates that there are 2 dominant terms and one less dominant i.e. the existence of 2 or less probably 3 clusters. Whereas, (right) indicates the existence of one cluster. Thus for even small variation of σ we got different number of clusters.

One way to get around this problem is to look at the scatter plot of KPCA projected data. Fig. 3 shows the first 2 principal components of KPCA with Gaussian kernel of width 0.1 (left) and 0.2 (right) applied to the 2-spheres dataset. Note that, here we can clearly distinguish the existence of two clusters. Therefore, for different width values, visualization of projected data by KPCA can indicate the number of clusters within the dataset. This is due to the capacity of KPCA to map the data into a higher space where the separation is linear.

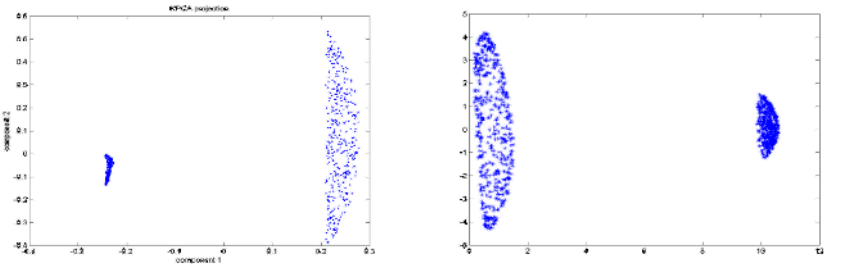


Fig. 3. Scatter plots of the two first kernel principal components for the 2-spheres in R3 with different kernel width values. First image (left) $\sigma = 0.1$, the second (right) $\sigma = 0.2$. We see that for the same variation of σ as in Fig. 2, we can easily distinguish 2 clusters.

4.1 Davies and Bouldin Index

In order to evaluate the number of clusters taken by the KPCA scatter plot, we will use the Davies and Bouldin index (DB) [8]. This index doesn't depend on either the number of clusters or the clustering method. It was originally proposed for deciding when to stop clustering algorithm. The index is plotted against the number of clusters and minimal value indicates the optimal number of clusters within the data. The DB index for K-cluster is:

$$DB(K) = \left(\frac{1}{K}\right) \sum_{k=1}^{K} \max_{k \neq i} \left\{ \frac{S_k + S_i}{d_{ik}} \right\}. \tag{8}$$

Where S_k is the average error for the k^{th} cluster and d_{ik} is the Euclidean distance between the centers of the i^{th} and the k^{th} clusters.

5 Simulation

The purpose of this section is to test the effectiveness of KPCA to visualize high-dimensional data as a two dimensional scatter plot. The resulting representations allow a straightforward analysis of the inherent structure of clusters within the input data and thus determining the number of clusters. Once this number is identified, we apply the well known K-means clustering algorithm. Simulations on synthetic and real datasets are presented. The kernel function used for all experiments is the Gaussian one of width σ. As we saw before the value of σ is tricky, and all results depend highly on it. In our experiments we choose the value that are adequate with the dataset, hence the width σ is tuned.

5.1 Three Gaussian Dataset

This example is a synthetic dataset in which clusters are linearly separable. It is composed of 90 3-D data points generated from 3 Gaussian distribution (30 for each class) of mean vectors (-0.5, -0.2, 0); (0, 0.6, 0) and (0.5, 0, 0) and of variance values equal to 0.1 on each component.

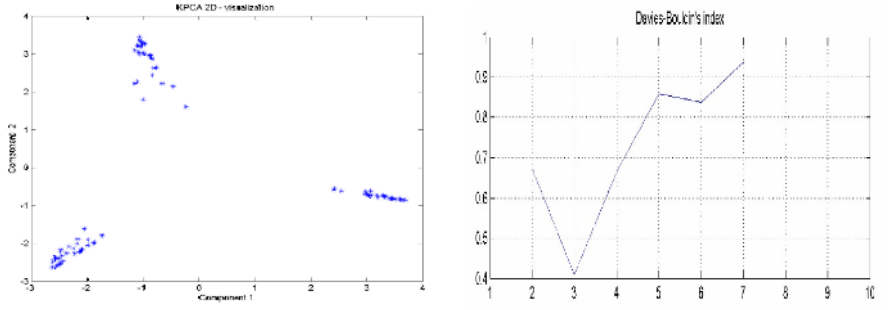


Fig. 4. The scatter plot of the KPCA projection using a Gaussian kernel of width 0.1 showing that there are three clusters with in the data (left). The DB index on K-means clustering algorithm which is minimal for 3 clusters (right).

5.2 2-Spheres

The second example we investigate is composed of two spheres in 3D. The dataset consists of 800 points in three dimensions. 400 points are selected randomly within a hemisphere of radius 0.6 and the rest 400 from a shell defined by two hemispheres of radius 2 and 2.013.

The number of clusters K can be easily estimated by looking at the scatter plot of 2-dimensional projected data by KPCA (Fig. 3). For different σ, KPCA visualization shows the distinct cluster within the data. This number is then used to initialize K-means algorithm.

Fig. 5 shows the scatter plots of DB indexes on both input data and 2-D projected data by KPCA. In input space, this index fails to estimate the correct cluster's number and the value is minimal for 3 clusters. Whereas, scatter plot of DB in projected space, indicates the existence of 2 clusters.

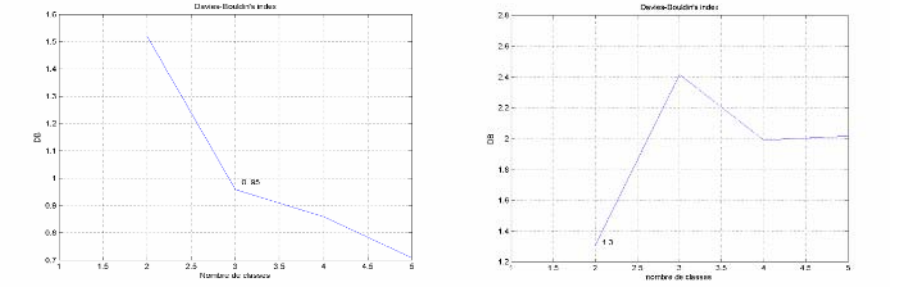


Fig. 5. DB index in input space (left) and KPCA projected space (right). We see that after projection the DB index is minimal for 2 clusters which indicates the existence of 2 clusters.

5.3 Petals Dataset

The dataset consist of 100 datapoints in 2-dimensional space [10]. There are 4 clusters within the data. Fig. 6 shows that KPCA succeed to separate the clusters (left) on the other hand, DB index in input space (middle) and DB in projected space give minimal value for 4 clusters.

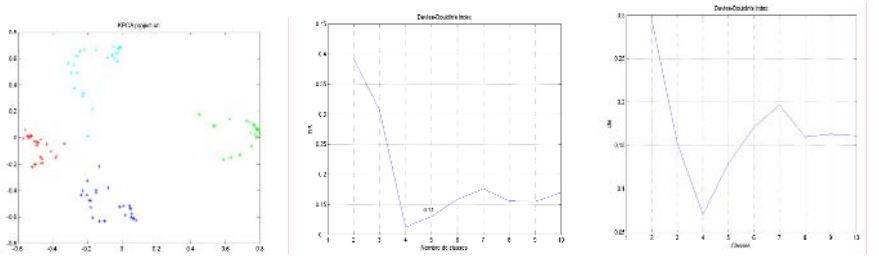


Fig. 6. KPCA projected data (left), DB index in input space (middle), DB index in KPCA projected space (right) the value of σ used is 0.5

5.4 Wine Dataset

As a final example, we present results from a wine recognition problem from the UCI databases. The dataset consists of 178 data points in 12-dimensional space which are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determines the quantities of 12 constituents found in each of the three types of wines. The plot onto the first two principal components of KPCA shows that the number of clusters is equal to three.

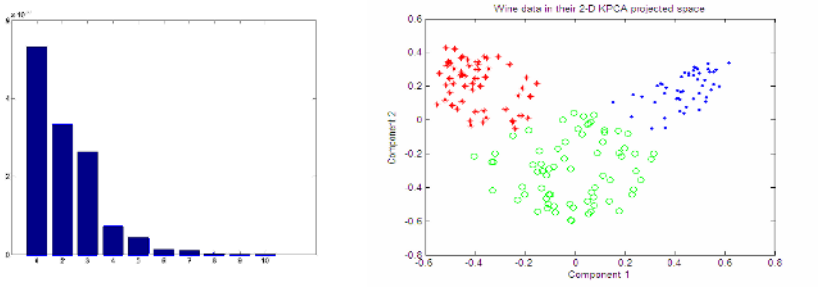


Fig. 7. A Gaussian kernel of width 0.4 was used. Plot of the most dominant terms $\lambda_i \{1_i^T u_i\}^2$ for this dataset indicates that there are 3 dominant terms i.e. the existence of 3 clusters (left). Scatter plot of KPCA projection into the plan indicating the existence of three clusters (right).

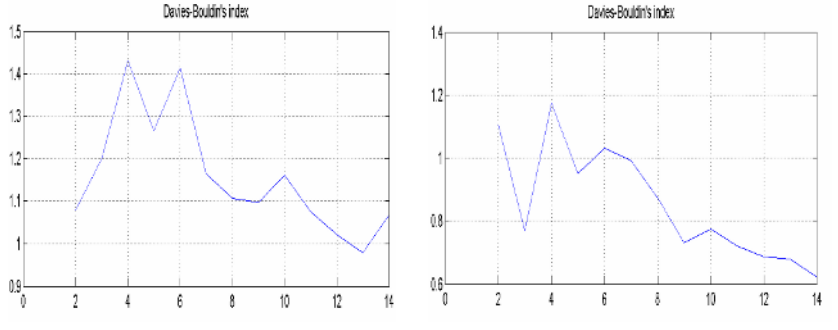


Fig. 8. The BD index in data space (left) and the DB index in 2-D KPCA projected space which is minimal for 2 clusters (right)

5.5 Discussion

Table 1. summarizes the clustering accuracy obtained by applying K-means on original and projected data by KPCA method. It also shows the number of clusters discovered by KPCA and those calculated by the DB index in data and projected spaces respectively. We can state that for the four datasets KPCA correctly gives the number of clusters especially for complex and nonlinearly separable data. This is due to the nonlinear transformation of KPCA which maps the input space to a high dimensional feature space where the transformed data could be linearly separable. Whereas for complex data, the DB index in data space fails to estimate the correct number of clusters. This number correctly estimated by the DB index on projected data by KPCA method.

Table 1. Well-classification rate (WCR) of K-means on input and projected data, estimation of the cluster's number by KPCA and DB index in data space (DB data) and in projected space (DB projected)

Dataset	cluster' Nb.	Cluster's by KPCA	DB data	DB projected	K-means WCR	KPCA + K-means WCR
3 Gaussian	3	3	3	3	100	100
2 Spheres	2	2	3	2	78.87	100
Petals	4	4	4	4	100	100
Wines	3	3	2	3	49.43	95.5

We mention that K-means has been executed many times with different initialized centers, and we pick the partitions that give the smallest DB value. On the other hand, the results of projected data by KPCA were slight sensitive to the Gaussian width of the kernel function used. We conducted a series of experiments for the same data set with different width values; we noticed that results were similar except for large value of σ where KPCA approaches the linear case as it has been proven in [11].

6 Conclusion

We investigate the use of the nonlinear principal component analysis KPCA as a visualization tool; by looking at the scatter plot of the projected data, we can distinguish the different clusters within the original data. Visualization given by KPCA projection method is used in order to decide the number of cluster. K-means clustering algorithm in both data and projected space is then examined and tested. The number of cluster used to initialize this algorithm is evaluated by the Davies-Bouldin index and the most dominant terms. The results show that KPCA visualization on synthetic and real datasets are most accurate than those given by DB index in data space and the most dominant terms $\lambda_i \left\{ l_i^T u_i \right\}^2$. Whereas BD index for projected data gave good estimation of the number of clusters; This is due to the nonlinear

transformation of KPCA which maps the input space to a high dimensional feature space where the transformed data could be linearly separable.

Acknowledgments. This work is supported by region Nord Pas-de-Calais under AutoRIS project.

References

1. Roberts, S.J., Everson, R. & Rezek, I, "Maximum Certainty Data Partitioning", Pattern Recognition, 33:5, (2000).
2. Girolami M., "Mercer kernel-based clustering in feature space", IEEE Transactions on Neural Networks, Volume 13, Issue 3, (2002) Pages:780 – 784.
3. Shölkopf B., Smola A.J., "Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond", the MIT Press, Cambridge, Massachusetts, London, England, (2002).
4. Mu-Chun S., Hsiao-Te C. "A new model of self-organizing neural networks and its application in data projection", IEEE trans, Neural Network vol. 12, Issue 1, P. 153-158, (2001).
5. Ng A.Y., Jordan M.I. and Weiss Y., "On Spectral Clustering: Analysis and an algorithm", NIPS 14, (2002).
6. Christianini N., Shawe-Taylore J. and Kandola J., "Spectral kernel methods for clustering", Neural Information Processing Systems 14,(2002).
7. Han J., Kamber M, Data Mining: Concepts and Techniques. Morgan Kaufmann, (2000).
8. Jain, A.K., Dubes, R.C., "Algorithms for Clustering Data", Prentice Hall, (1988).
9. MacQueen J. B., "Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press, 1:281-297, (1967).
10. http://www.informatics.bangor.ac.uk/~kuncheva/activities/artificial_data.htm.
11. Twining C.J., Taylor, C.J., The use of kernel principal component analysis to model data distributions. Pattern Recognition 36(1): 217-227 (2003).

A Fast Fixed-Point Algorithm for Two-Class Discriminative Feature Extraction*

Zhirong Yang and Jorma Laaksonen

Laboratory of Computer and Information Science
Helsinki University of Technology
P.O. Box 5400, FI-02015 HUT, Espoo, Finland
{zhirong.yang, jorma.laaksonen}@hut.fi

Abstract. We propose a fast fixed-point algorithm to improve the Relevant Component Analysis (RCA) in two-class cases. Using an objective function that maximizes the predictive information, our method is able to extract more than one discriminative component of data for two-class problems, which cannot be accomplished by classical Fisher's discriminant analysis. After prewhitening the data, we apply Newton's optimization method which automatically chooses the learning rate in the iterative training of each component. The convergence of the iterative learning is quadratic, i.e. much faster than the linear optimization by gradient methods. Empirical tests presented in the paper show that feature extraction using the new method resembles RCA for low-dimensional ionosphere data and significantly outperforms the latter in efficiency for high-dimensional facial image data.

1 Introduction

Supervised linear dimension reduction, or discriminative feature extraction, is a common technique used in pattern recognition. Such a preprocessing step not only reduces the computation complexity, but also reveals relevant information in the data.

Fisher's *Linear Discriminant Analysis* (LDA) [3] is a classical method for this task. Modeling each class by a single Gaussian distribution and assuming all classes share a same covariance, LDA maximizes the Fisher criterion of between-class scatter over within-class scatter and can be solved by *Singular Value Decomposition* (SVD). LDA is attractive for its simplicity. Nevertheless, it yields only one discriminative component for two-class problems because the between-class scatter matrix is of rank one. That is, the discriminative information can only be coded with a single number and a lot of relevant information may be lost during the dimensionality reduction.

Loog and Duin [2] extended LDA to the heteroscedastic case based on the simplified Chernoff distance between two classes. They derived an alternative

* Supported by the Academy of Finland in the projects *Neural methods in information retrieval based on automatic content analysis and relevance feedback* and *Finnish Centre of Excellence in Adaptive Informatics Research*.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 330–339, 2006.

criterion which uses the individual scatter matrices of both classes. The generalized objective can still be optimized by SVD and their method can possibly output more than one projecting direction.

LDA and the above Chernoff extension, as well as many other variants such as [4,7], only utilize up to second-order statistics of the class distribution. Peltonen and Kaski [5] recently proposed an alternative approach, *Relevant Component Analysis* (RCA), to find the subspace as informative as possible of the classes. They model the prediction by a generative procedure of classes given the projected values, and the objective is to maximize the log-likelihood of the supervised data. In their method, the predictive probability density is approximated by Parzen estimators. The training procedure requires the user to specify a proper starting learning rate, which however is lacking theoretical instructions and may be difficult in some cases. Moreover, the slow convergence of the stochastic gradient algorithm would may lead to time-consuming learning.

In this paper, we propose an improved method to speed up the RCA training for two-class problems. We employ three strategies for this goal: prewhitening the data, learning the uncorrelated components individually, and optimizing the objective by Newton's method. Finally we obtain a fast *Fixed-Point Relevant Component Analysis* (FPRCA) algorithm such that the optimization convergence is quadratic. The new method inherits the essential advantages of RCA. That is, it can handle distributions more complicated than single Gaussians and extract more than one discriminative component of the data. Furthermore, the user does not need to specify the learning rates because they are optimized by the algorithm.

We start with a brief review of RCA in Section 2. Next, we discuss the objective function of RCA in two-class cases and its fast optimization algorithm in Section 3. Section 4 gives the experiments and comparisons on ionosphere and facial image data. Section 5 concludes the paper.

2 Relevant Component Analysis

Consider a supervised data set which consists of pairs $(\mathbf{x}_j, c_j)$, $j = 1, \ldots, n$, where $\mathbf{x}_j \in \mathbb{R}^m$ is the primary data, and the auxiliary data c_j takes values from binary categorical values. *Relevant Component Analysis* (RCA) [5] seeks a linear $m \times r$ orthonormal projection $\mathbf{W}$ that maximizes the predictive power of the primary data. This is done by constructing a generative probabilistic model of c_j given the projected value $\mathbf{y}_j = \mathbf{W}^T\mathbf{x}_j \in \mathbb{R}^r$ and maximizing the total estimated log-likelihood over $\mathbf{W}$:

$$\underset{\mathbf{W}}{\text{maximize}} \quad J_{\text{RCA}} = \sum_{j=1}^{n} \log \hat{p}(c_j|\mathbf{y}_j). \tag{1}$$

In RCA, the estimated probability $\hat{p}(c_j|\mathbf{y}_j)$ is computed by the definition of conditional probability density function as:

$$\hat{p}(c_j|\mathbf{y}_j) = \frac{\Omega(\mathbf{y}_j, c_j)}{\sum_c \Omega(\mathbf{y}_j, c)}. \tag{2}$$

Here

$$\Omega(\mathbf{y}_j, c) = \frac{1}{n}\sum_{i=1}^{n} \psi(i,c)\omega(\mathbf{y}_i, \mathbf{y}_j) \tag{3}$$

is the Parzen estimation of $\hat{p}(\mathbf{y}_j, c)$ and the membership function $\psi(i,c) = 1$ if $c_i = c$ and 0 otherwise. Gaussian kernel is used in [5] as the Parzen window function

$$\omega(\mathbf{y}_i, \mathbf{y}_j) = \frac{1}{(2\pi\sigma^2)^{r/2}} \exp\left(-\frac{\|\mathbf{y}_i - \mathbf{y}_j\|^2}{2\sigma^2}\right), \tag{4}$$

where σ controls the smoothness of the density estimation.

Peltonen and Kaski [5] derived the gradient of J_{RCA} with respect to $\mathbf{W}$, based on which one can compute the gradients for Givens rotation angles and then update $\mathbf{W}$ for the next iteration. The RCA algorithm applies stochastic gradient optimization method and the iterations converge to a local optimum with a properly decreasing learning rate.

3 Two-Class Discriminant Analysis by RCA with a Fast Fixed-Point Algorithm

3.1 Objective Function

For two-class problems, we point out that not only the maximum, but also the minimum of J_{RCA} optimizes the predictiveness. Peltonen and Kaski has proven the following asymptotical result [5]:

$$\frac{1}{n} J_{\mathrm{RCA}} \underset{n\to\infty}{\longrightarrow} I(C,Y) - E_{p(\mathbf{y})}\left[D_{KL}(p(c|\mathbf{y}), \hat{p}(c|\mathbf{y}))\right] - H(C). \tag{5}$$

The second term is close to zero if one applies a good density approximation and the first term $I(C,Y) = H(C) - H(C|Y)$. Therefore

$$\begin{aligned}\frac{1}{n} J_{\mathrm{RCA}} &\approx -H(C|Y) \\ &= -E_{p(\mathbf{y})}\left\{\sum_c p(c|\mathbf{y}) \log p(c|\mathbf{y})\right\} \\ &\approx -\frac{1}{n}\sum_{j=1}^{n}\sum_c p(c|\mathbf{y}_j) \log p(c|\mathbf{y}_j),\end{aligned} \tag{6}$$

where the last step is obtained by approximating the expectation by sample averaging. The inner summation is recognized as the conditional entropy of the class symbols given $\mathbf{y}_j$. Maximizing J_{RCA} is hence asymptotically equivalent to minimizing the mean uncertainty of prediction at each projected data point. For two-class cases, both the minimum and the maximum of $p(c_j|\mathbf{y}_j)$ lead to the same least entropy at the point $\mathbf{y}_j$, and the same applies to the sum over j.

3.2 Preprocessing the Data

Suppose the data has been centered to be zero mean. Our algorithm requires prewhitening the primary data, i.e. to find an $m \times m$ symmetric matrix $\mathbf{V}$ and to transform $\mathbf{z} = \mathbf{V}\mathbf{x}$ such that $E\{\mathbf{z}\mathbf{z}^T\} = \mathbf{I}$. The matrix $\mathbf{V}$ can be obtained for example by

$$\mathbf{V} = \mathbf{E}\mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T, \tag{7}$$

where $[\mathbf{E}, \mathbf{D}, \mathbf{E}^T] = \text{svd}(E\{\mathbf{x}\mathbf{x}^T\})$ is the singular value decomposition of the scatter matrix of the primary data.

Prewhitening the primary data greatly simplifies the algorithm described in the following section. We can acquire a diagonal approximation of the Hessian matrix and then easily invert it. Another utility of whitening resides in the fact that for two projecting vectors $\mathbf{w}_p$ and $\mathbf{w}_q$,

$$E\{(\mathbf{w}_p^T\mathbf{z})(\mathbf{w}_q^T\mathbf{z})\} = \mathbf{w}_p^T E\{\mathbf{z}\mathbf{z}^T\}\mathbf{w}_q = \mathbf{w}_p^T\mathbf{w}_q, \tag{8}$$

and therefore uncorrelatedness is equivalent to orthogonality. This allows us to individually extract uncorrelated features by orthogonalizing the projecting directions. In addition, selecting the σ parameter in the Gaussian kernel function becomes easier because the whitened data has unit variance on all axes and σ has a readily fixed range for any data sets.

3.3 Optimization Algorithm

Let us first consider the case of a single discriminative component where $\mathbf{y}_j = y_j = \mathbf{w}^T\mathbf{z}_j \in \mathbb{R}$. Our fixed-point algorithm for finding the extreme point of J_{RCA} iteratively applies a Newton's update followed by a normalization:

$$\mathbf{w}^\dagger = \mathbf{w} - \left[\frac{\partial^2 J_{\text{RCA}}}{\partial \mathbf{w}^2}\right]^{-1} \frac{\partial J_{\text{RCA}}}{\partial \mathbf{w}}, \tag{9}$$

$$\mathbf{w}_{\text{new}} = \frac{\mathbf{w}^\dagger}{\|\mathbf{w}^\dagger\|}. \tag{10}$$

Denote $J_j = \log \hat{p}(c_j|y_j)$. The gradient in of J_{RCA} (1) with respect to $\mathbf{w}$ can then be expressed as

$$\frac{\partial J_{\text{RCA}}}{\partial \mathbf{w}} = \sum_{j=1}^{n} \frac{\partial J_j}{\partial \mathbf{w}} = \sum_{j=1}^{n}\sum_{i=1}^{n} \frac{dJ_j}{d(y_i - y_j)} \cdot \frac{\partial(y_i - y_j)}{\partial \mathbf{w}}. \tag{11}$$

Notice that the chain rule in the last step applies to the subscript i, i.e. treating y_i as an intermediate variable and y_j as a constant. We write $g_{ij} = dJ_j/d(y_i - y_j)$ for brevity. If the estimated predictive probability density $\hat{p}(c_j|y_j)$ is obtained by Parzen window technique as in (2) and (3), we can then (see Appendix) write out g_{ij} as

$$g_{ij} = \frac{\psi(i, c_j)\omega'(y_i, y_j)}{\sum_{k=1}^{n} \psi(k, c_j)\omega(y_k, y_j)} - \frac{\omega'(y_i, y_j)}{\sum_{k=1}^{n} \omega(y_k, y_j)}. \tag{12}$$

For notational simplicity, denote

$$\boldsymbol{\Delta}_{ij} = \frac{\partial(y_i - y_j)}{\partial \mathbf{w}} = \mathbf{z}_i - \mathbf{z}_j \tag{13}$$

and the average of all objects (vectors or scalars) $\mathbf{a}_{ij}$, $(i,j) \in [1,\ldots,n]\times[1,\ldots,n]$ as

$$\mathcal{E}\{\mathbf{a}\} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{a}_{ij}. \tag{14}$$

We can then write

$$\frac{\partial J_{\mathrm{RCA}}}{\partial \mathbf{w}} = \sum_{i=1}^{n}\sum_{j=1}^{n} g_{ij}\boldsymbol{\Delta}_{ij} = n^2\mathcal{E}\{\mathbf{g}\circ\boldsymbol{\Delta}\}, \tag{15}$$

where $\circ$ stands for element-wise product and $\boldsymbol{\Delta}$ consists of $n \times n$ vectors of size m.

By taking the derivative of g_{ij} with respect to $y_i - y_j$, we obtain

$$\begin{aligned} g'_{ij} = \frac{\partial g_{ij}}{\partial(y_i - y_j)} =& \frac{\psi(i,c_j)\omega''(y_i,y_j)}{\sum_{k=1}^{n}\psi(k,c_j)\omega(y_k,y_j)} - \frac{\psi(i,c_j)\omega'^2(y_i,y_j)}{\left(\sum_{k=1}^{n}\psi(k,c_j)\omega(y_k,y_j)\right)^2} \\ &- \frac{\omega''(y_i,y_j)}{\sum_{k=1}^{n}\omega(y_k,y_j)} + \frac{\omega'^2(y_i,y_j)}{\left(\sum_{k=1}^{n}\omega(y_k,y_j)\right)^2}. \end{aligned} \tag{16}$$

Based on g'_{ij} one can compute

$$\frac{\partial^2 J_{\mathrm{RCA}}}{\partial \mathbf{w}^2} = n^2\mathcal{E}\{\mathbf{g}'\circ\boldsymbol{\Delta}\boldsymbol{\Delta}^T\}. \tag{17}$$

Notice that $\mathcal{E}\{\boldsymbol{\Delta}\boldsymbol{\Delta}^T\} = 2E\{\mathbf{z}\mathbf{z}^T\} = 2\mathbf{I}$ if the data is centered and prewhitened (see Appendix for a proof). Furthermore, if we approximate $\mathcal{E}\{\mathbf{g}'\circ\boldsymbol{\Delta}\boldsymbol{\Delta}^T\} \approx \mathcal{E}\{\mathbf{g}'\}\mathcal{E}\{\boldsymbol{\Delta}\boldsymbol{\Delta}^T\}$, assuming $\mathbf{g}'$ and $\boldsymbol{\Delta}\boldsymbol{\Delta}^T$ are pair-wisely uncorrelated, the Hessian (17) can be approximated by

$$\frac{\partial^2 J_{\mathrm{RCA}}}{\partial \mathbf{w}^2} = 2n^2\mathcal{E}\{\mathbf{g}'\}\mathbf{I} \tag{18}$$

with $\mathcal{E}\{\mathbf{g}'\} \in \mathbb{R}$. Inserting (15) and (18) into (9), we obtain

$$\mathbf{w}^\dagger = \mathbf{w} - \frac{n^2\mathcal{E}\{\mathbf{g}\circ\boldsymbol{\Delta}\}}{2n^2\mathcal{E}\{\mathbf{g}'\}} = \frac{1}{2\mathcal{E}\{\mathbf{g}'\}}\left(2\mathcal{E}\{\mathbf{g}'\}\mathbf{w} - \mathcal{E}\{\mathbf{g}\circ\boldsymbol{\Delta}\}\right). \tag{19}$$

Because the normalization step (10) is invariant to scaling and the sign of projection does not affect the subspace predictiveness, we can drop the scalar factor in the front and change the order of terms in the parentheses. Then the update rule (9) simplifies to

$$\mathbf{w}^\dagger = \mathcal{E}\{\mathbf{g}\circ\boldsymbol{\Delta}\} - 2\mathcal{E}\{\mathbf{g}'\}\mathbf{w}. \tag{20}$$

In this work we employ a deflationary method to extract multiple discriminative components. Precisely, the *Fixed-Point Relevant Component Analysis* (FPRCA) algorithm comprises the following steps:

1. Center the data to make its mean zero and whiten the data to make its scatter to an identity matrix.
2. Compute $\boldsymbol{\Delta}$, the matrix of pair-wise sample difference vectors as in (13).
3. Choose r, the number of discriminative components to estimate, and σ if the Gaussian kernel (4) is used. Set $p \leftarrow 1$.
4. Initialize $\mathbf{w}_p$ (e.g. randomly).
5. Compute $\mathbf{g}$ and $\mathbf{g}'$ and then update $\mathbf{w}_p \leftarrow \mathcal{E}\{\mathbf{g} \circ \boldsymbol{\Delta}\} - 2\mathcal{E}\{\mathbf{g}'\}\mathbf{w}_p$.
6. Do the following orthogonalization:

$$\mathbf{w}_p \leftarrow \mathbf{w}_p - \sum_{q=1}^{p-1} (\mathbf{w}_p^T \mathbf{w}_q)\mathbf{w}_q. \tag{21}$$

7. Normalize $\mathbf{w}_p \leftarrow \mathbf{w}_p / \|\mathbf{w}_p\|$.
8. If not converged, go back to step 5.
9. Set $p \leftarrow p + 1$. If $p \leq r$, go back to step 4.

4 Experiments

We have tested the FPRCA algorithm on facial images collected under the FERET program [6] and ionosphere data which is available at [1]. The ionosphere data consists of 351 instances, each of which has 34 real numeric attributes. 225 samples are labeled *good* and the other 126 as *bad*. For the FERET data, 2409 frontal facial images (poses "fa" and "fb") of 867 subjects were stored in the database after face segmentation. In this work we obtained the coordinates of the eyes from the ground truth data of the collection, with which we calibrated the head rotation so that all faces are upright. Afterwards, all face boxes were normalized to the size of 32×32, with fixed locations for the left eye (26,9) and the right eye (7,9). Two classes, *mustache* (256 images, 81 subjects) and *no_mustache* (2153 images, 786 subjects), have been used in the following experiments.

4.1 Visualizing Discriminative Features

First we demonstrate the existence of multiple discriminative components in two-class problems. For illustrative purpose, we use two-dimensional projections. The first dimension is obtained from LDA as $\mathbf{w}_1$ and the second is trained by FPRCA as $\mathbf{w}_2$ and orthogonal to $\mathbf{w}_1$ as in (21). All the experiments of FPRCA in this paper use one-dimensional Gaussian kernel (4) with $\sigma = 0.1$ as the Parzen window function.

Figure 1 (a) shows the projected values of the two classes of facial images. The plot illustrates that the vertical $\mathbf{w}_2$ axis provides extra discriminative information in addition to the horizontal $\mathbf{w}_1$ axis computed by the LDA method. It can also be seen that the *mustache* class along the vertical axis comprises two separate clusters. Such projecting direction can by no means be found by LDA and its variants because they limit the projected classes to be single Gaussians.

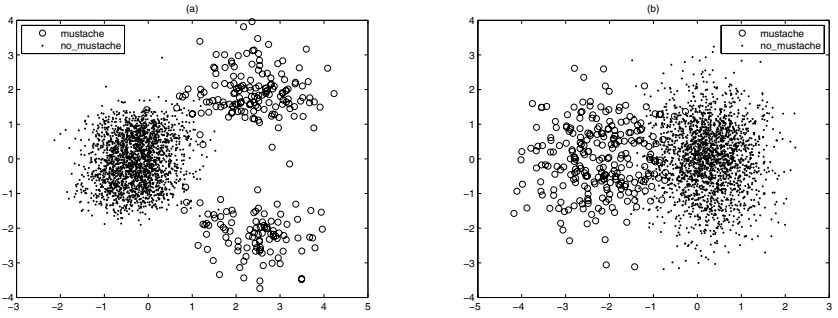


Fig. 1. Projected values of the classes *mustache* and *no_mustache*. (a) The horizontal axis is obtained by LDA and the vertical by FPRCA. (b) Both dimensions are learned by HLDR with the Chernoff criterion [2].

For comparison, the result of HLDR using Chernoff criterion [2] (CHERNOFF) is shown in Figure 1 (b). The first (horizontal) dimension resembles the LDA result, while the second (vertical) provides little discriminative information. It can be seen that the clusters are more overlapping in the latter plot.

4.2 Discriminative Features for Classification

Next we compared the classification results on the ionosphere data using the discriminative features extracted by FPRCA and three other methods: LDA, CHERNOFF [2], and RCA. Two kinds of FPRCA features were used. For $r = 1$, the one-dimensional projection was initialized by LDA and then trained by FPRCA. For $r = 2$, the first component was the training result of $r = 1$ and the additional component was initialized by a random orthogonal vector, and then trained by FPRCA.

The supervised learning and testing were carried out in three modes: ALL – the training set equals the testing set; LOO – leave one instance out for testing and the others for training, and loop for each instance; HALF – half of the samples are for training and the other half for testing. Both LOO and HALF measure the generalization ability. The latter mode is stochastic and tests the performance with a much smaller training set. For it, we repeated the experiment ten times with different random seeds and calculated the mean accuracy.

Figure 2 (a) illustrates the Nearest-Neighbor (NN) classification accuracies for the compared methods in the above three testing modes with the ionosphere data. The left two bars in each group show that FPRCA outperforms LDA in all the three modes when a single discriminative component is used. This verifies that the high-order statistics involved in the information theoretic objective can enhance the discrimination. The right three bars demonstrate the performance of two-dimensional discriminative features. It can be seen that the additional component learned by CHERNOFF even deteriorates the classification from that of LDA. Furthermore, CHERNOFF shows poor generalization when the amount of training data becomes small. In contrast, RCA and FPRCA ($r = 2$) exceed

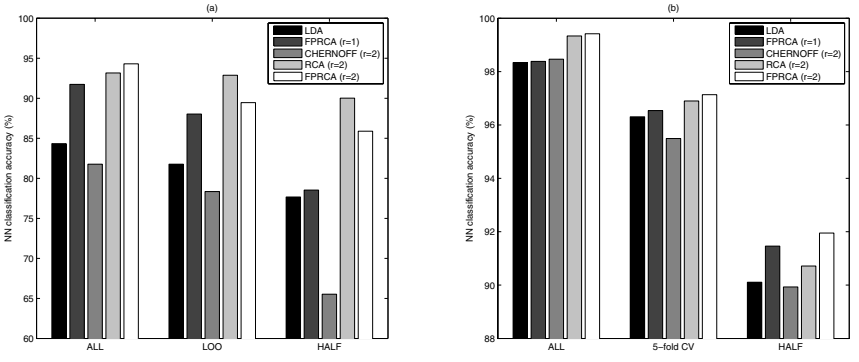


Fig. 2. Nearest-Neighbor (NN) classification accuracies for the compared methods in three testing modes: (a) the ionosphere data and (b) the FERET data

LDA and FPRCA ($r = 1$) with the second component added. The accuracies of FPRCA ($r = 2$) are comparable to those of RCA as the differences are within 3.5% units. RCA performs slightly better than FPRCA ($r = 2$) in the LOO and HALF modes because we applied a grid search for optimal parameters of RCA while we did not for FPRCA.

We also performed the classification experiments on the FERET data. We employed 5-fold cross-validations (CV) instead of LOO because the latter would be very time-consuming for such a large dataset. The data division for HALF and 5-fold CV modes is based on subject identities. That is, all the images of one subject belong either to the training set or to the testing set, never to both.

Figure 2 (b) shows the NN classification accuracies. In the one-dimensional case, FPRCA is again superior to LDA in all modes. The CHERNOFF method ranks better in the ALL mode, but behaves badly and becomes the worst one among the compared methods in the 5-fold CV and HALF modes. By contrast, RCA and FPRCA attain not only the least training errors, but also better generalization. The additional dimension introduced by FPRCA is more advantageous when the training data become scarce. The accuracies of FPRCA ($r = 2$) exceed the other compared methods and the difference is especially significant in the HALF mode. The result for the RCA method may still be suboptimal due to its computational difficulty which will be addressed in the next section.

4.3 RCA vs. FPRCA in Learning Time

We have recorded the running times of RCA and FPRCA using a Linux machine with 12GB RAM and two 64-bit 2.2GHz AMD Opteron processors. For the 34-dimensional ionosphere data, both RCA and FPRCA converged within one minute. The exact running times were 38 and 45 seconds, respectively. However, the calculation is very time-demanding for RCA when it is applied to the 1024-dimensional facial image data. Ten iterations of RCA learning on the FERET database required 598 seconds. A $5,000$-iteration RCA training, which merely utilizes each image roughly twice on the average, took about 83 hours,

i.e. more than three days. In contrast, the FPRCA algorithm converges within 20 iterations for the mustache classification problem and the training time was $4,400$ seconds.

On the other hand, RCA is problematic when a wrong σ or learning rate parameter is selected and one has to re-run the time-consuming procedure. Meanwhile, the user of FPRCA does not need to exhaustively try different parameters because the learning rate is automatically chosen by the algorithm and the range of σ in FPRCA is readily fixed.

5 Conclusions

The objective of maximizing predictive information is known to yield better discriminative power than methods based on only second-order statistics. We presented a fast fixed-point algorithm that efficiently learns the discriminative components of data based on an information theoretic criterion. Prewhitening the primary data facilitates the parameter selection for the Parzen windows and enables approximating the inverse Hessian matrix. The learning rate of each iteration is automatically optimized by the Newton's method, which eases the use of the algorithm. Our method converges quadratically and the extracted discriminative features are advantageous for both visualization and classification.

Like other linear dimensionality reduction methods, FPRCA is readily extended to its kernel version. The nonlinear discriminative components can be obtained by mapping the primary data to a higher-dimensional space with appropriate kernels.

References

1. C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 1998.
2. R.P.W Duin and M. Loog. Linear dimensionality reduction via a heteroscedastic extension of lda: the chernoff criterion. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(6):732–739, 2004.
3. R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7, 1963.
4. Peg Howland and Haesun Park. Generalizing discriminant analysis using the generalized singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):995–1006, 2005.
5. Jaakko Peltonen and Samuel Kaski. Discriminative components of data. *IEEE Transactions on Neural Networks*, 16(1):68–83, 2005.
6. P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET evaluation methodology for face recognition algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:1090–1104, October 2000.
7. Y. Xu, J.Y. Yang, and Z. Jin. Theory analysis on FSLDA and ULDA. *Pattern Recognition*, 36(12):3031–3033, 2003.

A Appendix

First we derive g_{ij} in (12):

$$g_{ij} = \frac{dJ_j}{d(y_i - y_j)} = \frac{\frac{d}{d(y_i-y_j)}\hat{p}(c_j|y_j)}{\hat{p}(c_j|y_j)}, \tag{22}$$

where we have

$$\frac{d\hat{p}(c|y_j)}{d(y_i - y_j)} = \frac{\frac{d}{d(y_i-y_j)}\Omega(y_j, c_j)}{\sum_c \Omega(y_j, c)} - \hat{p}(c_j|y_j)\frac{\sum_c \frac{d}{d(y_i-y_j)}\Omega(y_j, c)}{\sum_c \Omega(y_j, c)}. \tag{23}$$

Inserting (23) and (2) into (22), we obtain

$$\begin{aligned} g_{ij} =& \frac{\frac{d}{d(y_i-y_j)}\Omega(y_j, c_j)}{\Omega(y_j, c_j)} - \frac{\sum_c \frac{d}{d(y_i-y_j)}\Omega(y_j, c)}{\sum_c \Omega(y_j, c)} \\ =& \frac{\psi(i, c_j)\omega'(y_i, y_j)}{\sum_{k=1}^n \psi(k, c_j)\omega(y_k, y_j)} - \frac{(\sum_c \psi(i, c))\omega'(y_i, y_j)}{\sum_{k=1}^n (\sum_c \psi(k, c))\omega(y_k, y_j)}. \end{aligned} \tag{24}$$

$\sum_c \psi(i, c) = 1$ and $\sum_c \psi(k, c) = 1$ if each sample is assigned to only one class. Finally we have (12).

Next we show that $\mathcal{E}\{\boldsymbol{\Delta}\boldsymbol{\Delta}^T\} = 2E\{\mathbf{z}\mathbf{z}^T\}$:

$$\begin{aligned} \mathcal{E}\{\boldsymbol{\Delta}\boldsymbol{\Delta}^T\} =& \frac{1}{n^2}\sum_{i=1}^n\sum_{j=1}^n (\mathbf{z}_i - \mathbf{z}_j)(\mathbf{z}_i - \mathbf{z}_j)^T \\ =& \frac{1}{n}\sum_{i=1}^n \mathbf{z}_i\mathbf{z}_i^T - \frac{1}{n^2}\sum_{i=1}^n \mathbf{z}_i \sum_{j=1}^n \mathbf{z}_j^T - \frac{1}{n^2}\sum_{i=1}^n \mathbf{z}_j \sum_{j=1}^n \mathbf{z}_i^T + \frac{1}{n}\sum_{j=1}^n \mathbf{z}_j\mathbf{z}_j^T \\ =& \frac{2}{n}\sum_{i=1}^n \mathbf{z}_i\mathbf{z}_i^T \qquad (25) \\ =& 2E\{\mathbf{z}\mathbf{z}^T\}, \qquad (26) \end{aligned}$$

where the step (25) is obtained if the primary data is centered, i.e. of zero mean.

Feature Extraction with Weighted Samples Based on Independent Component Analysis

Nojun Kwak

Samsung Electronics, Suwon P.O. Box 105, Suwon-Si, Gyeonggi-Do, Korea 442-742
nojunk@ieee.org
http://csl.snu.ac.kr/∼nojunk

Abstract. This study investigates a new method of feature extraction for classification problems with a considerable amount of outliers. The method is a weighted version of our previous work based on the independent component analysis (ICA). In our previous work, ICA was applied to feature extraction for classification problems by including class information in the training. The resulting features contain much information on the class labels producing good classification performances. However, in many real world classification problems, it is hard to get a clean dataset and inherently, there may exist outliers or dubious data to complicate the learning process resulting in higher rates of misclassification. In addition, it is not unusual to find the samples with the same inputs to have different class labels. In this paper, Parzen window is used to estimate the correctness of the class information of a sample and the resulting class information is used for feature extraction.

1 Introduction

In this paper, the feature extraction for classification problems are dealt with and the focus is on the feature extraction by a linear transform of the original features. These methods are generally referred to as the *subspace methods* which includes principal component analysis (PCA) [1], independent component analysis (ICA) [2], Fisher's linear discriminant analysis (LDA) [3] and so on.

In our previous work, we developed ICA-FX (feature extraction based on independent component analysis) [4], a supervised feature extraction method for classification problems. Like ICA, it utilizes higher order statistics, while unlike ICA, it was developed as a supervised method in that it includes the output class information to find an appropriate feature subspace. This method is well-suited for classification problems in the aspect of constructing new features that are strongly related to output class.

In this paper, the ICA-FX is extended to incorporate the outliers and dubious data in the learning process. For a given training sample, the probability of the sample belonging to a certain class is calculated by Parzen window method [5] and this information is directly used as an input to the ICA-FX. By this preprocessing, the samples with higher class-certainty are enforced and those

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 340–349, 2006.

with lower class-certainty are suppressed in the learning process. The proposed method is applied to an artificial dataset to show effectiveness of the method.

This paper is organized as follows. In Section 2, Parzen window method is briefly reviewed. ICA-FX, our previous feature extraction algorithm, is reviewed in Section 3 and a new method, weighted ICA-FX, is presented in Section 4. Simulation results are presented in Section 5 and conclusions follow in Section 6.

2 A Review of Parzen Window

For a given sample in a dataset, to correctly estimate in what extent the sample belongs to a class, one need to know the *pdf*s of the data. The Parzen window density estimate can be used to approximate the probability density $p(\boldsymbol{x})$ of a vector of continuous random variables $\boldsymbol{X}$ [5]. It involves the superposition of a normalized window function centered on a set of random samples. Given a set of n d-dimensional training vectors $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$, the *pdf* estimate of the Parzen window is given by

$$\hat{p}(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} \phi(\boldsymbol{x} - \boldsymbol{x}_i, h), \tag{1}$$

where $\phi(\cdot)$ is the window function and h is the window width parameter. Parzen showed that $\hat{p}(\boldsymbol{x})$ converges to the true density if $\phi(\cdot)$ and h are selected properly [5]. The window function is required to be a finite-valued non-negative density function such that

$$\int \phi(\boldsymbol{y}, h) d\boldsymbol{y} = 1, \tag{2}$$

and the width parameter is required to be a function of n such that

$$\lim_{n \to \infty} h(n) = 0, \tag{3}$$

and

$$\lim_{n \to \infty} n h^d(n) = \infty. \tag{4}$$

For window functions, the rectangular and the Gaussian window functions are commonly used. In this paper, the Gaussian window function of the following is used:

$$\phi(\boldsymbol{z}, h) = \frac{1}{(2\pi)^{d/2} h^d |\Sigma|^{1/2}} \exp(-\frac{\boldsymbol{z}^T \Sigma^{-1} \boldsymbol{z}}{2h^2}), \tag{5}$$

where Σ is a covariance matrix of a d-dimensional random vector $\boldsymbol{Z}$ whose instance is $\boldsymbol{z}$.

Figure 1 is a typical example of the Parzen window density estimate. In the figure, a Gaussian kernel is placed on top of each data point to produce the density estimate $\hat{p}(x)$.

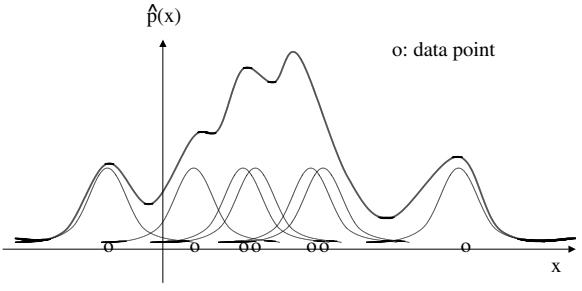


Fig. 1. An example of Parzen window density estimate

3 A Review of ICA-FX

ICA outputs a set of maximally independent vectors that are linear combinations of the observed data. Although these vectors might have some applications in such areas as blind source separation and data visualization, it is not suitable for feature extraction of classification problems, because it is the unsupervised learning that does not use class information. The effort to incorporate the standard ICA with supervised learning has been made in our previous work [4], where a new feature extraction algorithm, ICA-FX for classification problems was proposed. ICA-FX tries to solve the following problem:

(Problem statement). *Assume that there are a normalized input feature vector,* $\boldsymbol{x} = [x_1, \cdots, x_N]^T$*, and an output class,* $c \in \{c_1, \cdots, c_{N_c}\}$*. The purpose of feature extraction is to extract* $M(\leq N)$ *new features* $\boldsymbol{f_a} = [f_1, \cdots, f_M]^T$ *from* $\boldsymbol{x}$*, by a linear combination of the* x_i*'s, containing the maximum information on class c. Here* N_c *is the number of classes.*

The main idea of the ICA-FX is simple. It tries to apply the standard ICA algorithms to feature extraction for classification problems by making use of the class labels to produce two sets of new features; features that carry as much information on the class labels (these features will be useful for classification) as possible and the others that do not (these will be discarded). The advantage is that the general ICA algorithms can be used for feature extraction by maximizing the joint mutual information between the class labels and new features.

First, suppose $N_c(\geq 2)$ denotes the number of classes. To incorporate the class labels in the ICA structure, the discrete class labels need to be encoded into numerical variables. The 1-of-N_c scheme is used in coding classes, i.e., a class vector, $\boldsymbol{c} = [c_1, \cdots, c_{N_c}]^T$, is introduced and if a class label, c, belongs to the lth value, then c_l is activated as 1 and all the other c_i's, $i \neq l$, are set to -1. After all the training examples are presented, each $c_i, i = 1, \cdots, N_c$, is shifted in order to have zero mean and are scaled to have a unit variance.

Now consider the structure shown in Fig. 2. Here, the original feature vector $\boldsymbol{x}$ is fully connected to $\boldsymbol{u} = [u_1, \cdots, u_N]$, the class vector $\boldsymbol{c}$ is connected only

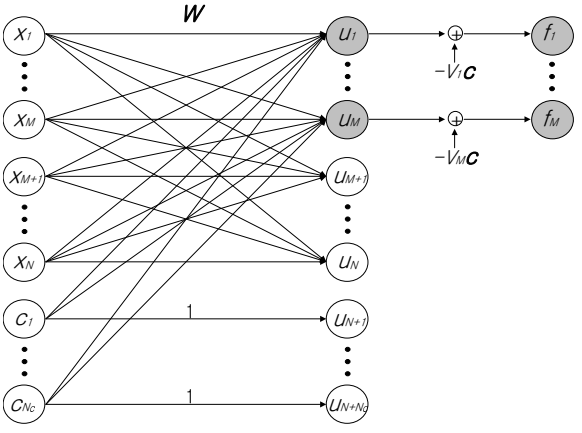


Fig. 2. Feature extraction algorithm based on ICA (ICA-FX)

to $\boldsymbol{u}_a = [u_1, \cdots, u_M]$, and $u_{N+l} = c_l$, $l = 1, \cdots, N_c$. In the figure, the weight matrix $\boldsymbol{W} \in \Re^{(N+N_c)\times(N+N_c)}$ becomes

$$\boldsymbol{W} = \left(\begin{array}{c|c} W & V \\ \hline \mathbf{0}_{N_c,N} & I_{N_c} \end{array}\right) = \left(\begin{array}{c|c} & w_{1,N+1} \quad \cdots \quad w_{1,N+N_c} \\ & \vdots \qquad\qquad \vdots \\ W & w_{M,N+1} \quad \cdots \quad w_{M,N+N_c} \\ & \mathbf{0}_{N-M,N_c} \\ \hline \mathbf{0}_{N_c,N} & I_{N_c} \end{array}\right). \tag{6}$$

where $W \in \Re^{N\times N}$ and $V = [V_a^T, \mathbf{0}_{N-M,N_c}^T]^T \in \Re^{N\times N_c}$. Here the first nonzero M rows of V is denoted as $V_a \in \Re^{M\times N_c}$.

In information theoretic view, the aim of feature extraction is to extract M new features $\boldsymbol{f}_a$ from the original N features, $\boldsymbol{x}$, such that $I(\boldsymbol{f}_a; c)$, the mutual information between newly extracted features $\boldsymbol{f}_a$ and the output class c, approaches $I(\boldsymbol{x}; c)$, the mutual information between the original features $\boldsymbol{x}$ and the output class c [4].

This can be satisfied if we can separate the input feature space $\boldsymbol{x}$ into two linear subspaces: one that is spanned by $\boldsymbol{f}_a = [f_1, \cdots, f_M]^T$, which contains the maximum information on the class label c, and the other spanned by $\boldsymbol{f}_b = [f_{M+1}, \cdots, f_N]^T$, which is independent of c as much as possible.

The condition for this separation can be derived as follows. If it is assumed that $\boldsymbol{W}$ is nonsingular, then $\boldsymbol{x}$ and $\boldsymbol{f} = [f_1, \cdots, f_N]^T$ span the same linear space, which can be represented with the direct sum of $\boldsymbol{f}_a$ and $\boldsymbol{f}_b$, and then by the data processing inequality [6],

$$I(\boldsymbol{x}; c) = I(W\boldsymbol{x}; c) = I(\boldsymbol{f}; c) = I(\boldsymbol{f}_a, \boldsymbol{f}_b; c) \geq I(\boldsymbol{f}_a; c). \tag{7}$$

The first equality holds because W is nonsingular. The second and the third equalities are from the definitions of $\boldsymbol{f}$, $\boldsymbol{f}_a$ and $\boldsymbol{f}_b$. In the inequality on the last line, the equality holds if $I(\boldsymbol{f}_b; c) = I(u_{M+1}, \cdots, u_N; c) = 0$.

If this is possible, the dimension of the input feature space can be reduced from N to $M(< N)$ by using only $\boldsymbol{f}_a$ instead of $\boldsymbol{x}$, without losing any information on the target class.

To solve this problem, the feature extraction problem is interpreted in the structure of the blind source separation (BSS) problem as shown in Fig. 3. The detailed description of each step is as follows:

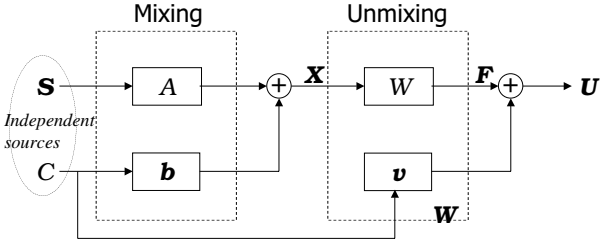


Fig. 3. Interpretation of Feature Extraction in the BSS structure

(Mixing). *Assume that there are N independent sources $\boldsymbol{s} = [s_1, \cdots, s_N]^T$ which are also independent of the class label c. Assume also that the observed feature vector $\boldsymbol{x}$ is a linear combination of the sources $\boldsymbol{s}$ and $\boldsymbol{c}$ with the mixing matrix $A \in \Re^{N \times N}$ and $B \in \Re^{N \times N_c}$; i.e.,*

$$\boldsymbol{x} = A\boldsymbol{s} + B\boldsymbol{c}. \tag{8}$$

(Unmixing). *The unmixing stage is slightly different from the BSS problem as shown in Fig. 2. In the figure, the unmixing equation becomes*

$$\boldsymbol{u} = W\boldsymbol{x} + V\boldsymbol{c}. \tag{9}$$

Suppose $\boldsymbol{u}$ is somehow made equal to $\boldsymbol{e}$, the scaled and permuted version of the source $\boldsymbol{s}$; i.e.,

$$\boldsymbol{e} \triangleq \Lambda\Pi\boldsymbol{s} \tag{10}$$

where Λ is a diagonal matrix corresponding to an appropriate scale and Π is a permutation matrix. The u_i's ($i = 1, \cdots, N$) are then independent of the class label c by the assumption. Among the elements of $\boldsymbol{f} = W\boldsymbol{x} (= \boldsymbol{u} - V\boldsymbol{c})$, $\boldsymbol{f}_b = [f_{M+1}, \cdots, f_N]^T$ will be independent of c because the ith row of V, $V_i = [w_{i,N+1}, \cdots, w_{i,N+N_c}] = \boldsymbol{0}$ and $f_i = u_i$ for $i = M+1, \cdots, N$. Therefore, the $M(< N)$ dimensional new feature vector $\boldsymbol{f}_a$ can be extracted by a linear transformation of $\boldsymbol{x}$ containing the most information on the class if the relation $\boldsymbol{u} = \boldsymbol{e}$ holds.

The learning rule for the ICA-FX is obtained in a similar way as that of ICA using the MLE approach as follows.

If it is assumed that $\boldsymbol{u} = [u_1, \cdots, u_N]^T$ is a linear combination of the source $\boldsymbol{s}$; i.e., it is made equal to $\boldsymbol{e}$, a scaled and permutated version of the source, $\boldsymbol{s}$, as in (10), and that each element of $\boldsymbol{u}$ is independent of the other elements of $\boldsymbol{u}$, which is also independent of the class vector $\boldsymbol{c}$, the log likelihood of the data for a given $\boldsymbol{W}$ becomes the following:

$$L(\boldsymbol{u}, \boldsymbol{c}|\boldsymbol{W}) = \log|\det \boldsymbol{W}| + \sum_{i=1}^{N} \log p_i(u_i) + \log p(\boldsymbol{c}) \tag{11}$$

because

$$p(\boldsymbol{x}, \boldsymbol{c}|\boldsymbol{W}) = |\det \boldsymbol{W}|\ p(\boldsymbol{u}, \boldsymbol{c}) = |\det \boldsymbol{W}| \prod_{i=1}^{N} p_i(u_i)\ p(\boldsymbol{c}). \tag{12}$$

Now, L can be maximized, and this can be achieved by the steepest ascent method. Because the last term in (11) is a constant, differentiating (11) with respect to $\boldsymbol{W}$ leads to

$$\begin{aligned} \frac{\partial L}{\partial w_{i,j}} &= \frac{adj(w_{j,i})}{|\det \boldsymbol{W}|} - \varphi_i(u_i) x_j \qquad 1 \le i, j \le N \\ \frac{\partial L}{\partial w_{i,N+j}} &= -\varphi_i(u_i) c_j \qquad 1 \le i \le M, 1 \le j \le N_c \end{aligned} \tag{13}$$

where $adj(\cdot)$ is adjoint and $\varphi_i(u_i) = -\frac{dp_i(u_i)}{du_i} / p_i(u_i)$. Note that each c_i has binary numerical values depending on the class label c.

It can be seen that $|\det \boldsymbol{W}| = |\det W|$ and $\frac{adj(w_{j,i})}{|\det \boldsymbol{W}|} = W_{i,j}^{-T}$. Thus the learning rule becomes

$$\begin{aligned} \Delta W &\propto\ W^{-T} - \boldsymbol{\varphi}(\boldsymbol{u})\boldsymbol{x}^T \\ \Delta V_a &\propto\ -\boldsymbol{\varphi}(\boldsymbol{u}_a)\boldsymbol{c}^T. \end{aligned} \tag{14}$$

Here $\boldsymbol{\varphi}(\boldsymbol{u}) \triangleq [\varphi_1(u_1), \cdots, \varphi_N(u_N)]^T$ and $\boldsymbol{\varphi}(\boldsymbol{u}_a) \triangleq [\varphi_1(u_1), \cdots, \varphi_M(u_M)]^T$.

Applying a natural gradient on updating W, by multiplying $W^T W$ on the right side of the first equation of (14), the following is obtained.

$$\begin{aligned} W^{(t+1)} &= W^{(t)} + \mu_1 [I_N - \boldsymbol{\varphi}(\boldsymbol{u})\boldsymbol{f}^T] W^{(t)} \\ V_a^{(t+1)} &= V_a^{(t)} - \mu_2 \boldsymbol{\varphi}(\boldsymbol{u}_a)\boldsymbol{c}^T. \end{aligned} \tag{15}$$

Here μ_1 and μ_2 are the learning rates that can be set differently. By this weight update rule, the resulting u_i's will have a good chance of fulfilling the assumption that u_i's are not only independent of one another but also independent of the class label c.

Note that the learning rule for W is the same as the original ICA learning rule [2], and also note that $\boldsymbol{f}_a$ corresponds to the first M elements of $W\boldsymbol{x}$. Therefore, the optimal features $\boldsymbol{f}_a$ can be extracted by the proposed algorithm when it finds the optimal solution for W by (15).

4 Weighted ICA-FX

In ICA-FX presented in the above section, the 1-of-N_c scheme was used to code the discrete class labels into numerical ones, but in many real world problems the same sample may be classified as either one or another class with probability. In addition, the training data may contain incorrect class information resulting errors in classification. This problem may be solved if the probabilistic coding scheme is used for coding the discrete class information into numerical values. That is, suppose there are 3 classes and a training sample says that it belongs to *class 1*. Because the class information of this sample may or may not be correct, instead of using $(1, 0, 0)$ for coding the class of this sample, probabilistic coding such as $(0.7, 0.1, 0.2)$ using the other training data can be used to train ICA-FX. This is done if we know the conditional distribution of classes for a given dataset $p(c|\boldsymbol{x})$.

For this purpose, Parzen window presented in Section 2, is used to estimate the probability that the sample belongs to either *class 1*, *class 2* or *class 3* as follows.

By the Bayesian rule, the conditional probability $p(c|\boldsymbol{x})$ can be written as

$$p(c|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|c)p(c)}{p(\boldsymbol{x})}. \tag{16}$$

If the class has N_c values, say $1, 2, \cdots, N_c$, the estimate of the conditional *pdf* $\hat{p}(\boldsymbol{x}|c)$ of each class is obtained using the Parzen window method as

$$\hat{p}(\boldsymbol{x}|c) = \frac{1}{n_c} \sum_{i \in I_c} \phi(\boldsymbol{x} - \boldsymbol{x}_i, h), \tag{17}$$

where $c = 1, \cdots, N_c$; n_c is the number of the training examples belonging to class c; and I_c is the set of indices of the training examples belonging to class c. Because the summation of the conditional probability equals one, i.e.,

$$\sum_{k=1}^{N_c} p(k|\boldsymbol{x}) = 1,$$

the conditional probability $p(c|\boldsymbol{x})$ is

$$p(c|\boldsymbol{x}) = \frac{p(c|\boldsymbol{x})}{\sum_{k=1}^{N_c} p(k|\boldsymbol{x})} = \frac{p(c)p(\boldsymbol{x}|c)}{\sum_{k=1}^{N_c} p(k)p(\boldsymbol{x}|k)}.$$

The second equality is by the Bayesian rule (16). Using (17), the estimate of the conditional probability becomes

$$\hat{p}(c|\boldsymbol{x}) = \frac{\sum_{i \in I_c} \phi(\boldsymbol{x} - \boldsymbol{x}_i, h_c)}{\sum_{k=1}^{N_c} \sum_{i \in I_k} \phi(\boldsymbol{x} - \boldsymbol{x}_i, h_k)}, \tag{18}$$

where h_c and h_k are the class specific window width parameters. Here $\hat{p}(k) = n_k/n$ is used instead of the true density $p(k)$.

If the Gaussian window function (5) is used with the same window width parameter and the same covariance matrix for each class, (18) becomes

$$\hat{p}(c|\boldsymbol{x}) = \frac{\sum_{i \in I_c} \exp(-\frac{(\boldsymbol{x}-\boldsymbol{x}_i)^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{x}_i)}{2h^2})}{\sum_{k=1}^{N_c} \sum_{i \in I_k} \exp(-\frac{(\boldsymbol{x}-\boldsymbol{x}_i)^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{x}_i)}{2h^2})}. \tag{19}$$

Note that for multi-class classification problems, there may not be enough samples such that the error for the estimate of class specific covariance matrix can be large. Thus, the same covariance matrix is used for each class throughout this paper.

Using $\hat{p}(c|\boldsymbol{x})$ obtained above, the class vector $\boldsymbol{c}$ in Section 3 becomes probabilistic depending on the whole dataset. And this can be used in training ICA-FX directly. The advantage of this coding scheme over 1-of-N_c scheme is that the class information of a sample is affected by its neighboring samples and it becomes more tolerant to outliers. This smoothing process acts as giving more (less) weights on samples whose class information is trustworthy (uncertain). From now on, the proposed algorithm will be referred to as the weighted ICA-FX (wICA-FX).

5 Simulation Results

In this section, the performance of wICA-FX is compared with those of other methods. Consider the simple problem of the following:

Suppose we have two independent input features x_1 and x_2 uniformly distributed on [-0.5,0.5] for a binary classification, and the output class c is determined as follows:

$$c = \begin{cases} 0 & \text{if } x_1 + 3x_2 < 0 \\ 1 & \text{if } x_1 + 3x_2 \geq 0. \end{cases}$$

For this problem, 5 datasets were generated where the class c was randomly flipped with probability of 0 to 0.4. Each dataset contains 500 samples on which PCA, LDA, ICA, ICA-FX and wICA-FX were performed. These feature extraction methods were tested on a separate test dataset with no flip of class information.

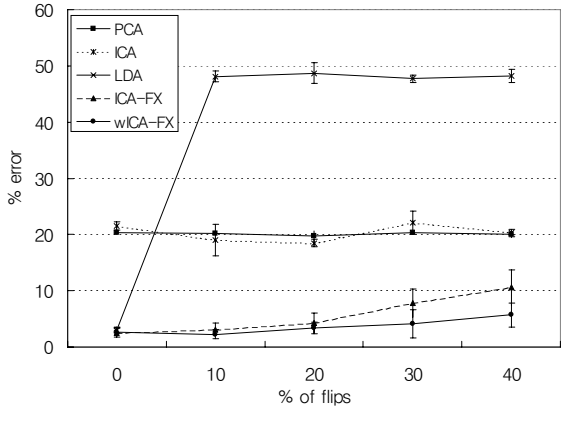
Table 1 is the classification performances of various feature extraction methods on these datasets. One feature is extracted with each method. Averages of 10 experiments with standard deviations are reported here. Standard multi-layer perceptron (MLP) with one hidden layer was used for the classification. Three hidden nodes were used with learning rate of 0.02 and momentum of 0.9. The number of iterations was set to 100. In wICA-FX, h was set to $\frac{1}{\log_{10} n}$ as in [7], where n is the number of training samples.

In the table, the performances of LDA, ICA-FX, and wICA-FX are almost the same when there are no flipped classes. As the number of flipped samples

increases, the error rates of wICA-FX increase more slowly than those of ICA-FX. Comparing to ICA-FX and wICA-FX, the error rates of LDA suddenly jump to 48% when only 10% of the samples are flipped. Note that the error rates of PCA and ICA stays the same around 20 % because these are unsupervised learning methods.

Table 1. Classification performance for the simple dataset (Averages of 10 experiments. Numbers in the parentheses are the standard deviations)

% of flips	Classification error (%) (MLP)				
	PCA	ICA	LDA	ICA-FX	**wICA-FX**
0	20.41	21.53	2.90	2.54	**2.64**
	(0.32)	(0.70)	(0.42)	(0.84)	(0.86)
10	20.22	19.06	48.10	3.16	**2.28**
	(0.28)	(2.82)	(0.98)	(1.15)	(0.74)
20	19.74	18.67	48.71	4.24	**3.42**
	(0.98)	(0.71)	(1.83)	(1.74)	(1.05)
30	20.30	22.18	47.72	7.82	**4.16**
	(0.14)	(2.12)	(0.71)	(2.51)	(2.47)
40	20.02	20.37	48.21	10.56	**5.68**
	(0.56)	(0.70)	(1.13)	(3.21)	(2.09)



6 Conclusions

This study investigates a new method of feature extraction for classification problems with a considerable amount of outliers. In our previous work ICA-FX, class information was added in training ICA. The added class information plays a critical role in the extraction of useful features for classification. With the additional class information we can extract new features containing maximal information about the class. However in many real world classification problems, it is hard to get a clean dataset and inherently, there may exist outliers or dubious

data to complicate the learning process resulting errors in classification. In addition, a sample may be classified as either one or another class with probability. The proposed method focuses on this problem and it is a weighted version of ICA-FX. Parzen window is used to estimate the correctness of the class information of a sample and the resulting class information is used to code the class in ICA-FX. The advantage of this coding scheme over 1-of-N_c scheme is that the class information of a sample is affected by its neighboring samples, thus becomes more tolerant to outliers. This smoothing process acts as giving more (less) weights on samples whose class information is trustworthy (uncertain). Experimental result on the simple artificial dataset shows that the wICA-FX is very effective in dealing with the incorrect class information.

References

1. I.T. Joliffe, *Principal Component Analysis*, Springer-Verlag, 1986.
2. A.J. Bell and T.J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, June 1995.
3. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, second edition, 1990.
4. N. Kwak and C.-H. Choi, "Feature extraction based on ica for binary classification problems," *IEEE Trans. on Knowledge and Data Engineering*, vol. 15, no. 6, pp. 1374–1388, Nov. 2003.
5. E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statistics*, vol. 33, pp. 1065–1076, Sept. 1962.
6. T.M. Cover and J.A. Thomas, *Elements of Information Theory*, John Wiley & Sons, 1991.
7. C. Meilhac and C. Nastar, "Relevance feedback and catagory search in image databases," in *Proc. IEEE Int'l Conf. on Content-based Access of Video and Image databases*, Florence, Italy, June 1999.

Discriminant Analysis by a Neural Network with Mahalanobis Distance

Yoshifusa Ito[1], Cidambi Srinivasan[2], and Hiroyuki Izumi[1]

[1] Department of Policy Science, Aichi-Gakuin University
Nisshin, Aichi-ken, 470-0195 Japan
ito@psis.aichi-gakuin.ac.jp
[2] Department of Statistics, University of Kentucky
Patterson Office Tower, Lexington, Kentucky 40506, USA
srini@ms.uky.edu

Abstract. We propose a neural network which can approximate Mahalanobis discriminant functions after being trained. It can be realized if a Bayesian neural network is equipped with two additional subnetworks. The training is performed sequentially and, hence, the past teacher signals need not be memorized. In this paper, we treat the two-category normal-distribution case. The results of simple simulations are included.

1 Introduction

The goal of this paper is to show that a neural network can be trained so that it can approximate Mahalanobis discriminant functions. Though it is well known that a neural network can be trained to approximate the Bayesian discriminant function [2,4-9], there has been no known result as to the approximation of the Mahalanobis discriminant function by a trained neural network.

We treat the classification of d-dimensional Euclidean vectors by the Mahalanobis discriminant in the context of two-category normal-distribution. The network is realized if a Bayesian neural network is equipped with two additional subnetworks. The Bayesian neural network to be used is the one proposed in [5]. It has one output unit, d hidden layer units, d input layer units, and direct connections between the input layer and the output unit: only $2d+1$ units in total. Except for the direct connections, the neural of this structure is well known. A trained Bayesian neural network can approximate the posterior probability [2,4-9]. This implies that the inner potential of the output unit can approximates the log ratio of posterior probabilities in the two-category case provided the activation function of the output unit of the network is the logistic function [2,4-6]. Both the posterior probability and the log ratio of posterior probabilities can be used as Bayesian discriminant functions [1]. Below, in Preliminaries, we show that the latter differs from one form of the Mahalanobis discriminant function only by an additive constant. This fact is used in the construction of the Mahalanobis neural network.

The constant, as shown in Preliminaries, consists of two parts: one is related to the prior probabilities and the other to the covariance matrices. The purpose

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 350–360, 2006.

of the subnetworks is to approximate the two constants. The former is the log ratio of the prior probabilities. The method of approximating the log ratio is in principle the same as that of approximating the Bayesian discriminant function but considerably simple. So it is realized by a subnetwork having only a single unit.

The other subnetwork approximates the second component, the log ratio of the determinants of the covariance matrices. It is unavoidable for this subnetwork to include a process to approximate the sample means, variances and covariances. Hence, it needs rather many units (at least $d(d+3)$ units). However, learning of these statistics is easy and, to calculate the log ratio from them, learning is unnecessary. Hence, this subnetwork does not cause any difficulty in learning even in the higher dimensional cases.

From the perspective of probabilistic classifications, the Bayesian decision is usually better than the discriminant analysis with the Mahalanobis generalized distance. We also experienced this well known fact in the simulations presented in Section 5. However, the Mahalanobis generalized distance is often used for the discriminant analysis and this analysis includes comparison of measurements. Classification by measurements is often more important than that by probabilities, particularly in the case of health science data. Hence, this article is meaningful and of practical value.

This paper includes the results of simple simulations to illustrate that the proposed algorithm actually works well. The probability distributions used in the simulations are one-dimensional normal distributions.

2 Preliminaries

The two categories are denoted by θ_1 and θ_2 respectively and we set $\Theta = \{\theta_1, \theta_2\}$. Let $\mathbf{R}^d$ be the d-dimensional Euclidean space ($\mathbf{R} = \mathbf{R}^1$) and let $x \in \mathbf{R}^d$ be the vectors to be classified. Denote by $N(\mu_i, \Sigma_i)$, $i = 1, 2$, the normal distributions, where μ_i and Σ_i are the mean vectors and the covariance matrices. They are the distributions of the vectors x from the respective categories. The probability density functions of the normal distributions are

$$\frac{1}{\sqrt{2\pi\Sigma_i}} e^{-\frac{1}{2}(x-\mu_i)^t \Sigma_i^{-1}(x-\mu_i)}, \qquad i = 1, 2. \tag{1}$$

For simplicity, suppose that the covariance matrices are not degenerate. Hence, Σ_i as well as Σ_i^{-1} are positive definite. Let x and y be two vectors. The respective normal distributions define the Mahalanobis generalized distances of the two vectors by

$$d_i(x, y) = |(x-y)^t \Sigma_i^{-1}(x-y)|^{1/2}. \tag{2}$$

The distances from a vector x to the mean vectors of the respective categories are $d_i(x, \mu_i)$, $i = 1, 2$. In the case of the discriminant analysis with the Mahalanobis generalized distances, if $d_1(x, \mu_1) < d_2(x, \mu_2)$ then the vector x is allocated to the category θ_1. Hence, the difference $d_2(x, \mu_2) - d_1(x, \mu_1)$ is a natural discriminant

function for the discriminant analysis. However, the difference of the squares of the respective distances divided by 2

$$g_1(x) = -\frac{1}{2}\{d_1(x,\mu_1)^2 - d_2(x,\mu_2)^2\} \tag{3}$$

is also one form of the Mahalanobis discriminant function. If $g_1(x) > 0$, the vector x is allocated to the category θ_1 and vice versa. By (2), we have

$$g_1(x) = -\frac{1}{2}\{(x-\mu_1)^t \Sigma_1^{-1}(x-\mu_1) - (x-\mu_2)^t \Sigma_2^{-1}(x-\mu_2)\}. \tag{4}$$

In the case of the Bayesian decision, the posterior probabilities are compared. Let $P(\theta_i)$, $i = 1, 2$, be the prior probabilities and let $p(x|\theta_i)$, $i = 1, 2$, be the state-conditional probabilities. We set $p(x) = P(\theta_1)p(x|\theta_1) + P(\theta_2)p(x|\theta_2)$. In the two-category case, one of the posterier probabilities, say $p(\theta_1|x)$, as well as the difference $p(\theta_1|x) - p(\theta_2|x)$ can be used as the Bayesian discriminant function [1]. Furthermore, the ratio $P(\theta_1|x)/P(\theta_2|x)$ of the posterior probabilities can also be a Bayesian discriminant function. Since a monotone transform of a discriminant function is again a discriminant function [1],

$$g_2(x) = \log \frac{P(\theta_1|x)}{P(\theta_2|x)} \tag{5}$$

can be used as the Bayesian discriminant function. If $g_2(x) > 0$, x is allocated to the category θ_1. Though the functions (3) and (5) are based on different ideas, they differ only by a constant. By the Bayes formula, $P(\theta_i|x) = P(\theta_i)p(x|\theta_i)/p(x)$, we have

$$g_2(x) = \log \frac{P(\theta_1)}{P(\theta_2)} + \log \frac{p(x|\theta_1)}{p(x|\theta_2)}. \tag{6}$$

Hence, by (1), we have

$$g_2(x) = -\frac{1}{2}\{(x-\mu_1)^t \Sigma_1^{-1}(x-\mu_1) - (x-\mu_2)^t \Sigma_2^{-1}(x-\mu_2)\} + \log \frac{P(\theta_1)}{P(\theta_2)} - \frac{1}{2}\log \frac{|\Sigma_1|}{|\Sigma_2|}. \tag{7}$$

Consequently, by subtracting a constant

$$C = \log \frac{P(\theta_1)}{P(\theta_2)} - \frac{1}{2}\log \frac{|\Sigma_1|}{|\Sigma_2|}, \tag{8}$$

from the Bayesian discriminant function $g_2(x)$, we obtain the Mahalanobis discriminant function $g_1(x)$.

3 Bayesian Neural Network

The main part of the Mahalnobis neural network is a Bayesian neural network. It is used to approximate the log ratio (5). Furthermore, a simplified version of the Bayesian neural network is used to approximate the log ratio, $\log P(\theta_1)/P(\theta_2)$, the first term of the constant (8). To provide the necessary back ground, we summarize here the Bayesian neural network for the two-category case.

The logistic function σ is defined by

$$\sigma(t) = \frac{1}{1+e^{-t}}.$$

Since a monotone transform of a discriminant function is again a discriminant function [1],

$$\sigma(g_2(x)) = \sigma\left(\log \frac{P(\theta_1|x)}{P(\theta_2|x)}\right) \tag{9}$$

is also a Bayesian discriminant functions. As

$$\sigma\left(\log \frac{P(\theta_1|x)}{P(\theta_2|x)}\right) = \frac{P(\theta_1|x)}{P(\theta_1|x)+P(\theta_2|x)}, \tag{10}$$

we have

$$\sigma(g_2(x)) = P(\theta_1|x). \tag{11}$$

The inner potential of the output unit of the Bayesian neural network, proposed in [5], having d hidden layer units can approximate any quadratic form in $\mathbf{R}^d$ and, hence, $\log P(\theta_1|x)/P(\theta_2|x)$ in the sense of $L^p(\mathbf{R}^d, \mu)$. Here, we omit the proof but it is based on the theory of quadratic forms and Lemma 1 [3] below.

Lemma 1. Let μ be a probability measure on $\mathbf{R}$. If $t^n \in L^p(\mathbf{R}, \mu)$, $0 \le p < \infty$, $\phi \in C^n(\mathbf{R})$ and $\phi^{(k)}$, $0 \le k \le n$, are bounded, then, for any $\varepsilon > 0$, there exists a constant δ for which

$$\|\frac{1}{n!}\phi^{(n)}(0)t^n - \frac{1}{\delta^n}\phi(\delta t) - \sum_{i=0}^{n-1}\frac{1}{i!}\phi^{(i)}(0)(\delta t)^i\|_{L^p(\mathbf{R},\mu)} < \varepsilon. \tag{12}$$

Of course, a quadratic form cannot be approximated on the whole space $\mathbf{R}^d$ uniformly by a finite sum of activation functions such as the logistic function. However, if the probability measure μ decreases more rapidly than the quadratic form and the activation function increase, then their contribution to the $L^p(\mathbf{R}^d, \mu)$ norm outside a bounded domain is small, and, hence, the approximation can be achieved.

The activation function of the output unit of the Bayesian neural network is the logistic function. Hence, if the inner potential of the output unit approximates the log ratio, $\log P(\theta_1|x)/P(\theta_2|x)$, in the sense of $L^p(\mathbf{R}^d, \mu)$, then, by (11), the output approximates the posterior probability $P(\theta_1|x)$. This approximation also holds in the sense of $L^p(\mathbf{R}^d, \mu)$, because the logistic transform is a contraction. Conversely, if the output approximates $P(\theta_1|x)$, the inner potential of the output unit has no other choice other than approximating $\log P(\theta_1|x)/P(\theta_2|x)$.

Let $F(x, w)$ denote the output of the neural network with weight vector w. For an integrable function $\xi(x, \theta)$ defined on $\mathbf{R}^d \times \Theta$, let $E[\xi(x, \cdot)|x]$ and $V[\xi(x, \cdot)|x]$ be the conditional expectation and variance of $\xi(x, \theta)$. The following proposition is proved in [8]:

Proposition 2. Set

$$E(w) = \int_{\mathbf{R}^d} \sum_{i=1}^{2} (F(x, w) - \xi(x, \theta_i))^2 P(\theta_i) p(x|\theta_i) dx. \tag{13}$$

Then,

$$E(w) = \int_{\mathbf{R}^d} (F(x, w) - E[\xi(x, \cdot)|x])^2 p(x) dx + \int_{\mathbf{R}^d} V[\xi(x, \cdot)|x] p(x) dx. \tag{14}$$

If $\xi(x, \theta_1) = 1$ and $\xi(x, \theta_2) = 0$, then $E[\xi(x, \cdot)|x]$ is equal to the posterior probability $P(\theta_1|x)$. Hence, when $E(w)$ is minimized, the output $F(x, w)$ is expected to approximate $P(\theta_1|x)$. Accordingly, the network learning is carried out by minimizing

$$E_n(w) = \frac{1}{n} \sum_{t=1}^{n} (F(x^{(k)}, w) - \xi(x^{(k)}, \theta^{(k)}))^2, \tag{15}$$

where $\{(x^{(k)}, \theta^{(k)})\}_{k=1}^{n} \subset \mathbf{R}^d \times \Theta$ is the training set. Minimization of (13) can be realized by sequential learning. This method of training has actually been stated by many authors [2,4-9].

4 Mahalanobis Discriminant Function

In this section we discuss how to construct a neural network which can approximate the Mahalanobis discriminant function $g_1(x)$. The starting point of this section is an equation

$$g_1(x) = g_2(x) - \log \frac{P(\theta_1)}{P(\theta_2)} + \frac{1}{2} \log \frac{|\Sigma_1|}{|\Sigma_2|}, \tag{16}$$

which can be obtained from (4) and (7). The first term on the right hand side of (16) can be approximated by the inner potential of the output unit of the Bayesian network as stated. As the second and third terms are constant in x, it may not be difficult to approximate them. If the network can perform this task, the discriminant function $g_1(x)$ can be realized as their algebraic sum.

To approximate the second term, we use a single unit network having the logistic function as the activation function. It has no input. Hence, its inner potential is the bias itself. Though its structure is simple, its learning rule is

in principle the same as that of the Bayesian network. Let $\zeta(\theta)$ be a function defined on Θ. Then, for

$$e(v) = \int_{\mathbf{R}^d} \sum_{i=1}^{2} (v - \zeta(\theta_i))^2 P(\theta_i) p(x|\theta_i) dx, \tag{17}$$

we have

$$e(v) = (v - E[\zeta(\cdot)])^2 + V[\zeta(\cdot)], \tag{18}$$

where $E[\zeta(\cdot)]$ and $V[\zeta(\cdot)]$ are respectively the expectation and variance of $\zeta(\cdot)$. This is a simplification of Proposition 2. If (17) is minimized with respect to v, then the minimizing v approximates $E[\zeta(\cdot)]$. Moreover, when $\zeta(\theta_1) = 1$ and $\zeta(\theta_2) = 0$, $E[\zeta(\cdot)] = P(\theta_1)$. The equations (17) and (18) correspond to (13) and (14) respectively. By the method of least squares, (17) can be easily minimized and, hence, the output v can approximate $P(\theta_1)$. Then, by the same reason as in the case of the Bayesian neural network, the inner potential approximates the log ratio, $\log P(\theta_1)/P(\theta_2)$.

The approximation of the log ratio of the determinants $|\Sigma_1|$ and $|\Sigma_2|$, the third term on the right-hand side of (16), is achieved by a different method. In this case the network does not need to approximate probabilistic function. Hence, learning is easy. The sample means, variances and covariances can be approximated by the method of least squares, which can be realized by sequential learning. To calculate the log ratio, $\log |\Sigma_1|/|\Sigma_2|$, from these statistics, learning is unnecessary. The module for this calculation can be fixed beforehand. Hence, no difficulty is expected in learning of the third term on the right-hand side of (16).

There may be another direct method. Let n_i be the number of pairs $(x^{(k)}, \theta^{(k)})$ from the category θ_i up to time n. Define determinants by

$$|S_i| = |\frac{1}{n_i} \sum_{k=1}^{n} (x_{ir}^{(k)} - m_{ir})(x_{is}^{(k)} - m_{is})|_{r,s=1,\cdots,d}, \qquad i = 1, 2, \tag{19}$$

where $x_i = (x_{i1}, \cdots, x_{id})$ are the vectors from the category θ_i. If m_i are the sample mean vectors, then, $|S_i|$ are the determinants of sample covariance matrices. If m_i are variables, then (19) are functions in m_i respectively. We can prove that when m_i is equal to the respective sample mean vectors $\hat{\mu}_i$, the gradients of the functions (19) are null and the Hessians of (19) are positive definite. This implies that the determinants can be obtained by the method of least squares with the gradient descent.

In conclusion, each term of the right-hand side of (16) can be approximated by sequential learning respectively. Hence, the past teacher signals are unnecessary for updating these terms. We can realize the Mahalanobis neural network, putting together these results. The inner potentials of the output units of the main part and one of the subnetworks are fed into a linear unit with the output of the other subnetwork. Then, the output of the linear unit approximates the Mahalanobis discriminant function.

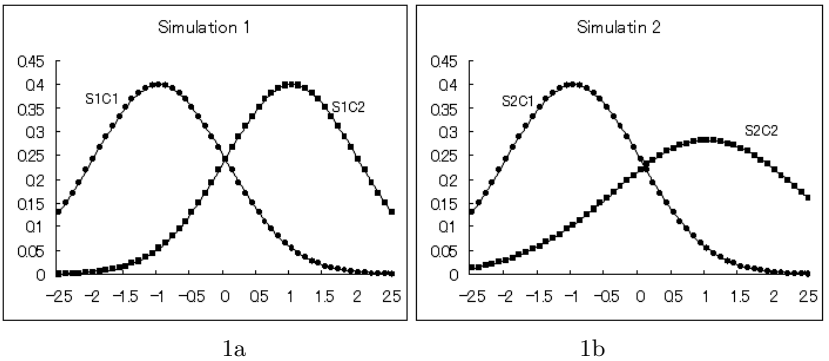
5 Simulations

Simple simulations are performed to confirm that the idea of this article works well. In each simulation, 1000 numbers are randomly chosen from two one-dimensional normal distributions. They are not vectors as $d = 1$. The two normal distributions are denoted by $N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$, where σ_i^2, $i = 1, 2$, are the variances. The results of two simulations are illustrated in this section. The prior probabilities, the means and variances in the two simulations are listed in Table 1. In this table, only the variances are distinct in the two simulations. In Simulation 1, the inner potential of the output unit is to approximate a linear function but, in Simulation 2, it has to approximate a quadratic function. Hence, the approximation tasks in the two simulations are distinct in difficulty. However, the distributions of the categories θ_1 are common to compare the two results conveniently.

Table 1. Parameters of the probability distributions in the respective simulations

	$P(\theta_1)$	$P(\theta_2)$	μ_1	μ_2	σ_1^2	σ_2^2
Simulation 1	0.3	0.7	-1	1	1	1
Simulation 2	0.3	0.7	-1	1	1	2

The probability density functions of the two categories in each simulation are illustrated in Figures 1a and 1b respectively. The curve S1C1, for example, is the probability density function used in the simulation 1 for the category θ_1.



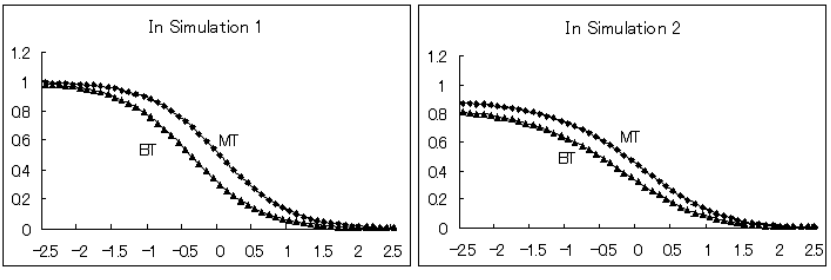
1a 1b

Fig 1. Probability density functions in the respective simulations

The Bayesian and Mahalanobis discriminant functions obtained theoretically with the parameters in Table 1 in the respective simulations are compared in

Figures 2a and 2b. The curves BT illustrate the Bayesian discriminant functions and the curves MT the Mahalanobis discriminant functions obtained theoretically.

The training sequences are cyclic. First, a sequence of 1000 pairs $(x,\ \theta) \in \mathbf{R}^d \times \Theta$ are generated using 2000 random numbers in each simulation. The random numbers used in both simulations are the same. Hence, the sequences $\{\theta^{(k)}\}$ are common in the two simulations. When $\theta^{(k)} = \theta_i$, $x^{(k)}$ is chosen from the distribution $N(m_i, \sigma_i^2)$. Hence, when $\theta^{(k)} = \theta_1$, the pairs $(x^{(k)},\ \theta^{(k)})$ are the same in the two simulations. These are convenient, when we compare the results in the two simulations. These sequences are repeatedly used respectively. Hence, the training sequences are cyclic. Using cyclic training sequences has a merit in that the sample statistics can be exactly calculated and compared with the results of simulations.



2a 2b

Fig 2. Theoretically obtained discriminant functions

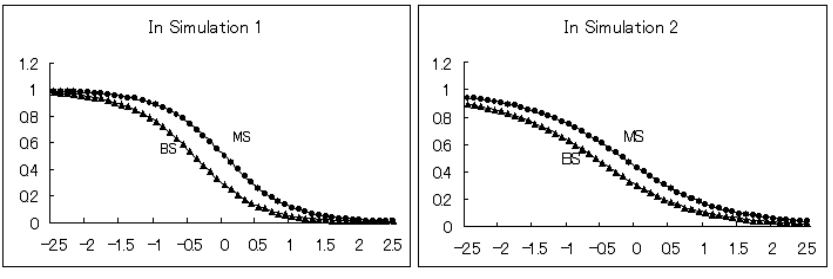
The sequences included 289 pairs from the category θ_1 and 711 pairs from the category θ_2. Let $\hat{\mu}_i$ be the sample means. In the case $d = 1$, the determinants $|\Sigma_i|$ are the sample variances $\hat{\sigma}_i^2$. These sample statistics and the approximation $\hat{P}(\theta_1)$ of the prior probability are listed in Table 2 with $\hat{P}(\theta_2) = 1 - \hat{P}(\theta_1)$. The networks obtained these quantities with accuracy of several significant digits. The log ratio of the variance, $\log \hat{\sigma}_1^2/\hat{\sigma}_2^2$, was calculated outside the network in our simulation. However, it can be obtained by the subnetwork if it has log units.

Table 2. Parameters of the probability distributions based on the outputs of the trained Mahalanobis neural networks

	$\hat{P}(\theta_1)$	$\hat{P}(\theta_2)$	$\hat{\mu}_1$	$\hat{\mu}_2$	$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$
Simulation 1	0.289	0.711	-0.977	1.049	1.036	0.991
Simulation 2	0.289	0.711	-0.977	1.070	1.036	1.982

When the two tables are compared, one may find slight differences between the corresponding values. These differences are caused by randomness of the

training set. In Figures 3a and 3b, the Bayesian and Mahalanobis discriminant functions obtained by the respective simulations are shown. The curves BS illustrate the Bayesian discriminant functions and the curves MS the Mahalanobis discriminant functions obtained by simulation. The formers are the outputs of the trained Bayesian neural networks and the latters are those of the trained Mahalanobis neural networks. They are obtained by shifting the inner potentials of the output units of the formers. The sizes of the shifts are decided based on Table 2 which is a list of statistics obtained by the networks themselves. Note that they are not based on the parameters given in Table 1. Using these four discriminant functions the test data were classified. The test sequences were constructed from 2000 random numbers distinct from those used for generating the teacher sequences. The test sequences were constructed in the same way as the teacher sequences. Hence, the sequences of the categories $\{\theta^{(k)}\}$ are the same in both simulations.



3a 3b

Fig. 3. Discriminant functions obtained by simulations

The test sequences contained 298 pairs from the category θ_1 and 702 pairs from the category θ_2. The numbers of correctly classified x among the 1000 are listed in Table 3. The numbers in the MT, MS, BT and BS columns in Table 3 are respectively the classification results by the Mahalanobis dicriminant functions obtained theoretically (MT) and by simulation (MS), and those by the Bayesian discriminant functions obtained theoretically (BT) and by simulation (BS). The capabilities of the discriminant functions obtained by simulation are comparable to the corresponding theoretical discriminant functions respectively.

Table 3. Classification results by the four discriminant functions. See Text.

	MT	MS	BT	BS
Simulation 1	825	825	839	838
Simulation 2	772	773	796	799

6 Discussions

The theoretically proposed learning algorithm of the neural network for the Mahalanobis discriminant function was proved to work at least in the simple cases. The simulation results suggest its usefulness for practical applications. Since the training of the network is sequential, the network does not need to memorize the individual past teacher signals. It is well-known that the Bayesian decision is generally better than any other methods of decision. We have unexpectedly experienced this fact by comparing the Bayesian and Mahalanobis decisions with the discriminant functions obtained theoretically and by simulation (Table 3). However, the discriminant analysis by the Mahalanobis generalized distance is also often used in many occasions. In such cases, our network may be useful.

The main part of the neural network has d hidden layer units. This number is about a half of that of the Bayesian neural network proposed by Funahashi [2]. However, our network has met some difficulty in training when the discriminant function is not very simple. Even in Simulation 2 in this paper, the approximation of the discriminant function was not very accurate, though the trained neural network worked well as listed in Table 3. If the initial values of parameters are adjusted, the accuracy of the approximation may be improved. It can be theoretically proved that our network can approximate the discriminant function with any accuracy. But the accurate approximation is realized in the world of "δ and ε". The free learning may rarely step into such a world. We have just occasionally observed that the absolute value of the coefficient of x, the constant corresponding to δ in Lemma 1, is minimized to less than 0.001.

The log ratios, $\log P(\theta_1)/P(\theta_2)$ and $\log p(\theta_1|x)/p(\theta_2|x)(= g_2(x))$, can be directly approximated as the inner potentials of the output units by the method of least squares with respect to the prior probability $P(\theta_1)$ and postrior probability $p(\theta_1|x)$ respectively. However, the approximation of the log ratio, $\log|\Sigma_1|/|\Sigma_2|$, of the determinants may not be achieved in such a way. There may be two ways of approximating the determinants. One is to approximate the sample means, variances and covariances respectively using a neural network simple but having many output units. The determinants can be calculated from these statistics. If this is a task of the network, the module for this calculation must be complicated. However, the module can be fixed beforehand and does not cause any difficulty in learning. The other way is to use the fact that the Hessian of (19) is positive definite as described in Section 4. In this case the determinants can be obtained directly but the structure of the network must be complicated. Moreover, unless the network has log units, it is also a complicated task to obtain their log ratio. A reasonable algorithm for the log ratio of the determinants is still under our consideration.

References

1. R.O. Duda and P.E. Hart, Pattern classification and scene analysis, Joh Wiley & Sons, New York, 1973.
2. Funahashi, K., Multilayer neural networks and Bayes decision theory, Neural Networks, 11, 209-213, 1998.

3. Y. Ito, Simultaneous L^p -approximations of polynomials and derivatives on $\mathbf{R}^d$ and their applications to neural networks, (in preparation)
4. Y. Ito and C. Srinivasan. Multicategory Bayesian decision using a three-layer neural network, in Proceedings of ICANN/ICONIP 2003, 253-261, 2003.
5. Y. Ito and C. Srinivasan. Bayesian decision theory on three-layer neural networks, Neurocomputing, vol. 63, 209-228, 2005.
6. Y. Ito, C. Srinivasan and H. Izumi. Bayesian learning of neural networks adapted to changes of prior probabilities, in Proceedings of ICANN 2005,
7. M.D.Richard and R.P. Lipmann, Neural network classifiers estimate Bayesian a posteriori probabilities, Neural Computation, vol. 3, pp461-483, 1991.
8. M.D.Ruck, S. Rogers, M. Kabrisky, H. Oxley, B. Sutter, The multilayer perceptron as approximator to a Bayes optimal discriminant function, IEEE Transactions on Neural Networks, vol. 1, pp296-298, 1990.
9. H. White Learning in artificial neural networks: A statistical perseptive. Neural Computation, vol. 1, pp425-464, 1989.

Assessment of an Unsupervised Feature Selection Method for Generative Topographic Mapping

Alfredo Vellido

Department of Computing Languages and Systems (LSI). Polytechnic University of Catalonia (UPC). C. Jordi Girona, 1-3. 08034, Barcelona, Spain
avellido@lsi.upc.edu

Abstract. Feature selection (FS) has long been studied in classification and regression problems. In comparison, FS for unsupervised learning has received far less attention. For many real problems concerning unsupervised data clustering, FS becomes an issue of paramount importance. An unsupervised FS method for Gaussian Mixture Models, based on Feature Relevance Determination (FRD), was recently defined. Unfortunately, the data visualization capabilities of general mixture models are limited. Generative Topographic Mapping (GTM), a constrained mixture model, was originally defined to overcome such limitation. In this brief study, we test in some detail the capabilities of a recently described FRD method for GTM that allows the clustering results to be intuitively visualized and interpreted in terms of a reduced subset of selected relevant features.

1 Introduction

Finite mixture models have settled in recent years as a standard for statistical modelling [1]. Gaussian Mixture Models (GMM), in particular, have received especial attention for their computational convenience to deal with multivariate continuous data. This study focuses on their clustering capabilities.

Multivariate data visualization can be especially important in the exploratory stages of an analytical data mining process [2], and GMMs lack this capability. The GTM model was originally defined [3] as a constrained GMM allowing for multivariate data visualization on a low dimensional space. The model is constrained in that mixture components are equally weighted, share a common variance and their centres do not move independently from each other. This last feature also makes GTM a probabilistic alternative to the widely used Self-Organizing Maps [4].

The interpretability of the clustering results provided by GTM, even in terms of exploratory visualization, can be reduced when the data sets under analysis consist of a large number of features: a situation that is not uncommon in real clustering problems. The data analyst would benefit from a method that allowed ranking the features according to their relative relevance and, ultimately, from a feature selection method based on it. Feature selection (FS) has for long been the preserve of supervised methods and, in comparison, FS for unsupervised learning has received far

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 361 – 370, 2006.

less attention despite the fact that, in many real clustering problems, FS becomes an issue of paramount importance, as results have to meet interpretability and actionability requirements. Both the interpretability and the actionability (understood as the capability to act upon the clustering results) of clusters would be improved by their description in terms of a reduced subset of relevant variables.

A recent main advance on feature selection in unsupervised model-based clustering was presented in [5] for GMM and extended to GTM in [6] for a biomedical problem, but it has never been evaluated in detail. This brief study provides such evaluation.

The remaining of the paper is structured as follows. First, brief introductions to the standard Gaussian GTM and its extension for FRD are provided. This method is then tested on several artificial and real data sets and the results are presented and discussed. The paper wraps up with a brief conclusions section.

2 GTM as a Constrained Gaussian Mixture Model

In mixture models, the observed data are assumed to be samples of a combination or finite mixture of k=1,…,K components or underlying distributions, weighted by unknown priors $P(k)$. Given a D-dimensional dataset $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^{N}$, consisting of N random observations, the corresponding mixture density is defined as:

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(\mathbf{x}|k;\theta_k)P(k), \tag{1}$$

where each mixture component k is parameterized by θ_k. For continuous data, the choice of Gaussian distributions is a rather straightforward option, in which case:

$$p(\mathbf{x}|k;\mu_k,\boldsymbol{\Sigma}_k) = (2\pi)^{-D/2}|\boldsymbol{\Sigma}_k|^{-1/2}\exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\mu_k)\right\}, \tag{2}$$

where the adaptive parameters θ_k are the mean vector and the covariance matrix of the D-variate distribution for each mixture component, namely μ_k and $\boldsymbol{\Sigma}_k$. From Eq. (2), a log-likelihood can be defined, from which Maximum Likelihood estimates of μ_k and $\boldsymbol{\Sigma}_k$ can be obtained using the Expectation-Maximization (EM:[7]) algorithm.

2.1 The Standard GTM Model

One of the practical limitations of general finite mixture models is their lack of visualization capabilities, which reduces the interpretability of the model. The GTM was defined as a constrained mixture of distributions in order to provide such visualization capabilities, akin to those of the widely used SOM. The GTM is a constrained mixture of distributions model in the sense that all the components of the mixture are equally weighted by the constant term $P(k) = K^{-1}$, and all components share a common variance β^{-1} (therefore $\boldsymbol{\Sigma} = \beta^{-1}\mathbf{I}$). The GTM can also be seen as a non-linear latent variable model that defines a mapping from a low dimensional latent

space onto the high-dimensional data space. As such, it is further constrained in that the centres of the mixture components do not move independently from each other, as they are limited by definition to lie on a low-dimensional manifold embedded in the *D*-dimensional space. Such manifold constraint is made explicit through the definition of a prior distribution in the latent space in the form $p(\mathbf{u}) = K^{-1}\sum_{k=1}^{K}\delta(\mathbf{u}-\mathbf{u}_k)$, where the *K* latent points $\mathbf{u}_k$ are sampled, forming a regular grid, from the latent space of the GTM. This latent space discretization makes the model computationally tractable and provides an alternative to the clustering and visualization space of the SOM.

The mapping defined by the model is carried through by a set of basis functions generating a (mixture) density distribution. For each feature *d*, the functional form of this mapping is the generalized linear regression model $y_d(\mathbf{u},\mathbf{W}) = \sum_m^M \phi_m(\mathbf{u}) w_{md}$, where ϕ_m is one of *M* basis functions, defined here as spherically symmetric Gaussians, and $\mathbf{W}$ is the matrix of adaptive weights w_{md} that defines the mapping. The probability distribution for a data point $\mathbf{x}$, induced by $p(\mathbf{u})$ and given the adaptive parameters of the model, which are the matrix $\mathbf{W}$ and the common inverse variance of the Gaussians β, can be written as:

$$p(\mathbf{x}|\mathbf{u};\mathbf{W},\beta) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\beta/2 \|\mathbf{y}-\mathbf{x}\|^2\right\}, \tag{3}$$

where the *D* elements of $\mathbf{y}$ are given by the aforementioned functional form of the mapping. Integrating the latent variables out, we obtain the following mixture density:

$$p(\mathbf{x}|\mathbf{W},\beta) = \int p(\mathbf{x}|\mathbf{u};\mathbf{W},\beta)p(\mathbf{u})d\mathbf{u} = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\beta/2 \|\mathbf{y}_k-\mathbf{x}\|^2\right\} \tag{4}$$

where the *D*-dimensional centres of the GTM mixture components $\mathbf{y}_k$ are usually known as *reference vectors* or *prototypes* of a cluster. This leads to the definition of the log-likelihood as:

$$L(\mathbf{W},\beta|\mathbf{X}) = \sum_{n=1}^{N} \ln\left\{\frac{1}{K}\sum_{k=1}^{K}\left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\beta/2 \|\mathbf{y}_k-\mathbf{x}_n\|^2\right\}\right\} \tag{5}$$

The EM algorithm can then be used to obtain the Maximum Likelihood estimates of the adaptive parameters of the model. Defining $\mathbf{Z}$ as the matrix of indicators describing our lack of knowledge of which latent point $\mathbf{u}_k$ is responsible for the model generation of data point $\mathbf{x}_n$, the complete log-likelihood becomes:

$$L_c(\mathbf{W},\beta|\mathbf{X},\mathbf{Z}) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{kn} \ln\left[\left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\beta/2 \|\mathbf{y}_k-\mathbf{x}_n\|^2\right\}\right]. \tag{6}$$

The expected value of z_{kn} (or *responsibility* r_{kn}) is calculated in the E-step of EM as:

$$r_{kn} = P(\mathbf{u}_k|\mathbf{x}_n;\mathbf{W},\beta) = \frac{\exp\left\{-\frac{\beta}{2}\|\mathbf{y}_k - \mathbf{x}_n\|^2\right\}}{\sum_{k'=1}^{K}\exp\left\{-\frac{\beta}{2}\|\mathbf{y}_{k'} - \mathbf{x}_n\|^2\right\}}, \quad (7)$$

whereas the update expressions for $\mathbf{W}$ and β are obtained in the Maximization step of the algorithm. See details of these calculations in [3].

2.2 Feature Relevance Determination in GTM: The FRD-GTM

Despite having been defined to provide multivariate data exploratory visualization, the interpretability of the clustering results provided by the GTM can be limited for data sets of large dimensionality. Consequently, an unsupervised FRD method should help to improve model interpretability.

The problem of feature relative relevance determination for GMM was recently addressed in [5] and extended to GTM in [6]. Feature relevance in this unsupervised setting is understood as the likelihood of a feature being responsible for generating the data clustering structure. A similar counterpart procedure for supervised models is Automatic Relevance Determination (ARD: [8]).

In this unsupervised setting, relevance is defined through the concept of saliency. Formally, the saliency of feature d can be defined as $\rho_d = P(\eta_d = 1)$, where $\boldsymbol{\eta} = (\eta_1,...,\eta_D)$ is a further set of binary indicators that, like $\mathbf{Z}$, can be integrated in the EM algorithm as missing variables. A value of $\eta_d = 1$ ($\rho_d = 1$) indicates that feature d has the maximum possible relevance. According to this definition, the mixture density in Eq. (4) can be rewritten as:

$$p(\mathbf{x}|\mathbf{W},\beta,\mathbf{w}_o,\boldsymbol{\beta}_o,\boldsymbol{\rho}) = \sum_{k=1}^{K}\frac{1}{K}\prod_{d=1}^{D}\left\{\rho_d\, p(x_d|\mathbf{u}_k;\mathbf{w}_d,\beta) + (1-\rho_d)q(x_d|\mathbf{u}_o;w_{o,d},\beta_{o,d})\right\} \quad (8)$$

where $\mathbf{w}_d$ is the vector of $\mathbf{W}$ corresponding to feature d and $\boldsymbol{\rho} \equiv \{\rho_1,...,\rho_D\}$. The distribution p is a feature-specific version of Eq. (3). A feature d will be considered irrelevant, with *irrelevance* $(1-\rho_d)$, if $p(x_d|\mathbf{u}_k;\mathbf{w}_d,\beta) = q(x_d|\mathbf{u}_o;w_{o,d},\beta_{o,d})$ for all the mixture components k, where q is a common density followed by feature d. Notice that this is like saying that the distribution for feature d does not follow the cluster structure defined by the model. This common component requires the definition of two extra adaptive parameters: $\mathbf{w}_o \equiv \{w_{o,1},...,w_{o,D}\}$ and $\boldsymbol{\beta}_o \equiv \{\beta_{o,1},...,\beta_{o,D}\}$ (so that $\mathbf{y}_o = \phi_o(\mathbf{u}_o)\mathbf{w}_o$). For fully relevant ($\rho_d \to 1$) features, the common component variance vanishes: $(\beta_{o,d})^{-1} \to 0$. The Maximum Likelihood criterion can now be stated as the estimation of those model parameters that maximize:

$$L_c\left(\mathbf{W},\beta,\mathbf{w}_o,\boldsymbol{\beta}_o,\boldsymbol{\rho}|\mathbf{X},\mathbf{Z}\right)=\sum_{n,k} r_{kn}\sum_{d=1}^{D}\ln\left(a_{knd}+b_{knd}\right) \tag{9}$$

where

$$a_{knd}=\rho_d\left(\beta/2\pi\right)^{1/2}\exp\left(-\frac{\beta}{2}\left(\sum_m \phi_m\left(\mathbf{u}_k\right)w_{md}-x_{nd}\right)^2\right) \tag{10}$$

and

$$b_{knd}=\left(1-\rho_d\right)\left(\beta_{o,d}/2\pi\right)^{1/2}\exp\left(-\frac{\beta_{o,d}}{2}\left(\phi_o\left(\mathbf{u}_o\right)w_{o,d}-x_{nd}\right)^2\right) \tag{11}$$

The *responsibility* r_{kn} in Eq. (7) becomes:

$$r_{kn}=p\left(\mathbf{u}_k|\mathbf{x}_n;\mathbf{W},\beta,\mathbf{w}_o,\boldsymbol{\beta}_o,\boldsymbol{\rho}\right)=\frac{\prod_{d=1}^{D}\left(a_{knd}+b_{knd}\right)}{\sum_{k'=1}^{K}\prod_{d=1}^{D}\left(a_{k'nd}+b_{k'nd}\right)} \tag{12}$$

The maximization of the expected log-likelihood for GTM yields update formulae for the model parameters. The saliency is updated according to:

$$\rho_d^{new}=\frac{1}{N}\sum_{n,k} r_{kn}u_{knd}\ , \tag{13}$$

where $u_{knd}=a_{knd}/\left(a_{knd}+b_{knd}\right)$. For details on the calculations for the rest of the parameters, see [9].

3 Experimental Results and Discussion

Two initialization strategies for FRD-GTM were used in the experiments for this study. The first one fixes the initial values of all the adaptive parameters of the model, following a standard procedure [3,9], thus ensuring the replicability of the results. The second strategy entails initializing $\mathbf{W}$ and $\mathbf{w}_o$ with small values randomly sampled from a normal distribution. This way, different local minima might be reached by the model. For both strategies, saliencies were initialized at $\rho_d=0.5\ ,\forall d,d=1,...,D$.

For the experiments with synthetic data, the grid of GTM latent centres was fixed to a square layout of 3×3 nodes (i.e., 9 constrained mixture components). The corresponding grid of basis functions ϕ_m was fixed to a 2×2 layout. Alternative layouts were tested without significant differences (concerning the goals of the current analyses) being observed. For the experiments with real data, the grid of GTM latent centres was fixed to square layouts of 5×5 and 10×10 nodes (i.e., 25 or 100 constrained mixture components). The corresponding grid of basis functions ϕ_m was fixed to 3×3 and 5×5 layouts.

3.1 Results for Synthetic Data Sets

The aim of these first experiments was discerning whether the FRD-GTM model could approximate the feature relevance determination results for the more general GMM. Data with very specific characteristics were required and, for comparative purposes, we resorted to synthetic sets similar to those used in [5]. The first one (hereafter referred to as *synth1*), with 1,200 data points, consisted of a contrasting combination of features: the first two define four neatly separated Gaussian clusters with centres located at $(0,3),(1,9),(6,4)$ and $(7,10)$; they are meant to be relevant. The next four features are Gaussian noise and, therefore, irrelevant in terms of defining cluster structure. The second synthetic set (hereafter referred to as *synth2*: a variation on the *Trunk* data set used in [5]) was designed for its 10 features to be in decreasing order of relevance. It consisted of 10,000 data points sampled from two Gaussians $\boldsymbol{N}(\mu_1, \mathbf{I})$ and $\boldsymbol{N}(\mu_2, \mathbf{I})$, where $\mu_1 = \left(1, \frac{1}{\sqrt{3}}, ..., \frac{1}{\sqrt{2d-1}}, ..., \frac{1}{\sqrt{19}}\right)$ and $\mu_1 = -\mu_2$.

The FRD results for *synth1* for the fixed initialization strategy are shown in Table 1 and for the varying one in Fig. 1. For both, the first two features yielded the largest saliencies whereas the remaining features yielded very small ones, as expected. This was corroborated by the estimated values of $\boldsymbol{\beta}_o^{-1}$: quite small for the relevant features in comparison to the variances for the remaining features.

Table 1. FRD-GTM estimated values of $\boldsymbol{\rho}$ and $\boldsymbol{\beta}_o^{-1}$, using the fixed initialization strategy described in the main text, for the *synth1* data set

	Relevant features		**Irrelevant features**			
feat #	**1**	**2**	**3**	**4**	**5**	**6**
ρ_d	0.743	0.759	0.106	0.056	0.082	0.066
$\beta_{o,d}^{-1}$	0.102	0.072	1.035	1.028	1.061	0.941

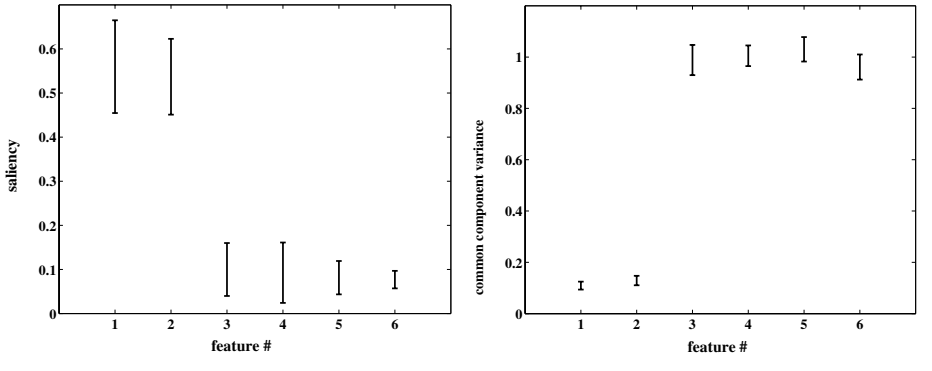


Fig. 1. FRD-GTM estimated values (represented by their means, over 20 runs, plus and minus one standard deviation) of parameters $\boldsymbol{\rho}$ (left) and $\boldsymbol{\beta}_o^{-1}$ (right), using the random varying initialization strategy described in the main text, for the *synth1* data set

Table 2 and Fig. 2 provide similar information for *synth2*. Even though the trend is not perfect, a clear decreasing order of relevance was provided by both the saliencies and $\boldsymbol{\beta}_o^{-1}$. Overall, these results did not reveal major differences between the performance of FRD-GTM and its counterpart procedure for GMM and, consequently, they are not enough to justify the development of FRD-GTM by themselves. As mentioned in previous sections, the extra edge is provided by the visualization capabilities of this model. Fig. 3 illustrates this by showing, in separate plots, the FRD-GTM latent space representation of the 10-dimensional 5,000 data points corresponding to each of the twin Gaussians of *synth2*. This representation is based on the posterior mean projection $\left\langle \mathbf{u} \middle| \mathbf{x}_n, \mathbf{W}, \mathbf{w}_o, \beta, \boldsymbol{\beta}_o, \boldsymbol{\rho} \right\rangle = \sum_k r_{kn} \mathbf{x}_n$ for each 10-dimensional data point $\mathbf{x}_n$. The points corresponding to each Gaussian turn out to be strictly ascribed, with very few exceptions, to a half of the visualization latent space, sharply defined by a vertical boundary. This result confirms that the FRD-GTM reproduces the natural cluster structure of *synth2* fairly well.

Table 2. FRD-GTM estimated values of $\boldsymbol{\rho}$ and $\boldsymbol{\beta}_o^{-1}$, using the fixed initialization strategy described in the main text, for the *synth2* data set

	Feature #									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
ρ_d	1.000	0.692	0.643	0.512	0.461	0.408	0.358	0.355	0.392	0.417
$\beta_{o,d}^{-1}$	0.038	0.625	0.724	0.760	0.843	0.858	0.821	0.867	0.896	1.124

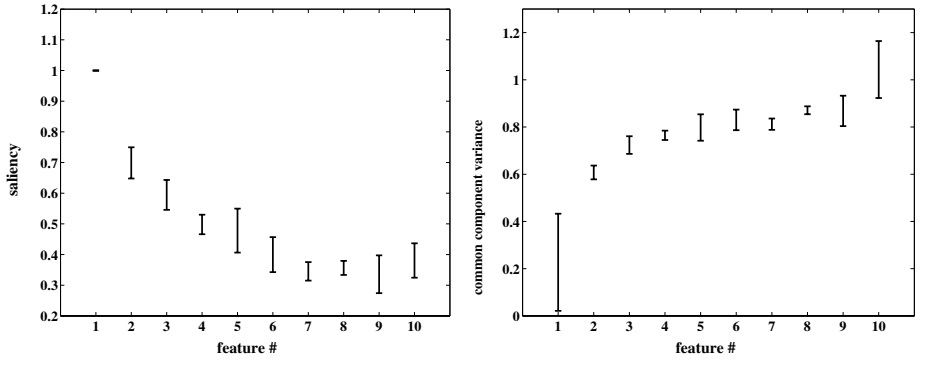


Fig. 2. FRD-GTM estimated values (represented by their means, over 20 runs, plus and minus one standard deviation) of parameters $\boldsymbol{\rho}$ (left) and $\boldsymbol{\beta}_o^{-1}$ (right), using the random varying initialization strategy described in the main text, for the *synth2* data set

3.2 Results for Real Data Sets

The FRD-GTM procedure defined in section 2.2 has already been successfully applied in two problems of real biomedical signal analysis [6,9]. Here, it will be further tested

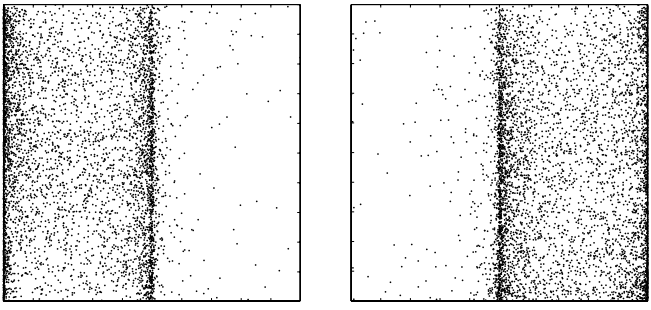


Fig. 3. FRD-GTM representation, on its 2-dimensional visualization latent space, of the 10-dimensional points corresponding to the twin Gaussians of *synth2*. This representation is based on the posterior mean projection described in the main text. (Right) 5,000 points sampled from the 1st Gaussian; (left) 5,000 points sampled from the 2nd Gaussian.

Table 3. Feature saliency rankings (in decreasing order of relevance) for the *Ionosphere* data set. On the top row, results for the FRD-GTM procedure; bottom row: results for the SUD procedure [11]. *Notice that, for FRD-GTM, features 1 and 2 were removed for the analysis. For SUD, these are precisely the most irrelevant features.

GTM*	15,13,21,11,17,19,9,5,23,7,31,27,25,29,3,33,10,8,6,12,22,4,14,20,16,28,30,24,18,26,34,32
SUD	13,15,11,9,7,17,19,21,5,3,23,25,27,29,31,33,10,4,6,12,14,8,16,20,18,22,28,26,24,30,32,34,2,1

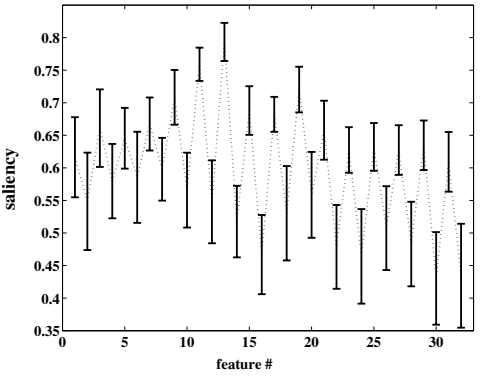


Fig. 4. FRD-GTM estimated values of parameter $\boldsymbol{\rho}$, as in Figs. 1 and 2, for the *Ionosphere* data set. Dotted lines linking the mean values have been kept to appreciate the differences in saliency between the real and complex parts of the radar signal values.

on the well known *Ionosphere* data set from the UCI machine learning repository. It contains radar data consisting of 351 instances and 34 features, the latter consisting of 17 pairs of values. Each pair is formed by the real and complex parts of the values of an autocorrelation function for a pulse number of the radar system signal. The first pair was removed (as in [10]) due to uninformative character of its complex part. The

ionosphere data were meant for classification, as they can be ascribed to one of two categories or classes: "bad radar returns" and "good radar returns". Such classes, in turn, indicate the lack of or the existence of ionosphere structure.

The estimation of the feature saliencies for the *Ionosphere* data yielded some striking results, shown in Fig.4. Consistently, all real parts had higher saliencies than their complex counterparts, meaning that the real parts describing the original signal have a richer cluster structure. To the best of the author's knowledge, this interpretation has not been reported elsewhere, although similar feature ranking results can be found for both unsupervised [11] and supervised [12] models. Table 3 provides comparative ranking results obtained with the application of FRD-GTM and a procedure called Sequential Backward Selection for Unsupervised Data (SUD: [11])

FRD-based feature selection should also ease the interpretability of the clustering results in terms of exploratory visualization. To illustrate this, we cluster a further data set consisting of 30 features and over 100 instances corresponding to physical, chemical and biological measurements from European, human-altered water streams (www.streames.org). The application of FRD-GTM yielded a saliency ranking in which the highest positions corresponded to three features, namely: NO_3^--N: nitrate concentration, conductivity, and D.O.C.: dissolved organic carbon.

Fig. 5 shows the cluster map (in which cluster membership is defined using the posterior mode projection $\mathbf{u}_{k^*,n} = \arg\max_{\mathbf{u}_k} r_{kn}$) for these data. Each cluster can be visually characterized using *reference maps*, which are colour-coded latent space representations of each of the D elements of the reference vectors $\mathbf{y}_k$. A characterization of a cluster on the basis of the 30 original features could be of no use; instead, a characterization on the basis of the 3 most relevant features, such as illustrated in Fig.5, can be far more actionable.

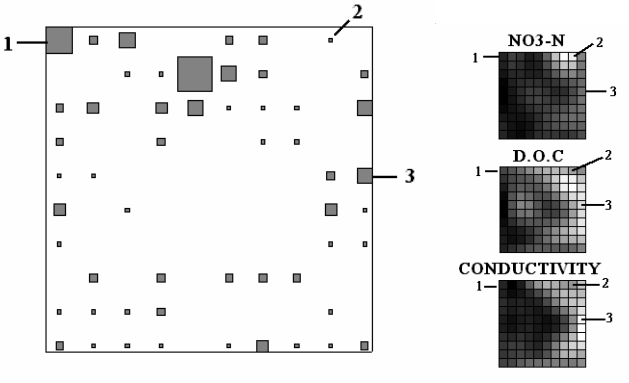


Fig. 5. (left): GTM 10×10 cluster map. The relative size of each cluster (square) indicates the ratio of instances assigned to it. Three clusters, labelled as '1', '2', and '3', are selected to illustrate their interpretation using (right): the 10×10 *reference maps* of the three features with highest saliency. The reference maps are coded in grey-scale, from black (lowest values) to white (highest values), allowing a straightforward interpretation: for instance (and simplifying for the sake of brevity), '1' is characterized by low values of all three features, while '2' is characterized by high values of NO_3^--N and '3' by high levels of D.O.C. and conductivity.

4 Conclusions

The clustering of high-dimensional databases can be difficult to interpret and act upon, and the use of unsupervised feature selection should help to alleviate this problem. A definition of feature relevance for unsupervised clustering with GMMs was recently provided in [5]. In this paper, we have assessed in some detail an extension of this method for the constrained mixture GTM model. The FRD-GTM is capable of simultaneous multivariate data clustering and data visualization, while providing a feature relevance ranking. A series of experiments have been carried out on artificial and real data sets, yielding similar results to GMM. Some of the data visualization capabilities of the GTM that GMMs lack have also been illustrated. They have the potential to ease the interpretation of the clustering results.

Future research should compare in more detail the performance of FRD-GTM with that of alternative FS and clustering methods such as those described in [10].

Acknowledgements

A. Vellido is a research fellow within the Ramón y Cajal program of the Spanish Ministry of Education and Science.

References

1. McLachlan, G.J., Peel, D.: Finite Mixture Models. John Wiley & Sons, New York (2000)
2. Wong, P.C.: Visual data mining. IEEE Comput. Graph. 19(5) (1999) 20-21
3. Bishop, C.M., Svensén, M., Williams, C.K.I.: GTM: The Generative Topographic Mapping. Neural Comput. 10(1) (1998) 215-234
4. Kohonen, T.: Self-Organizing Maps. 3rd edn. Springer-Verlag, Berlin (2000)
5. Law, M.H.C., Figueiredo, M.A.T., Jain, A.K.: Simultaneous Feature Selection and Clustering Using Mixture Models. IEEE T. Pattern Anal. 26(9) (2004) 1154-1166
6. Vellido, A., Lisboa, P.J.G., Vicente, D.: Robust Analysis of MRS Brain Tumour Data Using *t*-GTM. Neurocomputing 69(7-9) (2006) 754-768
7. Dempster, A.P., Laird, M.N., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. J. Roy. Stat. Soc. B 39(1) (1977) 1-38
8. MacKay, D.J.C.: Bayesian Methods for Back-Propagation Networks. In: Domany, E., van Hemmen, J.L., Schulten, K. (eds.): Models of Neural Networks III, Springer, New York (1994) 211-254
9. Andrade, A., Vellido, A.: Determining Feature Relevance for the Grouping of Motor Unit Action Potentials through Generative Topographic Mapping. In: Proc. of the 25th IASTED International Conference Modelling, Identification, and Control (MIC'06) (2006) 507-512
10. Dy, J.G., Brodley, C.E.: Feature Selection for Unsupervised Learning. J. Mach. Learn. Res. 5 (2004) 845–889
11. Dash, M., Liu, H., Yao, J.: Dimensionality Reduction for Unsupervised Data. In: Proc. of the 9th Int. Conf. on Tools with Artificial Intelligence (TAI'97) (1997) 532-539
12. Hunter, A.: Feature Selection Using Probabilistic Neural Networks. Neural Comput. Appl. 9 (2) (2000) 124-132

A Model Selection Method Based on Bound of Learning Coefficient

Keisuke Yamazaki[1,3], Kenji Nagata[2],
Sumio Watanabe[1], and Klaus-Robert Müller[3]

[1] Precision and Intelligence Laboratory,
Tokyo Institute of Technology
R2-5, 4259 Nagatsuta, Midori-ku, Yokohama, 226-8503 Japan
k-yam@pi.titech.ac.jp,
swatanab@pi.titech.ac.jp
[2] Dept. of Computational Intelligence and Systems Science,
Tokyo Institute of Technology
R2-5, 4259 Nagatsuta, Midori-ku, Yokohama, 226-8503 Japan
kenji.nagata@cs.pi.titech.ac.jp
[3] Fraunhofer FIRST, IDA, Kekuléstr. 7, 12489 Berlin, Germany
yamazaki@first.fhg.de,
klaus@first.fhg.de

Abstract. To decide the optimal size of learning machines is a central issue in the statistical learning theory, and that is why some theoretical criteria such as the BIC are developed. However, they cannot be applied to singular machines, and it is known that many practical learning machines e.g. mixture models, hidden Markov models, and Bayesian networks, are singular. Recently, we proposed the Singular Information Criterion (SingIC), which allows us to select the optimal size of singular machines. The SingIC is based on the analysis of the learning coefficient. So, the machines, to which the SingIC can be applied, are still limited. In this paper, we propose an extension of this criterion, which enables us to apply it to many singular machines, and evaluate the efficiency in Gaussian mixtures. The results offer an effective strategy to select the optimal size.

1 Introduction

Practical learning machines, e.g., mixture of distributions, Bayesian networks and hidden Markov models, are used in information engineering. In spite of their various applications, the theoretical properties have not been clarified yet. From the statistical point of view, there are two types of machines. One is regular, the other is singular. A machine is generally described by a probability density function, which has parameters. Roughly speaking, if the mapping from the parameters to the function is one-to-one, the machine is regular. Otherwise, it is singular. For example, Gaussian mixtures are singular. Let the mixture be $p(x|w) = aG(x,b) + (1-a)G(x,c)$, where $w = (a,b,c)$ is the parameter, and $0 \leq a \leq 1$. The function $G(x,b)$ is a Gaussian distribution, and b indicates the mean

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 371–380, 2006.

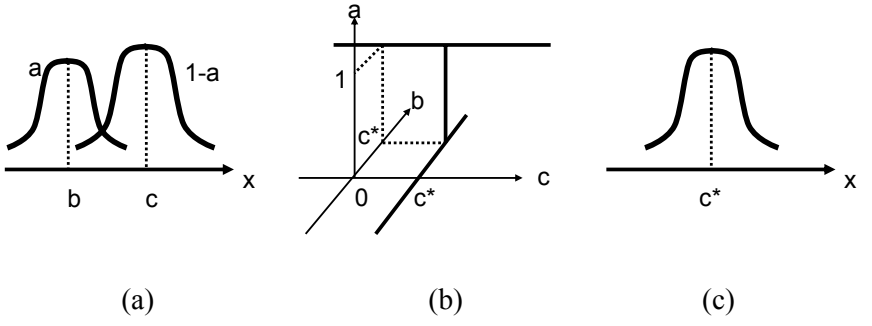


Fig. 1. (a) The Gaussian mixture (b) The parameter space: The bold lines are the union (c) The Gaussian distribution expressed by the union

(Fig. 1-(a)). Assume that $x \in R$ and that the variances of $G(x, b)$ and $G(x, c)$ are common and constant. The union $\{a = 0, c = c^*\} \cup \{a = 1, b = c^*\} \cup \{b = c = c^*\}$ expresses a Gaussian distribution $G(x, c^*)$ in the parameter space (Fig. 1-(b) and (c)). On the union, the mapping is not one-to-one. Moreover, the intersection of these three subspaces indicates singular points. Therefore, the machine is referred to as singular. The properties are unknown because of the singularities, and the importance to analyze them is pointed out [1],[2]

To decide the optimal size of the machine is a central issue, so called the model selection problem, in statistical learning theory [3]. For instance, it is important to estimate the number of Gaussian distributions in the classification. In the Bayes estimation, the BIC [4] uses the stochastic complexity [5] as an evaluation function. However, BIC cannot approximate the complexity in singular machines. In these years, the theory about algebraic geometry and the Bayes estimation was established [6]. As one of the applications, we proposed the Singular Information Criterion (SingIC), which is based on a relation between singularities and the stochastic complexity [7].

In the Bayes estimation, some evaluation functions, such as the stochastic complexity and the generalization error, have the coefficient which includes information about the size of the true machines. Note that the true machine means the one which generates the sample data. In this paper, the coefficient is referred to as a learning coefficient. The SingIC leverages this information to select the true size. Thus, the criterion requires the value of the evaluation function to be calculated by experiments, and the exact form of the coefficient to be written as a function of the size. The form of the coefficient was obtained in some singular machines such as the reduced rank regression [8], and left-to-right HMMs [9]. However, to obtain the exact form is not easy in general singular machines. Instead of the exact one, the bound form was clarified in many machines [10,11,12,13]. In this paper, we propose an extension of the SingIC based on the bound. It enables us to apply the SingIC to more useful models such as hidden Markov models,

Bayesian networks, mixture models, etc. We also show experimental results in Gaussian mixtures to confirm whether our extension works, because the mixture models are widely used in information engineering.

2 The Learning Coefficient

Let $X^n = \{X_1, \cdots, X_n\}$ be a set of sample data, that are independently and identically generated by the true distribution $q(x)$. Let $p(x|w)$ be a learning machine. An *a priori* distribution is $\varphi(w)$ on the set of parameters W. Then, the *a posteriori* distribution is defined by

$$p(w|X^n) = \frac{1}{Z(X^n)} \exp(-nH_n(w))\varphi(w),$$

where

$$Z(X^n) = \int \exp(-nH_n(w))\varphi(w)dw,$$

$$H_n(w) = \frac{1}{n}\sum_{i=1}^{n} \log \frac{q(X_i)}{p(X_i|w)}.$$

The stochastic complexity is defined by

$$F(X^n) = -\log Z(X^n), \tag{1}$$

which is the minus log marginal likelihood. We can select the optimal model in terms of the likelihood to minimize this function. So, it is important to know the mathematical behaviors. To analyze it, the average stochastic complexity is essential. It is defined by

$$F(n) = E_{X^n}\left[F(X^n)\right],$$

where $E_{X^n}[\cdot]$ stands for the expectation value over all sets of samples. Based on the algebraic geometrical method [6], the asymptotic expansion of $F(n)$ is written by

$$F(n) = \lambda \log n - (m-1)\log\log n + o(1),$$

where the rational number $-\lambda$ and natural number m are the largest pole and its order of

$$J(z) = \int H(w)^z \varphi(w)dw,$$

respectively, and

$$H(w) = \int q(x) \log \frac{q(x)}{p(x|w)} dx.$$

The coefficient λ is referred to as the learning coefficient. As you can see in the definition of $H(w)$, the learning coefficient depends on the relation between the true and the learner. According to previous studies [8,9], it is expressed as the

function of their sizes, though $\lambda = d/2$, where d is the dimension of parameter space in regular cases. In other words, λ is determined by both sizes and by only learner's size in singular and regular cases, respectively. This means the learning coefficient includes information about the true size in singular machines. The concept of the SingIC is to leverage this unique behavior.

3 Main Results

3.1 Proposed Model Selection

SingIC requires an observable function [7],

$$y = \mathcal{G}(\lambda),$$

where y stands for the calculated value by experiments and the right-hand side is the asymptotic form. Let $\bar{\lambda}, \underline{\lambda}$ be the upper and lower bounds of λ, respectively. We propose a method to select the true size based on the following inequality,

$$\underline{\lambda} \leq \mathcal{G}^{-1}(y) \leq \bar{\lambda}. \tag{2}$$

Assume that K_0 is the size of $q(x)$ and that $K(> K_0)$ is that of $p(x|w)$. In general, $\underline{\lambda}$ and $\bar{\lambda}$ can be functions of K_0 and K. According to (2), inequalities with respect to K_0 are obtained. Therefore, the true size K_0 can be estimated on the basis of the inequalities. Note that K, y, and a form of the inverse function $\mathcal{G}^{-1}(\cdot)$ are given.

3.2 Example

Our method can be applied to Gaussian mixtures. Let us defined an observable function [7],

$$\begin{aligned} y &= E_{X^n}\left[\frac{\partial F_0(X^n, t)}{\partial t}\right], \\ F_0(X^n, t) &= -\log \int \exp\left(-n\mathcal{H}_t(w)\right) dw, \\ \mathcal{H}_t(w) &= -\frac{1}{n}\left(t\sum_{i=1}^{n} \log p(X_i|w) + \log \varphi(w)\right). \end{aligned}$$

The derivation $\frac{\partial F_0(X^n,t)}{\partial t}$ is computable by using Markov Chain Monte Carlo (MCMC) method since it is written as the expectation value of a Boltzmann distribution [14]. In practical situations, it is difficult to calculate $E_{X^n}[\cdot]$. Thus, we regard the computed value of $\frac{\partial F_0(X^n,t)}{\partial t}$ as y. Moreover, this observable function is the derivation of $E_{X^n}[F_0(X^n, t)]$. The function $E_{X^n}[F_0(X^n, t)]$ corresponds to $F(n)$. The asymptotic form can easily calculated by substituting nt for n,

$$E_{X^n}[F_0(X^n, t)] = ntS + \lambda \log(nt) + o(\log(nt)),$$

where $t > 0$, and

$$S = -\int q(x) \log q(x) dx.$$

Note the difference between $H_n(w)$ and $\mathcal{H}_t(w)$, i.e., we have to add S to the asymptotic form because the definition of the latter function does not include $q(X_i)$. Thus, the observable function has the asymptotic form,

$$y = nS + \frac{\lambda}{t} + o\left(\frac{1}{t}\right).$$

Note that the desired λ is the slope of line with respect to $1/t$. Using the least-squares method, λ is calculated by

$$\lambda_{ty} = \frac{L\sum_{l=1}^{L} x_l y_l - \sum_{l=1}^{L} x_l \sum_{l=1}^{L} y_l}{L\sum_{l=1}^{L} x_l^2 - (\sum_{l=1}^{L} x_l)^2}, \tag{3}$$

where

$$x_l = \frac{1}{t_l},$$
$$y_l = \left.\frac{\partial F_0(X^n, t)}{\partial t}\right|_{t=t_l}$$

for $(1 \leq l \leq L)$. Note that eq. (3) corresponds to the inverse function $\mathcal{G}^{-1}(y)$ and that λ_{ty} is the function of $\{t_l, y_l\}(1 \leq l \leq L)$. In the experiments, $t_1, \cdots t_L$ are fixed, and they are the inverse temperatures of the Boltzmann distribution. Thus, λ_{ty} is regarded as the function of the computed values y_l.

The learning machine is a K component Gaussian mixture define by

$$p(x|w) = \sum_{k=1}^{K} a_k G(x, b_k),$$
$$G(x, b_k) = \frac{1}{(\sqrt{2\pi}\sigma)^M} \exp\left(-\frac{\|x - b_k\|^2}{2\sigma^2}\right),$$

where $a_K = 1 - \sum_{k=1}^{K-1}$, $a_k \geq 0$ for $1 \leq k \leq K$, and $b_k = (b_{k1}, \cdots, b_{kM})$. The parameter is $w = \{a_i, b_j\}(1 \leq i \leq K-1, 1 \leq j \leq K)$. The variance σ is a constant. The true mixture has $K_0 (< K)$ components,

$$q(x) = \sum_{k=1}^{K_0} a_k^* G(x, b_K^*),$$

where $0 < a_k^* < 1, \sum_{k=1}^{K_0} a_k^* = 1$. Since $q(x)$ is fixed, a_k^* and $b_k^* \in R^M (1 \leq k \leq K_0)$ are constants. Then, the following upper bound of λ was obtained [12],

$$\lambda \leq \bar{\lambda} = (K + MK_0 - 1)/2.$$

Thus,

$$(2\lambda_{ty} - K + 1)/M \leq K_0 < K. \tag{4}$$

The inequality is the proposed criterion based on (2). It shows an interval of the true size.

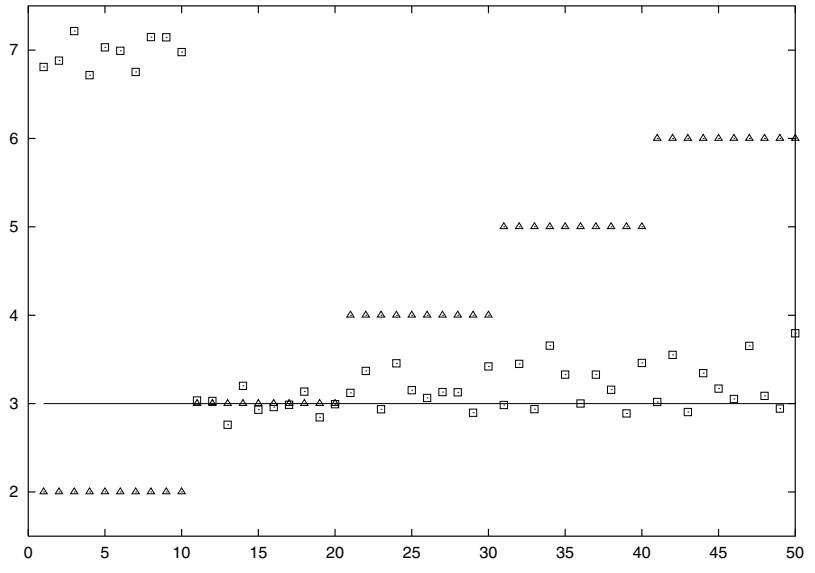


Fig. 2. Experimental results

3.3 Evaluation

Let us evaluate the extension of the SingIC. The true size was fixed as $K_0 = 3$. The sizes of the learner were $K = 2, 3, 4, 5, 6$. The experimental parameters were set as $n = 500$, $t_l = 0.8^l (0 \leq l \leq L = 21)$. The exchange MC (Monte Carlo) method was applied to calculate the value y_l [15]. The parameters t_l and L were determined by this MC method. In the MCMC method, the number of iterations in all inverse temperatures t_l was $10,000$. The sample data were different from each trial. The variance in $G(x, b_i)$ is fixed as $\sigma^2 = 1$. The a priori distributions were a uniform distribution on $[0, 1]$ for a_i, and a standard normal distribution for b_i. The results are summarized in Fig. 2. The vertical axis is the (estimated) size of the true distribution and horizontal one shows trials. The triangle marks stand for K. So, the learner's size is $K = 2$ from the first trial to tenth, $K = 3$ from eleventh to twentieth, and so on. There is a horizontal line, which indicates the true size $K_0 = 3$. The square marks are the calculated lower bound of K_0 in (4). Therefore, the true size is inferred between the triangle and square dots.

4 Discussion and Conclusion

First, let us consider the experimental results. The estimated lower bounds (the square dots) are all in range $[2.5, 4]$ independent of learners $K = 3, 4, 5, 6$. This means the bound of $\bar{\lambda}$ is tight, since the theoretical lower bound is always equal to the true size in $\bar{\lambda} = \lambda$. When the learner is smaller than the true ($K = 2 < K_0 = 3$), the estimated lower bound is clearly unusual: the square dots are much larger

than K. It is easy to find that the assumption, where the learner can achieve the true, does not hold. Hereafter, we discuss the results in $K = 3, \cdots, 6$. The square dots actually tend to be larger than the theoretical bound. So, there are some data, which make the lower bound larger than the true size. The predicted reasons are

- The bound is asymptotic.
- The MCMC methods includes the error.

The former can be solved to collect more data, and the latter to take more time or to modify the method. However, it is necessary to select the size without enough data or without enormous time in practical situations. Then, the experimental results offer the strategy,

1. Initialize a size of the learning machine.
2. Estimate the lower bound with the size.
3. If the bound is unusual (much larger than the learner's size or minus), set a larger size. Go to 2
4. If the bound is around the same size as learner's, the size will be true. [The end of the procedure]
5. Otherwise, set the size around the bound. Go to 2.

This strategy is based on the facts that the calculated lower bound (the square dots) can be larger than the true size (the horizontal line) and that they are not far from each other.

Next, we compare our method to other methods to decide the optimal size. The stochastic complexity, eq.(1), can also be a criterion. After the values at the all candidate sizes are obtained, the minimum point shows the optimal size in the sense of the marginal likelihood (Fig.3-(a)). The calculation of the stochastic complexity at any size corresponds to the estimation of the lower bound in our method. Our method does not need the estimation at all candidate sizes. Consequently, it reduces the computational cost.

The schematic figure (Fig.3) shows that the optimal size $K_0 = 3$ is selected from candidate sizes $K = 1, \ldots, 6$ in both methods. On the one hand, the method with the stochastic complexity needs the whole points to draw the curve (a). On the other hand, our method with the strategy does not need all sizes (b). The cross-validation also needs computation at all candidate sizes to decide the optimal size. The Bayes predictive distribution at each size is based on the posterior realized by the MCMC method. Therefore, we need the MCMC calculation at all sizes in the cross-validation. It is well known that the MCMC method requires huge amount of computational cost when the dimension of the parameter space is large. Our method has an advantage in such a case, i.e., to reduce the candidate sizes is efficient in terms of the total amount, even though the cost for one size increases. The well known BIC also reduce the cost in regular models. However, note that it cannot approximate the stochastic complexity in singular cases. The parameter which expresses the true distribution is not one point but a set of parameters.

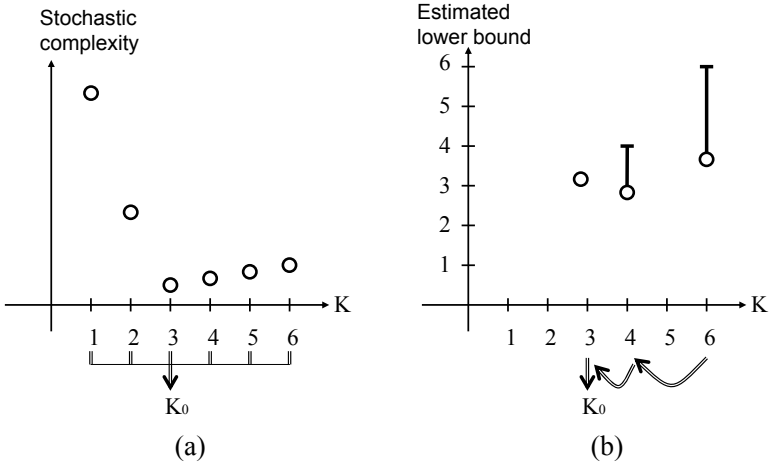


Fig. 3. (a) The method with the stochastic complexity. (b) The proposed method with the strategy.

The BIC uses the expansion around this one true point. This is why the BIC cannot have reasonable approximation in singular cases.

Last, we summarize the conditions to use our method.

– The learning coefficient or its bound in the machine has to be known.

The method leverages information of the true size in a bound of the learning coefficient. The original SingIC needs the exact form of λ though the analysis is quite complex [8],[9]. Our proposed extension requires the form of $\bar{\lambda}$ or $\underline{\lambda}$. The upper bound is obtained in many machines [12], [13], [16], [17]. So, we can easily apply this extension to these machines. The tighter bound of λ achieves the more precise estimation of the true size. Conversely, our method provides an evaluation how tight the bound is according to eq. (4).

– The computable function should have the coefficient or its bound.

Our experiments used the exchange MC method, which is referred to as one of precise methods in order to calculate the observation function. The analysis of the variational Bayesian method was recently developed [10], [11]. In this analysis, it is shown that the bound of the stochastic complexity has a similar coefficient including the true size. Our method can be applied to the results.

– The learner can attain the true distribution.

It is very important to consider the situations where this condition is violated, though we often regard our learner as an attainable model. In the experimental results, $K = 2$ is an example of such situations. Actually, the results at $K = 2$ do not seem to have reasonable computed lower bounds. It might be thought

that our method does not work in the situations. However, this behavior as the SingIC depends on the observable function, i.e., another observable function can make it different. For the fist step to tackle this issue, we need to investigate a relation between the behavior and the function. It is one of our future studies.

This paper showed an extension of the SingIC and evaluated its efficiency in Gaussian mixtures. Our future goal is to apply the theoretical concept to a more practical model and more realistic data.

Acknowledgment

This work was partially supported by the Alexander von Humboldt Foundation and by the Ministry of Education, Science, Sports, and Culture in Japan, Grant-in-aid for scientific research 15500130 and 18-5809.

References

1. Amari, S., Ozeki, T.: Differential and algebraic geometry of multilayer perceptrons. IEICE Trans **E84-A 1** (2001) 31–38
2. Hartigan, J.A.: A failure of likelihood asymptotics for normal mixtures. Proc. of the Berkeley Conference in Honor of J.Neyman and J.Kiefer **2** (1985) 807–810
3. Akaike, H.: A new look at the statistical model identification. IEEE Trans. on Automatic Control **19** (1974) 716–723
4. Schwarz, G.: Estimating the dimension of a model. Annals of Statistics **6 (2)** (1978) 461–464
5. Rissanen, J.: Stochastic complexity and modeling. Annals of Statistics **14** (1986) 1080–1100
6. Watanabe, S.: Algebraic analysis for non-identifiable learning machines. Neural Computation **13 (4)** (2001) 899–933
7. Yamazaki, K., Nagata, K., Watanabe, S.: A new method of model selection based on learning coefficient. In: Proc. of International Symposium on Nonlinear Theory and its Applications. (2005) 389–392
8. Aoyagi, M., Watanabe, S.: The generalization error of reduced rank regression in bayesian estimation. In: Proc. of ISITA. (2004) 1068–1073
9. Yamazaki, K., Watanabe, S.: Learning coefficient of hidden markov models. Technical Report of IEICE **NC2005-14** (2005) 37–42
10. Hosino, T., Watanabe, K., Watanabe, S.: Stochastic complexity of variational bayesian hidden markov models. In: Proc. of International Joint Conference on Neural Networks. (2005) 1114–1119
11. Watanabe, K., Watanabe, S.: Variational bayesian stochastic complexty of mixture models. In: MIT press. (to appear)
12. Watanabe, S., Yamazaki, K., Aoyagi, M.: Kullback information of normal mixture is not an analytic function. Technical Report of IEICE (in Japanese) **NC2004-50** (2004) 41–46
13. Yamazaki, K., Watanabe, S.: Stochastic complexity of bayesian networks. In: Proc. of UAI. (2003) 592–599
14. Ogata, Y.: A monte carlo method for an objective bayesian procedure. Ann. Inst. Statis. Math. **42 (3)** (1990) 403–433

15. Hukushima, K., Nemoto, K.: Exchange monte carlo method and application to spin glass simulations. Journal of Physical Society of Japan **65**(6) (1996) 1604–1608
16. Yamazaki, K., Watanabe, S.: Singularities in complete bipartite graph-type boltzmann machines and upper bounds of stochastic complexities. IEEE Trans. on Neural Networks **16**(2) (2005) 312–324
17. Yamazaki, K., Watanabe, S.: Generalization errors in estimation of stochastic context-free grammar. In: The IASTED International Conference on ASC. (2005) 183–188

Sequential Learning with LS-SVM for Large-Scale Data Sets

Tobias Jung[1] and Daniel Polani[2]

[1] Department of Computer Science, University of Mainz, Germany
[2] School of Computer Science, University of Hertfordshire, UK

Abstract. We present a subspace-based variant of LS-SVMs (i.e. regularization networks) that sequentially processes the data and is hence especially suited for online learning tasks. The algorithm works by selecting from the data set a small subset of basis functions that is subsequently used to approximate the full kernel on arbitrary points. This subset is identified online from the data stream. We improve upon existing approaches (esp. the kernel recursive least squares algorithm) by proposing a new, supervised criterion for the selection of the relevant basis functions that takes into account the approximation error incurred from approximating the kernel as well as the reduction of the cost in the original learning task. We use the large-scale data set 'forest' to compare performance and efficiency of our algorithm with greedy batch selection of the basis functions via orthogonal least squares. Using the same number of basis functions we achieve comparable error rates at much lower costs (CPU-time and memory wise).

1 Introduction and Related Work

Introduction. In this paper we adress the problem of sequential learning when the predictor has the form of least squares SVM (LS-SVM). Since there is no way we can achieve this using in our model one independent parameter for each training example (i.e. basis function), we use a projection-based technique that only considers a small subset of all possible basis functions. This subset is selected online from the training data by just inspecting the most recent example. Our resulting algorithm is conceptually similar to the kernel recursive least squares (KRLS) algorithm proposed in [4], yet improves it in two important ways: one is that we consider a *supervised* criterion for the selection of the relevant basis functions that takes into account the reduction of the cost in the original learning task in addition to the error incurred from approximating the kernel. Since the per-step complexity only depends on the size of the subset, making sure that no unnecessary basis functions are selected ensures more efficient usage of otherwise scarce resources. And second, by considering a *pruning* operation we can also delete basis functions from the subset to have an even tighter control over its size. Overall the algorithm is very resource efficient, and only depends on the number of examples stored in the subset.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 381–390, 2006.

Related work. The unfavorable $\mathcal{O}(n^3)$ scaling of kernel-based learning has spawned a number of approaches where the exact solution is approximated by a solution with lower complexity. Well known examples are the Nyström method [13] or the subset of regressors method (SR), mentioned e.g. in [7,12,10]. Both methods work by projecting the kernel onto a much smaller subset of kernels chosen from the full data, say of size $m \ll n$, and reduce computational complexity to $\mathcal{O}(nm^2)$. To select the subset we can categorize the various approaches as being unsupervised and supervised. Unsupervised approaches like random selection [13] or the incomplete Cholesky decomposition (IC) [5] do not use information about the task we want to solve, i.e. the response variable we wish to regress upon. Random selection does not use any information at all whereas IC aims at reducing the error from approximating the kernel matrix. Supervised choice of the subset does take into account the response variable and usually proceeds by greedy forward selection, using e.g. matching pursuit techniques [11] or the recent Cholesky decomposition with side information [1]. However, none of these approaches are directly applicable for sequential learning, since they all use information from the complete data set. Working in the context of Gaussian process regression (GPR), [2] and also [4] have proposed an online variant, which adds examples directly from the data stream and is the basis of our work presented here.

2 Background

Traditional setup. Given t examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^t$ with $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ being the inputs and $y_i \in \mathcal{Y} \subset \mathbb{R}$ being the outputs, the goal is to reconstruct (learn) the underlying function. Consider as the space of candidate functions the reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$ of functions $f : \mathcal{X} \to \mathcal{Y}$ endowed with reproducing kernel k, where $k : \mathcal{X} \times \mathcal{X} \to \mathcal{Y}$ is a symmetric, positive definite function (e.g. think of Gaussian RBF). The underlying function can be reconstructed solving the Tikhonov functional: $\min_{f \in \mathcal{H}_k} J[f] = \sum_{i=1}^t \big(y_i - f(\mathbf{x}_i)\big)^2 + \gamma \|f\|^2_{\mathcal{H}_k}$ with $\gamma > 0$ being the regularization parameter. The Representer theorem tells us that any solution to this variational problem has a representation in the form $f(\cdot) = \sum_{i=1}^t \beta_i k(\mathbf{x}_i, \cdot)$ i.e. as a sum of kernels centered on the data. Plugging this back into the original variational problem leads to the optimization problem

$$\min_{\beta \in \mathbb{R}^t} \|\mathbf{y} - \mathbf{K}\boldsymbol{\beta}\|^2 + \gamma \boldsymbol{\beta}^T \mathbf{K} \boldsymbol{\beta} \tag{1}$$

with $\mathbf{y}$ being the $t \times 1$ vector of observations, $\mathbf{K}$ being the dense $t \times t$ kernel matrix $[\mathbf{K}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ of pairwise similarities and $\boldsymbol{\beta}$ being a $t \times 1$ vector. From (1) the coefficients $\boldsymbol{\beta}$ can be obtained by solving

$$\big(\mathbf{K}^T\mathbf{K} + \gamma \mathbf{K}\big)\boldsymbol{\beta} = \mathbf{K}^T\mathbf{y} \tag{2}$$

which gives the solution as $\boldsymbol{\beta} = (\mathbf{K} + \gamma \mathbf{I})^{-1}\mathbf{y}$ due to $\mathbf{K}$ being symmetric and positive definite. Solving (2) is generally a matter of $\mathcal{O}(t^3)$ operations. The overbearing computational burden stems from the fact that every training example will contribute one parameter to the resulting model.

The subset of regressors method (SR). Consider a subset $\{\tilde{\mathbf{x}}_i\}_{i=1}^m$, $m \ll t$, of data points selected from the full set $\{\mathbf{x}_i\}_{i=1}^t$, without loss of generality assume that these are the first m examples. We approximate the kernel on arbitrary points through linear combination of kernels from the subset (termed the *dictionary* or set of *basis vectors* $\mathcal{BV}$ in [2] which we adopt for the remainder of this paper) in the following way: $k(\mathbf{x}, \cdot) \approx \sum_{i=1}^m a_i k(\tilde{\mathbf{x}}_i, \cdot)$. The $m \times 1$ vector $\mathbf{a} = (a_1, \ldots, a_m)^T$ is determined such that the distance in $\mathcal{H}_k$ for a given $\mathbf{x}$

$$\delta = \min_{\mathbf{a}} \left\| k(\mathbf{x}, \cdot) - \sum_{i=1}^m a_i k(\tilde{\mathbf{x}}_i, \cdot) \right\|_{\mathcal{H}_k}^2 \tag{3}$$

is minimized. The solution to this problem follows as

$$\mathbf{a} = \mathbf{K}_{mm}^{-1} \mathbf{k}_m(\mathbf{x}) \tag{4}$$

where the $m \times m$ matrix $\mathbf{K}_{mm}$ is the kernel matrix corresponding to the dictionary (i.e. the upper left $m \times m$ submatrix of $\mathbf{K}$) and $m \times 1$ vector $\mathbf{k}_m(\mathbf{x})$ is shorthand for vector $\mathbf{k}_m(\mathbf{x}) = (k(\tilde{\mathbf{x}}_1, \mathbf{x}), \ldots, k(\tilde{\mathbf{x}}_m, \mathbf{x}))^T$. For arbitrary $\mathbf{x}, \mathbf{x}'$ we thus have the approximation

$$k(\mathbf{x}, \mathbf{x}') \approx [\mathbf{k}_m(\mathbf{x})]^T \mathbf{K}_{mm}^{-1} \mathbf{k}_m(\mathbf{x}'). \tag{5}$$

If either $\mathbf{x}$ or $\mathbf{x}'$ are in $\mathcal{BV}$ then (5) is exact. Replacing the true kernel by (5) gives $\mathbf{K}_{tm} \mathbf{K}_{mm}^{-1} \mathbf{K}_{tm}^T \approx \mathbf{K}$ as an approximation to the true kernel matrix $\mathbf{K}$, where $\mathbf{K}_{tm}$ is the $t \times m$ submatrix of the first m columns of $\mathbf{K}$ (again, corresponding to the $\mathcal{BV}$). Defining the $t \times m$ matrix $\mathbf{A}$ with rows $\mathbf{a}_i^T = \mathbf{K}_{mm}^{-1} \mathbf{k}_m(\mathbf{x}_i)$, $i = 1, \ldots, t$ from (4) we can write

$$\mathbf{K}_{tm} = \mathbf{A} \mathbf{K}_{mm}. \tag{6}$$

In the SR-method [7,11,10] instead of using the full representation one only uses the kernels in $\mathcal{BV}$, i.e. $f(\cdot) = \sum_{i=1}^m \beta_i k(\tilde{\mathbf{x}}_i, \cdot)$, and obtain in place of (1) the penalized least squares problem

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^m} \|\mathbf{y} - \mathbf{K}_{tm} \boldsymbol{\beta}\|^2 + \gamma \boldsymbol{\beta}^T \mathbf{K}_{mm} \boldsymbol{\beta} \tag{7}$$

which has the solution $\boldsymbol{\beta} = \left(\mathbf{K}_{tm}^T \mathbf{K}_{tm} + \gamma \mathbf{K}_{mm}\right)^{-1} \mathbf{K}_{tm}^T \mathbf{y}$. Despite its cursory similarity with (2) we have gained much since now we are only dealing with m parameters and computational complexity is down to $\mathcal{O}(tm^2)$.

Csató and Opper's sparse greedy online approximation. Still, the SR-method is not directly applicable for online learning. Assume that the data arrives sequentially at $t = 1, 2, \ldots$ and that only one pass over the data set is possible, so that we cannot select the subset $\mathcal{BV}$ in advance. Working in the context of GPR, [2] and later [4] have proposed sparse greedy online approximation: start from an empty set $\mathcal{BV}$ and examine at every time step t, if the current example needs to be included in $\mathcal{BV}$ or if it can be processed without augmenting $\mathcal{BV}$. The approximation in (5) is modified such that it uses the most

recent version of $\mathcal{BV}$ and sets to zero those entries from $\mathbf{a}$ that correspond to basis vectors added in future time steps (denoted by $\tilde{\mathbf{A}}$). Thus the matrix used in (7) no longer equals the submatrix $\mathbf{K}_{tm}$ from (6), since now $\tilde{\mathbf{K}}_{tm} =_{\text{def}} \tilde{\mathbf{A}}\mathbf{K}_{mm}$ is only an approximation.

The one crucial advantage of this approach is that now we can use (penalized) least squares methods as in (7) together with online growing and pruning operations for sequential learning by using only the examples memorized in the set $\mathcal{BV}$. (Otherwise, to augment or prune an existing model we would need to work with all previously seen data or resort to a window of a fixed given size.)

3 Time-Recursive LS-SVM

In this section we present the main contribution of our work: online LS-SVM using sparse online approximation and a novel criterion for the selection of relevant basis functions to include in the subset. The algorithm works along the lines of recursive least squares, i.e. propagates forward the inverse of the cross product matrix.

Let t be the current time step, $(\mathbf{x}_{t+1}, y^*)$ the currently observed input-output pair and assume that from the past t examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{t}$ the m examples $\{\tilde{\mathbf{x}}_i\}_{i=1}^{m}$ were selected into the dictionary $\mathcal{BV}$. Consider the penalized least squares problem that is LS-SVM (restated here from (7) for clarity)

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^m} J_{tm}(\boldsymbol{\beta}) = \left\|\mathbf{y}_t - \tilde{\mathbf{K}}_{tm}\boldsymbol{\beta}\right\|^2 + \gamma\boldsymbol{\beta}^T\mathbf{K}_{mm}\boldsymbol{\beta} \tag{8}$$

with $\tilde{\mathbf{K}}_{tm} = \tilde{\mathbf{A}}\mathbf{K}_{mm}$ being the (approximated) $t \times m$ design matrix from (6) and $\mathbf{y}_t$ being the $t \times 1$ vector of the observed output values. Note that we are using a double index to indicate the dependence on the number of examples t and the number of basis functions m. If we define the $m \times m$ cross product matrix $\mathbf{P}_{tm} = (\tilde{\mathbf{K}}_{tm}^T\tilde{\mathbf{K}}_{tm} + \gamma\mathbf{K}_{mm})$ then the solution to (8) is given by $\boldsymbol{\beta}_{tm} = \mathbf{P}_{tm}^{-1}\tilde{\mathbf{K}}_{tm}^T\mathbf{y}_t$. Finally we introduce the costs $\xi_{tm} = J_{tm}(\boldsymbol{\beta}_{tm})$. Assuming that $\{\mathbf{P}_{tm}^{-1},\ \boldsymbol{\beta}_{tm},\ \xi_{tm}\}$ are known from previous computations, every time a new example $(\mathbf{x}_{t+1}, y^*)$ is presented we will perform one or more of the following update operations:

1. *Normal step*: Process $(\mathbf{x}_{t+1}, y^*)$ in the usual way using the fixed set of basis functions $\mathcal{BV}$.
2. *Growing step*: If the new example is sufficiently different from the previous examples in $\mathcal{BV}$ (i.e. the reconstruction error in (3) exceeds a given threshold) and strongly contributes to the solution of the problem (i.e. the decrease of the loss when adding the new basis function is greater than a given threshold) then the current example is added to $\mathcal{BV}$ and the number of basis functions in the model is increased by one.
3. *Pruning step*: If the current size of the $\mathcal{BV}$ set exceeds the allowed maximum number of $\mathcal{BV}$s specified prior to starting the algorithm, remove from $\mathcal{BV}$ the basis function that contributes the least to the reduction of the cost function.

Integral to these updates are two well-known matrix identities for recursively computing the inverse of a matrix: (for suitable matrices)

$$\text{if } \mathbf{B}_{t+1} = \mathbf{B}_t + \mathbf{b}\mathbf{b}^T \text{ then } \mathbf{B}_{t+1}^{-1} = \mathbf{B}_t^{-1} - \frac{\mathbf{B}_t^{-1}\mathbf{b}\mathbf{b}^T\mathbf{B}_t^{-1}}{1 + \mathbf{b}^T\mathbf{B}_t^{-1}\mathbf{b}} \tag{9}$$

which is used when adding a row to the design matrix. Likewise,

$$\text{if } \mathbf{B}_{t+1} = \begin{bmatrix} \mathbf{B}_t & \mathbf{b} \\ \mathbf{b}^T & b^* \end{bmatrix} \text{ then } \mathbf{B}_{t+1}^{-1} = \begin{bmatrix} \mathbf{B}_t^{-1} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} + \frac{1}{\Delta_b} \begin{bmatrix} -\mathbf{B}_t^{-1}\mathbf{b} \\ 1 \end{bmatrix} \begin{bmatrix} -\mathbf{B}_t^{-1}\mathbf{b} \\ 1 \end{bmatrix}^T \tag{10}$$

with $\Delta_b = b^* - \mathbf{b}^T\mathbf{B}_t^{-1}\mathbf{b}$. This second update is used when adding a column to the design matrix.

3.1 Normal Step: From $\{\mathbf{P}_{tm}^{-1}, \boldsymbol{\beta}_{tm}, \boldsymbol{\xi}_{tm}\}$ to $\{\mathbf{P}_{t+1,m}^{-1}, \boldsymbol{\beta}_{t+1,m}, \boldsymbol{\xi}_{t+1,m}\}$

Let $\mathbf{k}_{t+1}$ be $\mathbf{k}_{t+1} = (k(\tilde{\mathbf{x}}_1, \mathbf{x}_{t+1},), \ldots, k(\tilde{\mathbf{x}}_m, \mathbf{x}_{t+1}))^T$, then

$$\tilde{\mathbf{K}}_{t+1,m} = \begin{bmatrix} \tilde{\mathbf{K}}_{tm} \\ \mathbf{k}_{t+1}^T \end{bmatrix} \quad \text{and} \quad \mathbf{y}_{t+1} = \begin{bmatrix} \mathbf{y}_t \\ y^* \end{bmatrix}.$$

Thus $\mathbf{P}_{t+1,m} = \mathbf{P}_{tm} + \mathbf{k}_{t+1}\mathbf{k}_{t+1}^T$ and we obtain from (9) the well-known RLS updates

$$\begin{aligned} \mathbf{P}_{t+1,m}^{-1} &= \mathbf{P}_{tm}^{-1} - \frac{\mathbf{P}_{tm}^{-1}\mathbf{k}_{t+1}\mathbf{k}_{t+1}^T\mathbf{P}_{tm}^{-1}}{\Delta}, \qquad \boldsymbol{\beta}_{t+1,m} = \boldsymbol{\beta}_{tm} + \frac{\varrho}{\Delta}\mathbf{P}_{tm}^{-1}\mathbf{k}_{t+1} \\ \xi_{t+1,m} &= \xi_{tm} + \frac{\varrho^2}{\Delta} \end{aligned} \tag{11}$$

with scalars $\Delta = 1 + \mathbf{k}_{t+1}^T\mathbf{P}_{tm}^{-1}\mathbf{k}_{t+1}$ and $\varrho = y^* - \mathbf{k}_{t+1}^T\boldsymbol{\beta}_{tm}$. The set $\mathcal{BV}$ is not altered during this step. Operation count is $\mathcal{O}(m^2)$.

3.2 Growing Step: From $\{\mathbf{P}_{tm}^{-1}, \boldsymbol{\beta}_{tm}, \boldsymbol{\xi}_{tm}\}$ to $\{\mathbf{P}_{t,m+1}^{-1}, \boldsymbol{\beta}_{t,m+1}, \boldsymbol{\xi}_{t,m+1}\}$

How to add a $\mathcal{BV}$. When adding an additional basis function (centered on $\mathbf{x}_{t+1}$) to the model we augment the set $\mathcal{BV}$ with $\tilde{\mathbf{x}}_{m+1}$ (note that $\tilde{\mathbf{x}}_{m+1}$ is the same as $\mathbf{x}_{t+1}$ from above). Again, define $\mathbf{k}_{t+1} = (k(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_{m+1}), \ldots, k(\tilde{\mathbf{x}}_m, \tilde{\mathbf{x}}_{m+1}))^T$ and $k^* = k(\tilde{\mathbf{x}}_{m+1}, \tilde{\mathbf{x}}_{m+1})$. Adding a basis function means appending a new $t \times 1$ vector $\mathbf{q}$ to the design matrix and appending $\mathbf{k}_{t+1}$ as row/column to the penalty matrix $\mathbf{K}_{mm}$, thus

$$\mathbf{P}_{t,m+1} = \begin{bmatrix} \tilde{\mathbf{K}}_{tm} & \mathbf{q} \end{bmatrix}^T \begin{bmatrix} \tilde{\mathbf{K}}_{tm} & \mathbf{q} \end{bmatrix} + \gamma \begin{bmatrix} \mathbf{K}_{mm} & \mathbf{k}_{t+1} \\ \mathbf{k}_{t+1}^T & k^* \end{bmatrix}.$$

Invoking (10) we obtain the updated inverse $\mathbf{P}_{t,m+1}^{-1}$ via

$$\mathbf{P}_{t,m+1}^{-1} = \begin{bmatrix} \mathbf{P}_{tm}^{-1} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} + \frac{1}{\Delta_b} \begin{bmatrix} -\mathbf{w}_b \\ 1 \end{bmatrix} \begin{bmatrix} -\mathbf{w}_b \\ 1 \end{bmatrix}^T \tag{12}$$

where simple but tedious vector algebra reveals that

$$\begin{aligned}\mathbf{w}_b &= \mathbf{P}_{tm}^{-1}(\tilde{\mathbf{K}}_{tm}^T\mathbf{q} + \gamma\mathbf{k}_{t+1}) \\ \Delta_b &= \mathbf{q}^T\mathbf{q} + \gamma k^* - (\tilde{\mathbf{K}}_{tm}^T\mathbf{q} + \gamma\mathbf{k}_{t+1})^T\mathbf{w}_b. \end{aligned} \tag{13}$$

Without sparse online approximation this step requires us to recall all past examples $\{\mathbf{x}_i\}_{i=1}^t$ since $\mathbf{q}$ is given by $\mathbf{q}^T = (k(\tilde{\mathbf{x}}_{m+1}, \mathbf{x}_1), \ldots, k(\tilde{\mathbf{x}}_{m+1}, \mathbf{x}_t))^T$ and just obtaining (13) would come at the undesirable price of $\mathcal{O}(tm)$. However, we are going to get away with merely $\mathcal{O}(m)$ operations and only need to memorize examples in $\mathcal{BV}$. Due to the sparse approximation $\mathbf{q}$ is actually of the form $\mathbf{q}^T = \begin{bmatrix}\tilde{\mathbf{K}}_{t-1,m}\mathbf{a}_{t+1} & k^*\end{bmatrix}^T$ with $\mathbf{a}_{t+1} = \mathbf{K}_{mm}^{-1}\mathbf{k}_{t+1}$ from (4). Hence new information is injected only through the last component. Exploiting this special structure of $\mathbf{q}$ equation (13) becomes

$$\begin{aligned}\mathbf{w}_b &= \mathbf{a}_{t+1} + \frac{\delta}{\Delta}\mathbf{P}_{t-1,m}^{-1}\mathbf{k}_{t+1} \\ \Delta_b &= \frac{\delta^2}{\Delta} + \gamma\delta \end{aligned} \tag{14}$$

where $\delta = k^* - \mathbf{k}_{t+1}^T\mathbf{a}_{t+1}$ from (3). If we cache and reuse those terms already computed in the preceding step (see Sect. 3.1) then we can obtain $\mathbf{w}_b, \Delta_b$ in $\mathcal{O}(m)$ operations.

To obtain the updated coefficients $\boldsymbol{\beta}_{t,m+1}$ we first multiply (12) from the right side by $\tilde{\mathbf{K}}_{t,m+1}^T\mathbf{y}_t = \begin{bmatrix}\tilde{\mathbf{K}}_{tm}^T\mathbf{y}_t & \mathbf{q}^T\mathbf{y}_t\end{bmatrix}^T$ and get

$$\boldsymbol{\beta}_{t,m+1} = \begin{bmatrix}\boldsymbol{\beta}_{tm} \\ 0\end{bmatrix} + \kappa\begin{bmatrix}-\mathbf{w}_b \\ 1\end{bmatrix} \tag{15}$$

where scalar κ is defined by $\kappa = \mathbf{y}_t^T(\mathbf{q} - \tilde{\mathbf{K}}_{tm}\mathbf{w}_b)/\Delta_b$. Again we can now exploit the special structure of $\mathbf{q}$ to show that κ is equal to

$$\kappa = -\frac{\delta\varrho}{\Delta_b\Delta}$$

And again we can reuse terms computed in the previous step (see Sect. 3.1).

Skipping the necessary computations, we can show that the reduced (regularized) cost $\xi_{t,m+1}$ is recursively obtained from ξ_{tm} via the expression:

$$\xi_{t,m+1} = \xi_{tm} - \kappa^2\Delta_b. \tag{16}$$

Finally, every time we add an example to the $\mathcal{BV}$ set we must also update the inverse kernel matrix $\mathbf{K}_{mm}^{-1}$ needed during the computation of $\mathbf{a}_{t+1}$ and δ. This can be done using the formula for partitioned matrix inverses (10).

When to add a $\mathcal{BV}$. To decide whether or not the current example $\mathbf{x}_{t+1}$ should be added to the $\mathcal{BV}$ set, we employ a two-part criterion, similar to the one used in resource-allocating networks [8]. The first part measures the 'novelty' of the

current example: only examples that are 'far' from those already stored in the $\mathcal{BV}$ set are considered for inclusion. To this end we compute as in [2,4] the squared norm of the residual from projecting (in RKHS) the example onto the span of the current $\mathcal{BV}$ set, i.e. we compute (restated from (3)) $\delta = k^* - \mathbf{k}_{t+1}^T \mathbf{a}_{t+1}$. If $\delta <$ `TOL1` for a given threshold `TOL1`, then $\mathbf{x}_{t+1}$ is well represented by the given $\mathcal{BV}$ set and its inclusion would not contribute much to reduce the error from approximating the kernel by the reduced set. On the other hand, if $\delta >$ `TOL1` then $\mathbf{x}_{t+1}$ is not well represented by the current $\mathcal{BV}$ set and leaving it behind could incur a large error in the approximation of the kernel.

However, using as sole criterion the reduction of the error incurred from approximating the kernel is probably too wasteful of resources, since examples could get selected into the subset that are unrelated to the original task [1]. We want to be more restrictive, particularly because the computational complexity per step scales with the square of basis functions in $\mathcal{BV}$ (so that the size of $\mathcal{BV}$ will soon become the limiting factor). Aside from novelty, here we thus consider as second part of the selection criterion the 'usefulness' of a basis function candidate. Usefulness is taken to be its contribution to the reduction of the regularized costs, i.e. the term $\kappa^2 \Delta_b$ from (16). Both parts together are combined into one rule: only if $\delta \cdot \kappa^2 \cdot \Delta_b >$ `TOL2` then the current example will become a new basis function and will be added to $\mathcal{BV}$.

3.3 Pruning Step: From $\{\mathbf{P}_{tm}^{-1}, \boldsymbol{\beta}_{tm}, \boldsymbol{\xi}_{tm}\}$ to $\{\mathbf{P}_{t,m\setminus i}^{-1}, \boldsymbol{\beta}_{t,m\setminus i}, \boldsymbol{\xi}_{t,m\setminus i}\}$

How to delete a $\mathcal{BV}$. First consider the case when we are trying to delete the last one. Take as starting point eqs. (12),(15),(16) and switch the role of old and new: eq. (12) becomes

$$\begin{bmatrix} \mathbf{P}_{t,m-1}^{-1} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} = \mathbf{P}_{tm}^{-1} - \frac{1}{\Delta_b} \begin{bmatrix} \mathbf{w}_b \\ -1 \end{bmatrix} \begin{bmatrix} \mathbf{w}_b \\ -1 \end{bmatrix}^T .$$

Both Δ_b and $\mathbf{w}_b$ can be obtained directly from $\mathbf{P}_{tm}^{-1}$: defining the $(m-1) \times 1$ vector $\mathbf{u}$ by $\mathbf{P}_{tm}^{-1}(1 : m-1, m)$ (i.e. the first $m-1$ rows of the m-th column) and scalar u^* by $\mathbf{P}_{tm}^{-1}(m, m)$ (i.e. the m-th diagonal element) we find $\Delta_b = 1/u^*$ and $\mathbf{w}_b = \mathbf{u}/u^*$. Hence

$$\begin{bmatrix} \mathbf{P}_{t,m-1}^{-1} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} = \mathbf{P}_{tm}^{-1} - \frac{1}{u^*} \begin{bmatrix} \mathbf{u} \\ -1 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ -1 \end{bmatrix}^T \tag{17}$$

where the left side is truncated to yield the $(m-1) \times (m-1)$ matrix $\mathbf{P}_{t,m-1}^{-1}$.

Likewise, to obtain $\boldsymbol{\beta}_{t,m-1}$ from $\boldsymbol{\beta}_{tm}$ we turn around update (15)

$$\begin{bmatrix} \boldsymbol{\beta}_{t,m-1} \\ 0 \end{bmatrix} = \boldsymbol{\beta}_{tm} - \kappa \begin{bmatrix} -\mathbf{w}_b \\ 1 \end{bmatrix} .$$

Again, we can see from update (15) that κ actually is the last component of $\boldsymbol{\beta}_{tm}$. So, defining $b^* = \boldsymbol{\beta}_{tm}(m)$ we get

$$\begin{bmatrix} \boldsymbol{\beta}_{t,m-1} \\ 0 \end{bmatrix} = \boldsymbol{\beta}_{tm} + \frac{b^*}{u^*} \begin{bmatrix} \mathbf{u} \\ -1 \end{bmatrix} \tag{18}$$

where the left side is truncated to yield the $(m-1) \times 1$ vector $\boldsymbol{\beta}_{t,m-1}$.

Finally, to obtain $\xi_{t,m-1}$ from ξ_{tm} we turn around update (16) to yield

$$\xi_{t,m-1} = \xi_{tm} + (b^*)^2/u^*. \tag{19}$$

If we need to delete an arbitrary basis function $i \in \{1, \ldots, m\}$ instead of just the m-th one, we exploit the fact that reordering the indices of the basis function within the set $\mathcal{BV}$ is equivalent to reordering the columns/rows of $\mathbf{P}_{tm}^{-1}$. So, to delete basis function i we just swap column/row i and m in all necessary places (i.e. in $\mathbf{P}_{tm}^{-1}, \boldsymbol{\beta}_{tm}, \mathbf{K}_{mm}$ and $\mathcal{BV}$). Afterwards we apply (17),(18),(19) as described above. Overall, deleting a basis function requires $\mathcal{O}(m^2)$ operations.

When to delete a $\mathcal{BV}$. To identify from the $\mathcal{BV}$ set the basis function best suited for removal we consider their contribution to the cost function. Compute as in (19) the score

$$\varepsilon_i = \frac{\boldsymbol{\beta}_{tm}(i)^2}{\mathbf{P}_{tm}^{-1}(i,i)} \qquad i = 1, \ldots, m$$

for every basis function in $\mathcal{BV}$ and delete the one with the lowest score. The computation of this criterion is very cheap and requires only $\mathcal{O}(m)$ operations.

4 Experiments

Comparing subset selection criteria. First, we compare our supervised approach with the unsupervised method used in the related KRLS algorithm [4]. As third competitor we consider greedy forward selection via orthogonal least squares (OLS). All three methods use the same dictionary of basis function candidates (built from RBF-kernels centered on the training data) to choose the subset from; note though that OLS is a batch method, whereas our method and KRLS process the data sequentially. We chose three well-known problems: the artifical *sinc* data set (noise $\sigma = 0.2$), and the small scale benchmarks *boston* (train 400, test 106) and *abalone* (train 3000, test 1177) from the UCI repository [3]. The data was scaled to have zero mean and unit variance. Parameters governing subset selection were `TOL1`$=10^{-2}$ and `TOL2`$=10^{-4}$; for OLS we used the GCV as stopping criterion. The remaining parameters were set as in [6]. Since our method and KRLS depend on the ordering of the data we averaged over 100 different permutations of every training set. Table 1 shows the resulting prediction error (MSE) along with the size of the subset. Our method shows a similar performance as KRLS but uses fewer (sometimes far fewer) basis functions.

Table 1. Prediction error (MSE) and number of selected basis functions (given in parentheses) for different subset selection variants

Data set	OLS+GCV(subset)	KRLS (subset)	Our (subset)
Sinc	5.6e-4(10)	9.1e-4±1.5e-4 (14.36)	7.5e-4±3.1e-4 (11.06)
Boston	0.88 (44)	0.65±0.039 (220.65)	0.63±0.2 (59.24)
Abalone	0.35 (62)	0.35±0.014 (124.3)	0.37±0.05 (31.54)

Table 2. Classification error for different sizes of the subset (given in parentheses)

Data set	OLS-59(100)	Our(100)	OLS-59(300)	Our(300)	OLS-59(500)	Our(500)
Forest-10k	22.80±0.18	24.61±2.31	21.26±0.23	22.33±0.51	20.35±0.11	21.77±0.34
Forest-50k	22.63±0.11	23.99±1.96	20.58±0.15	21.93±0.36	19.63±0.12	21.24±0.42
Forest-200k	—	24.49±1.64	—	21.83±0.36	—	21.19±0.30
Forest-500k	—	23.80±0.64	—	21.77±0.39	—	21.17±0.32

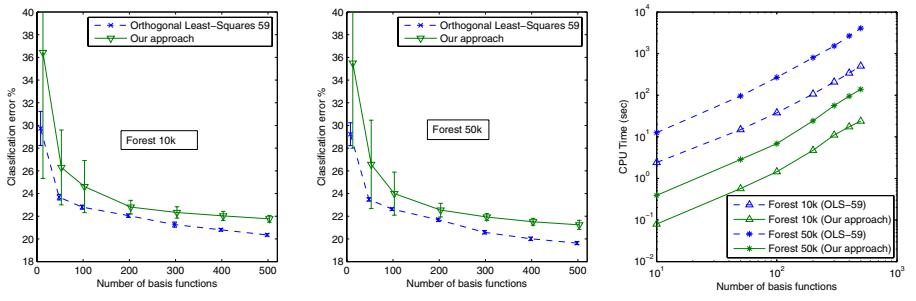


Fig. 1. Comparing our method with OLS

Large-scale real-world benchmark. Though our method is particularly tailored to online learning we show that it is also useful when dealing with large-scale data sets. To this end we chose the biggest data set available from UCI, the data set *forest*. [1] Before we started training we set aside 81,012 randomly chosen examples to serve as independent test set. All of the remaining 500,000 examples were used to train. Since this is a rather large number (for OLS), we also considered smaller training sets of size 10,000, 50,000 and 200,000. In case of OLS we used the 'rule of the 59' [11] heuristic to restrict the search among all remaining candidates to a subset of 59 randomly drawn ones (termed OLS-59). For our approach we set RBF width $\sigma = 1/d$ (with $d = 54$ the input dimensionality), $\gamma = t \cdot 10^{-5}$ (with t being the number of training examples), $\texttt{TOL1} = 10^{-2}$ and $\texttt{TOL2} = 10^{-4}$. The generalization performance and also the CPU time will of course largely depend on the number of basis functions in the model. Hence we examine different models using an increasing number of maximum basis functions. To rule out the influence of randomness each single run was repeated 10 times. Table 2 and Fig. 1 show the achieved classification error (given as percentage of misclassified examples) on the independent test set along with the amount of variation over the different trials (given in parentheses as one standard deviation). Using the same number of basis functions m, we could achieve a classification performance that is comparable with OLS (only slightly worse).

[1] *Forest* is a multi-class classification problem with 581,012 examples and 7 classes. As in [9] we transformed the problem into a two-class classification task: classify class 2 against the rest, which makes the resulting partitions of roughly the same size. *Forest* contains continuous as well as categorical attributes; the latter were transformed via a binary encoding so that the input dimensionality of the problem became $d = 54$. The inputs of the data were scaled to have zero mean and unit variance.

However, our approach being an online method needs far less resources (both CPU-time and memory) to achieve this result (see Fig. 1): the time needed for training is faster at nearly an order of magnitude, while the memory consumption is only $\mathcal{O}(m^2)$ as opposed to $\mathcal{O}(tm)$ when using OLS. In both cases our results are in line with the error rates achieved in the comparable experiments from [9].

5 Conclusion

We presented a subspace based variant of least squares SVM especially geared to online learning. It uses a novel criterion to select a subset of relevant basis functions from the full data set. Experiments indicate that our method improves upon the related KRLS algorithm by choosing a smaller subset and that it can even compete with powerful greedy forward selection; an alternative only amenable to offline learning and at considerably higher computational costs.

References

1. F. R. Bach and M. I. Jordan. Predictive low-rank decomposition for kernel methods. In *Proc. of ICML 22*, 2005.
2. L. Csató and M. Opper. Sparse representation for Gaussian process models. In *NIPS 13*, 2001.
3. C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.
4. Y. Engel, S. Mannor, and R. Meir. The kernel recursive least squares algorithm. *IEEE Trans. on Signal Processing*, 52(8):2275–2285, 2004.
5. S. Fine and K. Scheinberg. Efficient SVM training using low-rank kernel representation. *JMLR*, 2:243–264, 2001.
6. Hoegaerts L., Suykens J.A.K., Vandewalle J., and De Moor B. Subset based least squares subspace regression in RKHS. *Neurocomputing*, 63:293–323, 2005.
7. Z. Luo and G. Wahba. Hybrid adaptive splines. *J. Amer. Statist. Assoc.*, 92:107–114, 1997.
8. J. Platt. A resource-allocating network for function interpolation. *Neural Computation*, 3:213–225, 1991.
9. V. Popovici, S. Bengio, and J.-P. Thiran. Kernel matching pursuit for large datasets. *Pattern Recognition*, 38(12):2385–2390, 2005.
10. J. Quiñonero Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *JMLR*, 6:1935–1959, 2005.
11. A. J. Smola and P. L. Bartlett. Sparse greedy Gaussian process regression. In *NIPS 13*, 2001.
12. A. J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In *Proc. of ICML 17*, 2000.
13. C. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In *NIPS 13*, 2001.

A Nearest Features Classifier Using a Self-organizing Map for Memory Base Evaluation

Christos Pateritsas and Andreas Stafylopatis

School of Electrical and Computer Engineering
National Technical University of Athens
Iroon Polytexneiou 9, 15780 Zografou
Athens, Greece
pater@softlab.ntua.gr, andreas@cs.ntua.gr

Abstract. Memory base learning is one of main fields in the area of machine learning. We propose a new methodology for addressing the classification task that relies on the main idea of the k - nearest neighbors algorithm, which is the most important representative of this field. In the proposed approach, given an unclassified pattern, a set of neighboring patterns is found, but not necessarily using all input feature dimensions. Also, following the concept of the naïve Bayesian classifier, we adopt the hypothesis of the independence of input features in the outcome of the classification task. The two concepts are merged in an attempt to take advantage of their good performance features. In order to further improve the performance of our approach, we propose a novel weighting scheme of the memory base. Using the self-organizing maps model during the execution of the algorithm, dynamic weights of the memory base patterns are produced. Experimental results have shown superior performance of the proposed method in comparison with the aforementioned algorithms and their variations.

1 Introduction

The classification task is one of the most important problems in the area of data mining. For addressing this problem numerous algorithms and methodologies have been proposed. A significant number of these methods belong to the category of memory base learning algorithms. This group of algorithms mainly relies on the concept of nearest neighbor rule. The k–nearest-neighbor (k-NN) algorithm locates a set of neighbors with the smallest distance from the examined data pattern and classifies that pattern according to the majority class of the set of neighbors [1], [4], [15]. Many variations of this concept derive from the use of different distance measures. The most commonly used distance measure is the Euclidean distance, but more sophisticated approaches have also been tested [20].

Another well-known approach in the field of classification, with the use of the theory of probabilities, is the naïve Bayesian classifier [10], [23]. This method is based on the estimation of the posterior probability of a data pattern to belong to a specified class by calculating the probabilities for each feature value of the input

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 391 – 400, 2006.

pattern. The naïve Bayesian classification rule relies on the assumption that these probabilities are independent to each other and by using this assumption calculates the probability of an unlabeled data pattern to belong to a specific class. VFI and KNNFP [5], [7] are also algorithms that examine each feature value independently and then determine their response by combining the independent results.

Both aforementioned approaches to the classification task have been shown to produce good results in spite of their simplicity. This paper presents a methodology that merges these two approaches. We propose a novel method of finding a set of nearest neighbors and we introduce a more complex way of determining the class of an unlabeled data pattern than the simple majority vote among the neighboring data patterns. The algorithm finds data patterns that have the most input values "independently close" to the values of the unlabeled pattern and classify this pattern accordingly. Namely, from the feature values of all data patterns in the memory-base the algorithm finds the values that have the smallest difference to the input data pattern feature values and afterwards locates which of these values belong to the same patterns of the memory-base. This process results in finding the neighbors of the input pattern but not necessarily using all the input features. This differs from a weighted calculation of the nearest neighbors because in that case the weights of the features are a priori set and standard for all input data patterns. A preliminary study on this concept was presented in [13].

For improving the performance of the nearest neighbors algorithm a weighting scheme must be used. Wide range of methods for weighting the impact of the input feature values in the calculation of the distance have been proposed [3], [7], [19]. Reduction techniques for excluding data patterns from the memory-base have been suggested in order to reduce the space required to store data patterns, accelerate the classification process and also increase the overall classification performance. An extensive review of these techniques can be found in [21].

Self-organized Maps (SOMs) employ an unsupervised learning algorithm that achieves dimensionality reduction by compressing the data to a reasonable number of units (neurons) [9]. The map consists of a grid of units that contain all the significant information of the data set, while eliminating noise data, outliers or data faults. Applications of SOM clustering include, but are not limited to, feature extraction or feature evaluation from the trained map [12],[14] and data mining [18].

In [8] the Self-organizing maps were used for editing of the memory base in order to accelerate the classification task without decreasing the classification performance. In our approach the self- organizing maps are used in order to organize and evaluate different feature combinations and provide a dynamic weighting scheme in which, pattern weights are produced during the execution of the classification task described above. In a preprocessing stage, using leave one out cross-validation testing the feature combinations used for classification are gathered and assemble a new labeled data set. This data set of feature combinations forms a new memory base but this memory base is edited with the use of a self-organizing map. The map is then used to provide evaluation for the feature combinations that will appear during the classification of new, unclassified data patterns.

In Section 2 of this paper we describe the classification method, in Section 3 we explain how we incorporated the self-organizing maps for the evaluation of feature combinations. Section 4 includes the results from our experimental study and finally in Section 5 we conclude this paper with remarks about the method and future work.

2 Proposed Methodology

In our approach, the labeled data patterns that are similar to an unlabeled input data pattern are discovered, by using only a number of feature values, which are independently close to the unclassified pattern. Four factors have an impact on the classification outcome.

- The maximum number of feature values of a data pattern from the memory-base that are close to an unclassified data pattern's feature values.
- The number of patterns that achieve this maximum number with respect to the unclassified data pattern.
- The mean Euclidean distance of these data patterns from the unclassified pattern.
- The mean difference of independent feature values between the patterns of a class and the unclassified pattern.

In addition to the classification procedure, we incorporate to our methodology another procedure that has been developed for evaluating the memory-base data patterns, so as to eliminate outliers and possible noisy data.

Let $\vec{x}$ be an unclassified data pattern and $\vec{y}$ a data pattern belonging to set *D*, which is the set of data patterns of the memory-base. Similarity between these patterns is calculated by counting the number of their feature values, such that the difference between these values is below a confidence factor. This number represents the count of feature values of the first pattern that "resemble" the feature values of second pattern. It is the *Count of Confident Features (CCF)* value between patterns $\vec{x}$ and $\vec{y}$ and is the sum of the values of a kernel function *W* over all their input features:

$$CCF(\vec{x},\vec{y}) = \sum_i W(x_i, y_i), \tag{1}$$

where x_i and y_i are the values of the *i-th* feature of patterns $\vec{x}$ and $\vec{y}$ respectively. Apparently, the maximum value for this number is the total number of input features.

Consider the kernel function $W(x_i, y_i)$ defined as:

$$W(x_i, y_i) = \begin{cases} 1, & if \quad 1 - \dfrac{|x_i - y_i|}{width_i} >= Confidence \\ 0, & otherwise \end{cases}. \tag{2}$$

Measures of spread, such as the standard deviation of the values of feature *i* in the set *D*, can be used as $width_i$. The *width* is used for normalization of the feature values to allow a common *Confidence* factor to be used for all features. The maximum value for *Confidence* is 1 and in that case the algorithm selects only the feature values that are identical to x_i. Our experimental study has shown that smaller *Confidence* values (between 0.2 and 0.6) result to the best classification outcomes. Function $W(x_i, y_i)$ resembles the simple kernel function, known as Parzen window, used in kernel based estimation of probability density functions [11]. The first condition of (2) could be enhanced in order to limit even more the number of feature values that are considered

close enough to the values of the input data pattern. This is done by calculating the value of function *W* for all patterns in the memory-base and considering only the *k* smaller differences as being close enough. The modified condition is:

$$1-\frac{\left|x_i - y_i\right|}{width_i} \geq Confidence \wedge y_i \in \underset{k}{G}^i(x_i), \tag{3}$$

where $\wedge$ corresponds to the *and* logical operator and $\underset{k}{G}^i(x_i)$ is the set containing only the *k* nearest values to x_i of the feature *i* from the patterns of set *D*. Its use corresponds to the *k* value in the k-nearest-neighbors algorithm. Both the *Confidence* factor and the *k* parameter serve the same role, setting the boundaries in order to limit the individual feature values that are considered similar to the feature value of $\vec{x}$, but their values are inversely proportional to that. Small values of *k* limit the possible similar feature values, whereas small values of *Confidence* allow more feature values to be considered. The use of the confidence condition allows variable number of independently close values for each feature dimension. In other words, the kernel defined by function *W* is the intersection of two kinds of kernels. One fixed width simple kernel and a variable width kernel where its width is adapted in order to include the *k* nearest values.

In order for pattern $\vec{x}$ to be classified to one of the *c* classes, the *maximum CCF (MaxCCF)* between the pattern $\vec{x}$ and the patterns of each class is calculated:

$$MaxCCF_j(\vec{x}) = \max_{y \in Dj}\left(CCF(\vec{x},\vec{y})\right), \tag{4}$$

where *j* is one of the *c* classes and D_j is the subset of *D* whose patterns belong to class *j*. This number represents the maximum number of feature value differences that are below the confidence factor, between the pattern $\vec{x}$ and each pattern of a class.

It is imperative not only to find the maximum number of features of pattern $\vec{x}$ that are considered close to one or more patterns among all patterns of a specific class, but also to take into account the count of these patterns. More formally, the number of patterns of a class *j*, such that the *CCF* value between them and the unclassified pattern $\vec{x}$ is equal to the *MaxCCF* value between $\vec{x}$ and class *j*. This factor is the *Count of Similar Patterns (CSP)*:

$$\begin{aligned} CSP(\vec{x}) &= \left|K_j^x\right| \\ K_j^x &= \left\{\vec{y} \in D_j : CCF(\vec{x},\vec{y}) = \left(MaxCCF_j(\vec{x}) - m\right)\right\}, m = 0,1,...,n \ , \end{aligned} \tag{5}$$

where $|\,.\,|$ denotes the cardinality of a set and K_j^x is the set of nearest neighbors of pattern $\vec{x}$ among the patterns of class *j*, as it is defined by our methodology. If computed over all classes then the resulting set will be the total set of neighbors. The *m* parameter is an offset used to increase the number of data patterns that take part in the decision process by including patterns with *CCF* value smaller than the *MaxCCF*, in addition to the data patterns with *CCF* value equal to *MaxCCF* (which correspond to $m = 0$). The parameter *n* denotes the upper limit of the offset parameter *m*. Experiments have shown that the optimal value for *n* is 2.

Another factor to be considered, in addition to the cardinality of the above set of patterns (the CSP_j value), is the average Euclidean distance of these patterns from the unclassified data pattern. In this calculation, all feature values are used with the purpose of limiting the impact of outliers or noisy data patterns that could have small differences in some of the feature values and very large differences in the rest of the features. The *Average Distance of Similar Patterns (ADSP)* is defined as:

$$ADSP_j(\vec{x}) = \frac{1}{\underset{K_j}{\text{avg}}\left(\left\|\vec{x}, \vec{y}\right\|\right)} \ . \tag{6}$$

where $\| \ . \ \|$ denotes the Euclidean distance and *avg* the average value. We use the inverse of this value, so that all factors computed are proportional to the probability of pattern $\vec{x}$ to belong to a class.

The fourth factor to be considered is the normalized distance between all feature values of patterns belonging to a class and satisfying the conditions of (2) and input pattern $\vec{x}$, independently of the data pattern they belong to. For each class, the average and standard deviation of these differences are calculated in the *All Features Differences (AFD)* factor:

$$AFD_j(\vec{x}) = \left(1 - \text{avg}\left(\frac{|x_i - y_i|}{width_i}\right)\right) \cdot \left(1 - \text{std}\left(\frac{|x_i - y_i|}{width_i}\right)\right) , \tag{7}$$
$$\forall \vec{y} \in Dj, \forall i : \left(w(x_i, y_i) = 1\right)$$

where *std* denotes standard deviation. This factor is used to calculate the overall similarity between the features values of the input pattern and the neighboring set of labeled patterns of each class. Given that the condition $\left(w(x_i, y_i) = 1\right)$ is calculated on each feature value separately, not all feature values of a labeled pattern contribute to this factor. Finally, in order to predict the class of pattern $\vec{x}$, we combine the four factors from (4), (5), (6) and (7):

$$P_j(\vec{x}) = AFD_j(\vec{x}) \cdot ADSP_j(\vec{x}) \cdot \sum_{m=0}^{n} \left((MaxCCF_j(\vec{x}) - m) \cdot (\beta \cdot CSP_j(\vec{x}, m) + 1)\right) \cdot \tag{8}$$

The β parameter is a trade-off parameter for adjusting the influence of the *CSP* factor in the final result. The class with the maximum *P* is selected.

Each factor combined in (8) serves a different role. The combination of the likeliness of the labeled patterns feature values with the unlabeled pattern, the average distance of the nearest patterns as well as the number of these patterns, results in a multilateral approach to the definition of the unknown class in the classification procedure.

3 Memory-Base Evaluation

Although memory-based algorithms are also called “lazy” learning algorithms due to their lack of preprocessing on the labeled data, an important part of the corresponding literature describes preprocessing weighting methods. A possible categorization of these methods refers to whether the method receives feedback from the memory base

algorithm or not [19]. Some of the methods that do receive feedback implement even an iterative learning process in order to improve their weighting schemes [17].

Another categorization of the weighting of the methods can be done on the grounds of the generality of the weighting scheme: starting from a single set of global weights up to weighting schemes that differ among different regions of the space defined by the memory base or even for each pattern of the memory base.

Our methodology implements two different weighting schemes. One procedure for the evaluation of the data patterns of the memory-base has been developed, which aims at increasing the classification performance and handling noisy data patterns and outliers. The second procedure provides a dynamic weighting scheme derived from the core idea of our classification procedure and with use of a Self-Organizing Map.

3.1 Memory Editing

The first procedure belongs to the category of reduction methods of the memory-base algorithms and applies the classification methodology described above using the leave-one-out cross-validation (LOOCV) test.

An evaluation factor for each pattern of the memory-base is calculated. This factor is initially zero for all patterns. When a pattern is classified correctly, then the patterns that voted for it increase their evaluation factor by one. If a pattern is classified wrongly, then the voting patterns decrease their evaluation factor. After the end of the procedure, patterns with negative evaluation factor are excluded from the memory-base. Methods used for this purpose, such as ENN and ALLk-NN, [22],[16] exclude from the memory-base data patterns that during the testing procedure did not get classified correctly. Our method uses the opposite approach; it excludes data patterns that voted for the incorrect classification of other patterns.

3.2 Incorporating Self Organizing Maps

Our proposed methodology uses the combinations of the "confident" features in order to address the classification problem. As in most memory-based learning methods during the classification process a group of patterns take part in the classification procedure. During the same leave-one-out cross-validation test of the first evaluation procedure, our methodology stores the combinations of the "confident" features between the tested pattern and the patterns of the memory base. These combinations derive from the use of the kernel function W defined in equations (2) and (3). For patterns $\vec{x}$ and $\vec{y}$, a vector w is defined as:

$$w_{xy} = (w_1, w_2, ..., w_n) \text{ , } where \;\; w_i = W(x_i, y_i) \;\; , \tag{9}$$

where n is the total number of features. This vector indicates the combination of feature values of $\vec{x}$ that are considered to be similar to the feature values of $\vec{y}$. A vector w of feature combinations is calculated for each pattern belonging to the set defined by equation (5), which are considered to be similar to $\vec{x}$ (the nearest neighbors) according to our methodology. After the end of the classification procedure of all the patterns used in the LOOCV test, these combinations assemble a new data set that will be used to train a self-organizing map.

This procedure aims at finding clusters of similar feature combinations. In most domains with a large number of features a considerable variety of different combinations is observed. The self-organizing map units can quantize these combinations in a smaller number of vectors. The trained map is combined with the outcome produced when these combinations where used in the classification process during the evaluation procedure. This information is used in order to characterize the units of the trained map. Following the training procedure, every combination is assigned to its best matching unit (BMU). Consequently each unit can be labeled from these combinations. Each combination is labeled with the classification outcome (meaning correct or wrong) and the class of the pattern from the memory base that was compared with the unclassified pattern and produced it.

$$label\ of\ w_{xy} = (classification\ result\ for\ \vec{x}, j),\quad \vec{y} \in D_j\ . \tag{10}$$

Consequently, every map unit gathers a number of correct and a number of wrong classifications for each class. An evaluation factor u for the i-th unit with respect to the j-th class is calculated by dividing the number of the correct classifications to the total number of classifications of this class.

$$u_i^j = \frac{\left|\left\{w_{xy} \in U_i, label_w = (correct, j)\right\}\right|}{\left|\left\{w_{xy} \in U_i, \vec{y} \in Dj\right\}\right|}. \tag{11}$$

Where U_i is the set of combinations assigned to the i-th unit of the map. The map can act as an evaluator of feature combinations during the classification process of new unclassified patterns. As described above, for a pattern to be classified a set of nearest neighbors must by found. This set generates a new set of features combinations. These combinations can be labeled with the class label of the pattern that each time is compared with the unclassified pattern. Assigning each combination to its best matching unit of the SOM map provides an estimation of the correctness of the classification if this combination is used. This estimation is based on the past experience of the use of similar combinations in the classification and the experience is represented by the map units and their evaluation factors for each class.

The next step is to take advantage of the map structure during the classification process of a new pattern. In the same way as during the evaluation process, from the set of nearest neighbors that corresponds to a new pattern, a set of feature combinations is also generated. The feature combinations are labeled with the class labels of the patterns of the nearest neighbors as before. The difference in this case is that the classification outcome is unknown and is to be predicted.

Using the trained map, each feature combination is assigned the evaluation factor u of its best matching unit for the corresponding class label of the feature combination.

$$u(w_{xy}) = u_i^j,\ where\ \ w_{xy} \in U_i \wedge \vec{y} \in Dj\ . \tag{12}$$

Computing the average value of the $u(\ )$ for each class provides us with an extra factor to be embedded in the classification procedure (eq. 8).

$$cu_j = avg\left(u\left(w_{xy}\right)\right),\ \forall w_{xy}, \vec{y} \in Dj\ . \tag{13}$$

The evaluation of the feature combinations could also be exploited in another way. In equation (6) the average distance of the nearest neighbors of each class is calculated. This equation could be enhanced with the evaluation factor as a weighting scheme of the distance between the patterns of the memory base and the unclassified pattern. The modified equation will be:

$$ADSP_j(\vec{x}) = \frac{1}{\underset{K_j}{avg}\left(u\left(w_{xy}\right)\cdot\left\|\vec{x},\vec{y}\right\|\right)}. \tag{14}$$

By using the evaluation factor in this way, we provide a dynamic weighting scheme of the patterns of the memory base. This scheme assigns weights to patterns depending on the combination of features that each pattern appears to have similar to the unclassified pattern.

4 Experimental Results

We tested our method on five benchmark problems of real data from the UCI machine-learning repository [2]. The problems belong to different application domains and are mainly characterized by overlapping clusters and a large number of attributes. In all experiments we used 10-fold cross-validation and the results of our method are the average of 10 experiments. We include results from our method with and without the SOM-weighting scheme. We compare the obtained results with the simple k-NN classifier using Euclidean distance metric, as well as variations using different distance metrics, and the naïve Bayes classifier. Results for these algorithms originate from [6], [20]. In these studies no standard deviations where provided.

Table 1. Comparative results

Method	Accuracy (%)($\sqrt{s^2}$)				
	Vehicle	Pima Indians	Breast Cancer	Ionosphere	Image Segmentation
k – NN, Euclidean	70.93	71.09	94.99	86.32	92.86
k – NN, HOEM	70.22	70.31	95.28	86.33	93.57
k – NN, HVDM	70.93	71.09	94.99	86.32	92.86
k – NN, DVDM	63.72	71.89	95.57	92.60	92.38
k – NN, IVDM	69.27	69.28	95.57	91.17	92.86
k – NN, WVDM	65.37	70.32	95.57	91.44	93.33
Bayes	44.20	73.83	96.40	89.45	91.82
Our approach	74.01 (4.92)	73.92 (5.32)	97.02 (1.12)	89.90 (3.85)	**95.58 (1.22)**
Our approach using SOM	**75.12 (4.15)**	**74.75 (5.18)**	**97.28 (1.07)**	**94.02 (3.14)**	94.63 (1.20)

The results from the experiments, using the data sets described above, can be seen in Table 1. In for four of the five date sets used, our method without the weighting scheme outperforms the other algorithms. The use of the weighting scheme improves even more the performance of the method. The controlling parameters of our method used in each case are presented in Table 2. The *Confidence* is always set to 0.5 and the *n*, which controls the upper bound of the *m* is set to 2 for all experiments. The *k*

nearest values parameter is set to larger values than the corresponding *k* parameter in the k-NN algorithm. The value *All* implies that only the *Confidence* parameter was used for limiting the nearest values.

Table 2. Parameters

	Vehicle	Pima Indians	Breast Cancer	Ionosphere	Image Segmentation
k nearest values	215	All	All	165	254
β trade-off parameter	0.1	0.1	0.2	0.1	0.1

5 Conclusion

In this paper we presented a new hybrid approach to the classification task. Our methodology can be categorized as a memory-based classifier and its concept mainly derives from the idea of the k-NN classifier. It also combines elements from the group of probabilistic classifiers that are based on the assumption of the independence of input features, such as naïve Bayes classifiers. The novelty of this approach lies in the way of determining nearest neighbors, which does not simply use a new distance metric, but rather deals with the matter from a new perspective.

Furthermore, we described a methodology to take advantage of the self-organizing maps model in order to provide a dynamic weighting scheme for our approach. The evaluation procedure receives feedback from the classification task so as to push forward feature combinations which demonstrate better classification results.

The results of our experimental study have shown that the proposed approach achieves better results in comparison to similar algorithms, which are further improved with the use of the weighting scheme. This indicates that further study and improvement of our approach can produce even better results.

Future work includes testing larger data sets and also a study of the computational complexity of the method. Moreover, other ways of utilizing the trained self-organizing map can be investigated. For example the visual inspection of the most common feature combinations used in the classification process which can provide assistance to the domain expert regarding the correlation of features in respect to the classification problem.

Acknowledgment

This work was partly supported by Hrakleitos Project. The Project is co-funded by the European Social Fund (75%) and National Resources (25%).

References

1. Aha, D. W., Kibler, D.,Albert, M. K.: Instance-Based Learning Algorithms. Machine Learning, Vol. 6 (1991) 37-66.
2. Blake, C. L., Merz, C. J.: UCI Repository of Machine Learning Databases. http://www.ics.uci.edu/_mlearn/MLRepository.html, University of California, Irvine (1998).

3. Cost, S., Salzberg, S.: A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features. Machine Learning, Vol. 10 (1993) 57-78.
4. Dasarathy, Belur V.: Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. Los Alamitos, CA: IEEE Computer Society Press (1991).
5. Demiroz, G., Guvenir, H. A.: Classification by Voting Feature Intervals. Proceedings of the 9th European Conference on Machine Learning, Prague. (1997).
6. Fan, H., Ramamohanarao, K.: A Bayesian Approach to Use Emerging Patterns for Classification. Proceedings of the 14th Australasian Database Conference, Adelaide (2003)
7. Guvenir, H. A., Akkus¸, A.: Weighted K Nearest Neighbor Classification on Feature Projections. Proceedings of the 12-th International Symposium on Computer and Information Sciences, Antalya, Turkey (1997).
8. Hammerton, J., Erik F. Tjong Kim Sang,: Combining a self-organising map with memory-based learning. Conference on Computational Natural Language Learning (CoNLL) . Toulouse, France, July 6-7 (2001) 9-14.
9. Kohonen, T.: Self-Organizing Maps. Information Sciences. Springer, second edition, (1997)
10. Kononenko, I.: Naive Bayesian classifier and continuous attributes. Informatica, Vol 16,No. 1 (1992) 1-8.
11. Parzen, E.: On estimation of a probability density function and mode. Ann. Math. Statistics, Vol. 33 (1962) 1065-1076.
12. Pateritsas. C., Pertselakis, M., Stafylopatis, A,: A SOM-based classifier with enhanced structure learning. Proceedings of the IEEE International Conference on Systems, Man & Cybernetics. The Hague, Netherlands, 10-13 October (2004) 4832-4837.
13. Pateritsas. C., Stafylopatis, A,: Independent Nearest Features Memory-Based Classifier. International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA 2005). November 28-30. Vienna, Austria. Vol 2, 781-786.
14. Rauber, A.: LabelSOM: On the labeling of self-organizing maps. Proceedings of International Joint Conference on Neural Networks, Washington, DC, (1999)
15. Stanfill, C., Waltz, D.: Toward memory-based reasoning. Communications of the ACM, Vol. 29, No 12 (1986) 1213-1228.
16. Tomek, I.: An experiment with the edited nearest-neighbor rule. IEEE Transactions on Systems, Man, and Cybernetics, Vol 6(6) (1976) 448–452.
17. Tong Xin, Ozturk, P., Gu, Mingyang,: Dynamic feature weighting in nearest neighbor classifiers . Proceedings of 2004 International Conference on Machine Learning and Cybernetics. 26-29 Aug. (2004), Vol. 4, 2406-2411.
18. Vesanto, J,.: Using SOM in Data Mining. Licentiate's thesis in the Helsinki University of Technology. (2000)
19. Wettschereck, D., Aha W. D.: Weighting Features. First International Conference on Case-Based Reasoning, Lisbon, Portugal. Springer-Verlag, 1995, pp. 347-358.
20. Wilson, D. R., Martinez, T. R.: Improved heterogeneous distance functions. Journal of Artificial Intelligence Research, Vol. 6 (1997) 1-34.
21. Wilson, D. R., Martinez, T. R.: Reduction Techniques for Instance-Based Learning Algorithm. Machine Learning, Vol. 38, Kluwer Academic Publishers (2000) 257-286.
22. Wilson, D. L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics, Vol 2(3) (1972) 408–421.
23. Yang, Y., Webb, G. I.: Proportional k-interval discretization for naive-Bayes classifiers. Proceedings of the 12th European Conference on Machine Learning. (2001) 564-575.

A Multisensor Fusion System for the Detection of Plant Viruses by Combining Artificial Neural Networks

Dimitrios Frossyniotis[1], Yannis Anthopoulos[1], Spiros Kintzios[2], Antonis Perdikaris[2], and Constantine P. Yialouris[1]

[1] Agricultural University Of Athens
Informatics Laboratory, Department of Science
Iera Odos 75, 118 55, Athens, Greece
dfros@cslab.ntua.gr, yatho@netonline.gr, yialouris@aua.gr
[2] Agricultural University Of Athens
Laboratory of Plant Physiology and Morphology, Department of Agricultural Biotechnology
Iera Odos 75, 118 55, Athens, Greece
skin@aua.gr

Abstract. Several researchers have shown that substantial improvements can be achieved in difficult pattern recognition problems by combining the outputs of multiple neural networks. In this work, we present and test a multi-net system for the detection of plant viruses, using biosensors. The system is based on the Bioelectric Recognition Assay (BERA) method for the detection of viruses, developed by our team. BERA sensors detect the electric response of culture cells suspended in a gel matrix, as a result to their interaction with virus's cells, rendering thus feasible his identification. Currently this is achieved empirically by examining the biosensor's response data curve. In this paper, we use a combination of specialized Artificial Neural Networks that are trained to recognize plant viruses according to biosensors' responses. Experiments indicate that the multi-net classification system exhibits promising performance compared with the case of single network training, both in terms of error rates and in terms of training speed (especially if the training of the classifiers is done in parallel).

1 Introduction

Several paradigms for multi-classifier systems have been proposed in the literature during the last years. Classifier combination approaches can be divided along several dimensions, such as the representational methodology, the use of learning techniques or the architectural methodology [1], [2]. A major issue in the architectural design of multiple classifier systems concerns whether individual learners are *correlated* or *independent*. The first alternative is usually applied to multistage approaches (such as boosting techniques [3], [4], whereby specialized classifiers are serially constructed to deal with data points misclassified in previous stages. The second alternative advocates the idea of using a committee of classifiers which are trained independently (in parallel) on the available training patterns, and combining their decisions to produce the final decision of the system. The latter combination can be based on two general strategies, namely *selection* or *fusion*. In the case of selection, one or more

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 401 – 409, 2006.

classifiers are nominated "local experts" in some region of the feature space (which is appropriately divided into regions), based on their classification "expertise" in that region [5], whereas fusion assumes that all classifiers have equal expertise over the whole feature space. A variety of techniques have been applied to implement classifier fusion by combining the outputs of multiple classifiers [1], [6], [7], [8].

The methods that have been proposed for combining neural network classifiers can provide solutions to tasks which either cannot be solved by a single net, or which can be more effectively solved by a multi-net system. However, the amount of possible improvement through such combination techniques is generally not known. Sharkey [9], and Tumer and Ghosh [10], [11] outline a mathematical and theoretical framework for the relationship between the correlation among individual classifiers and the reduction in error, when an averaging combiner is used.

When multiple independent classifiers are considered, several strategies can be adopted regarding the generation of appropriate training sets. The whole set can be used by all classifiers [2], [12] or multiple versions can be formed as bootstrap replicates [13]. Another approach is to partition the training set into smaller disjoint subsets but with proportional distribution of examples of each class [12], [14].

The present work introduces a multi-net classifier system for the detection of plant viruses, using biosensors and Artificial Neural Networks (ANNs). The key feature of the method is the combination of specialized Artificial Neural Networks that are trained to recognize plant viruses according to biosensors' responses. Thus, instead of training a single neural network involving a lot of parameters and using the entire training set, neural networks with less parameters are trained on smaller subsets. Through the splitting of the original data, storage and computation requirements are significantly reduced. Moreover, in order to increase the stability and the generalization capability of the classification model we applied a smoothing technique of the data. This approach produces a set of correlated *specialized* classifiers which attack a complex classification problem by applying an appropriate decision combination.

2 Bioelectric Recognition Assay (BERA)

The Bioelectric Recognition Assay (BERA) is a novel technology that detects the electric response of culture cells, suspended in a gel matrix, to various ligands, which bind to the cell and/or affect its physiology. Preliminary studies [15], [16], [17], [18] have demonstrated the potential application of the method for ultra rapid (1-2 minutes), ultra cheap tests for infectious viruses in humans. Assays have been carried in an entirely crude sample and a high sensitivity of the method (0.1 ng) has been indicated, making it an attractive option for routine sample screening that could help reduce the exceeding use of advanced and costly molecular techniques, such as the reverse transcription polymerase chain reaction (RT-PCR).

After producing a series of different sensor generations, BERA sensors were radically redesigned in order to produce a fifth generation which is optimal for diagnostic applications. Fifth generation sensors are extremely miniaturized, consisting of a disposable array of gel beads loaded with cells. They are characterized by a very high degree of reproducibility (>99.9%), extremely low cost and speed of

manufacturing (with a production performance of approx. 1000 sensors per technician per hour). In addition, the duration of the assay has been reduced from approx. 40 seconds to a mere two seconds. A further variation of the method, called the "6th sensor generation" employs 5th generation sensors which contain engineered cells expressing target-specific antibodies on their membrane [19], [20].

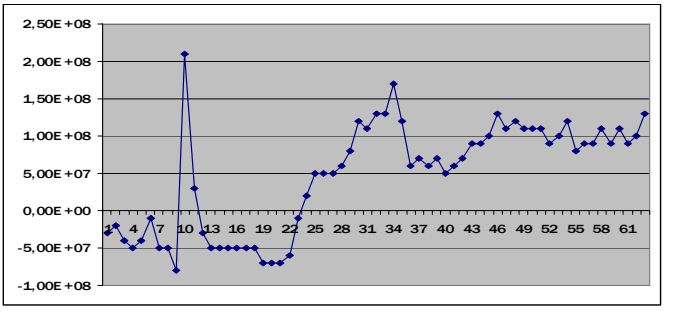


Fig. 1. Time series data produced from the BERA sensors (resampling rate=2)

The major applications of BERA technology are for detection of viruses and metabolic changes linked to disease; and for screening candidate molecules for use as commercial pharmaceutical agents [21]. In this work, BERA will be used to detect plant viruses, such as the tobacco rattle (TRV) and the cucumber green mottle mosaic (CGMMV) viruses, using appropriate plant cells as the sensing elements. In respect to virology applications, each virus demonstrates a unique pattern of biosensor response over a specific range of concentrations, like a *signature*. That is, individual viruses leave a characteristic *signature*, which can be read as a graphical curve, see Figure 1. The units of the X-axis are time stamps and for Y-axis are measurements produced from the sensor respectively.

3 Sensor Fusion

By the term *multisensor fusion* we mean the actual combination of different sources of sensory information into one representational format (this definition also applies to the fusion of information from a single sensory device acquired over a period of time) [22].

The potential advantages of fusing information from multiple sensors are that the information can be obtained more accurately, concerning features which are impossible to perceive with individual sensors, in less time and at a lesser cost. The above correspond respectively to the notions of redundancy, complementarity, timeliness and cost of the information provided to the system. The fusion can take place at different levels of representation (sensory information can be considered as data from a sensor that has been given a semantic content through processing and/or the particular context in which it was obtained). A useful categorization is to consider multisensor fusion as taking place at the signal, pixel, feature and symbol levels of representation. Signal level fusion refers to the combination of the signals provided by different sensors in order to provide a signal that is usually of the same form but of

higher quality. The sensor signals can be represented by random variables corrupted by uncorrelated noise, with the fusion process considered as an estimation procedure.

During the multisensor fusion process three possible problems can be encountered: a) error in fusion process: the major problem in fusing redundant information is that of "registration", i.e. the determination that the information from each sensor is referring to the same features in the environment, b) error in sensory information: the error in sensory information is generally assumed to be caused by a random noise process that can be modeled as a probability distribution, c) error in system operation: when error occurs during operation, it may still be possible to make the assumption that the sensor measurements are independent, if the error is incorporated into the system model through the addition of an extra state variable.

4 Data Pre-processing

In our system, the measurements produced from the sensors are time series data, see Figure 1. So, given a sequence of measurements, we can apply a smoothing technique, like resampling, to extract the necessary features. According to the resampling rate we define the number of the produced features and also the dimensionality of the problem. Furthermore, noise accompanies almost every real measurement and the presence of noise also affects the similarity significantly. Using smoothing techniques like a good resampling rate we can produce better quality of data without a considerable loss of information, see Figure 2.

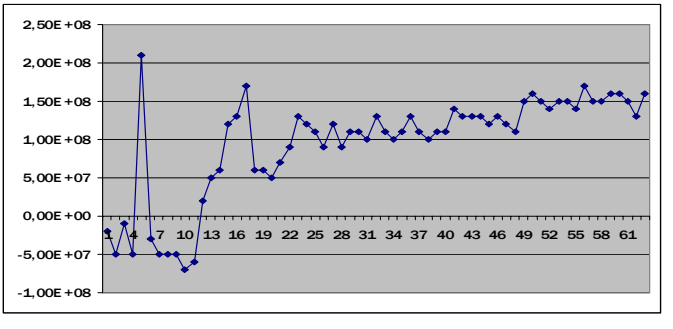


Fig. 2. Time series data produced from the BERA sensors (resampling rate=6)

5 Artificial Neural Networks (ANNs)

In what concerns the classification module, the primary idea is to train a neural classifier, in particular a multi-layered perceptron (MLP), to predict the presence of a virus. We applied Artificial Neural Networks (ANN) with different architecture in order to develop an intelligent system using biosensors for the detection of plant viruses.

It has been shown that a monolithic approach to challenging data mining problems is ineffective. Especially, in the domain of classification, a multiclassifier system can exhibit shorter training time and higher accuracy than a single classifier. Furthermore,

the multiple classifier system might be easier to understand and maintain. So, in order to increase the classification stability of the method and the generalisation performance, we used a combination of neural classifiers.

In this work, the proposed classification system will be used to detect plant viruses, such as the tobacco rattle (TRV) and the cucumber green mottle mosaic (CGMMV) viruses, using appropriate plant cells as the sensing elements. For the training of a classification model we used 200 different examples (plant cells), 100 examples for each plant virus. In addition, the proposed intelligent system is composed of three different types of BERA sensors according to the target-specific antibody that is contained on their membrane. So, for each example we get three different biosensor's response data curves. In particular, for each virus example we can get three different data measurements corresponding to three different patterns. In order to build a classification module to solve a two-class problem we employed two methods. First, we used a simple classifier trained with all the data set consisted of 600 patterns (300 patterns for each plant virus). Next, we used a multi-net classification system composed of three classifiers, each of them specialized with a specific type of BERA sensor. The later is accomplished by training each classifier with patterns produced from the corresponding sensor only.

5.1 Multilayer Perceptron

We considered MLP architectures consisting of the input layer (number of units according to the resampling rate), one hidden layer (20 to 30 sigmoid hidden units) and two output units (two viruses). We have applied the BFGS quasi-Newton algorithm [23] to train the MLP using the early stopping technique. Weights were randomly initialised in the range [-1, 1].

In our experimental study we want to discover the appropriate resampling rate and the MLP architecture (number of hidden units) that gives us the best results. To accomplish that we trained and tested several neural networks with different architectures and we also used several resampling rates to produce training data sets with different dimensionality.

5.2 Combination of MLPs

In this work, we combine specialized Artificial Neural Networks that are trained to recognize plant viruses according to biosensors' responses. In this sense, we produced a set of correlated classifiers which attack the classification problem. The latter is accomplished by splitting the original problem into subproblems, which are assigned to the single classifiers. Subsequently, the multiclassifier system combines the performance of multiple single classifiers so as to reach a global decision. A major issue here is the way of combining those individual decisions. In this work, we followed the simplest method witch is the scheme of the *majority wins*.

An important advantage of this method is that the training of each subnetwork can be done separately and in parallel. Thus, in the case of parallel implementation, the total training time of the system equals to the worst training time achieved among the neural classifiers. It must be noted that this total training time cannot be greater than the training time of a single neural classifier of the same type dealing with the entire

training set. Since such a single network usually requires more parameters, to learn the whole data set (which is much larger), the multi-net approach may lead to reduced execution times even in the case of implementation on a single processor.

In particular, each subnetwork is a fully connected multilayer perceptron (MLP), with one hidden layer of sigmoidal units. We have applied the BFGS quasi-Newton algorithm [22] to train the MLPs using the early stopping technique. The classifications produced by the multiple individual MLPs are appropriately combined to get the final decision.

6 Experimental Evaluation

To compare the different network architectures, several series of experiments had to be conducted. For each type of MLP, we employed the 10-cross-validation method, in particular, ten experiments were performed with splits of data into training and test sets of fixed size (70% for training and 30% for testing). The effectiveness of generalization can be expressed as the ratio of the correctly recognized input patterns to the total number of presented patterns during the test phase. The average generalization results were calculated from these ten trials and the best results are summarized in Table 1.

Table 1. Average generalization results using BFGS quasi-Newton algorithm for training MLP. For each case the resampling rate is indicated in parentheses.

MLP Number of units in the hidden layer	BFGS (2)	BFGS (4)	BFGS (6)	BFGS (8)
20	75.8%	84.3%	86.3%	78.9%
25	80.1%	86.1%	87.7%	81.2%
30	85.9%	87.2%	90.3%	81.7%

Next, in Table 2 we give the best average generalization results produced from the multi-net classification system. Each subclassifier of the proposed multi-system is an ANN specialized to a biosensor. Given a new unclassified pattern, a class label is produced from each subclassifer. The final decision of the system is produced using a simple voting scheme, named the majority wins. We used different architectures for each subclassifier and the best results are shown in Table 2.

Table 2. Average generalization results using the multi-net classification system. For each case the resampling rate is indicated in parentheses.

Multi-net System Number of units in the hidden layer for each subclassifier	BFGS (2)	BFGS (4)	BFGS (6)	BFGS (8)
(10,10,10)	85.8%	92.3%	92.6%	88.9%
(10,15,20)	90.1%	91.5%	94%	91.2%
(20,15,10)	85.9%	93.2%	90.3%	91.7%

Comparing the results in Table 1 with Table 2, we observe that using the proposed multi-net classifier system we can get more robust classification models with better generalization performance. The use of a smoothing technique like a good resampling rate improves even more the performance of the system.

Furthermore, the results of our experimental study have shown that the proposed approach using ANN achieves better results in comparison to the empirical techniques. Also, the corresponding time of the proposed classification system, which is critical in real applications, is very competitive to the time an expert needs, so as to make a decision by examining a data curve.

7 Conclusions

In this work, we applied Artificial Neural Networks (ANN) with different architecture in order to develop an intelligent system using biosensors for the detection of plant viruses. The system is based on already developed by the team method for detection of viruses named BERA. The main drawback of this method was the employment of an empiric way to intact a virus by examining the biosensor's response data curve. To overcome this problem, we used Artificial Neural Networks that are trained and specialized so that they recognize plant viruses. In order to increase the classification stability of the method and the generalisation performance, we proposed a combination of specialized Artificial Neural Networks that are trained to recognize plant viruses according to biosensors' responses. We also used resampling as a smoothing technique to produce better quality of data without a considerable loss of information.

An important strength of the proposed classification approach is that it does not depend on the type of the classifier, therefore, it is quite general and applicable to a wide class of models including neural networks and other classification techniques. The next target of our work will be to train the system to classify human viruses.

Acknowledgement

The project is co-funded by European Social Fund & National Resources – O.P. "Education" II.

References

1. Alpaydin E. Techniques for combining multiple learners. In Proceedings of Engineering of Intelligent Systems, volume 2, pages 6-12, ICSC Press, 1998.
2. Kuncheva L. Combining Classifiers by Clustering, Selection and Decision Templates. Technical report, University of Wales, UK, 2000.
3. Maclin R. and Opitz D. An empirical evaluation of bagging and boosting. In Proceedings of the Fourteenth International Conference on Artificial Intelligence, pages 546-551, AAAI Press/MIT Press, 1997.

4. Freund Y. and Schapire R.E. Experiments with a new boosting algorithm. In Proceedings of the Thirteenth International Conference on Machine Learning, pages 148-156, Morgan Kaufmann, 1996.
5. Kuncheva L. Clustering-and-selection model for classifier combination. In Proceedings of the 4th International Conference on Knowledge-based Intelligent Engineering Systems (KES'2000), Brighton, UK, 2000.
6. Vericas A., Lipnickas A., Malmqvist K., Bacauskiene M. and Gelzinis A. Soft combination of neural classifiers: A comparative study, Pattern Recognition Letters, volume 20, pages 429-444, 1999.
7. Tumer K. and Ghosh J. Classifier combining through trimmed means and order statistics. In Proceedings of the International Joint Conference on Neural Networks, Anchorage, Alaska, 1998.
8. Tumer K. and Ghosh J. Order statistics combiners for neural classifiers. In Proceedings of the World Congress on Neural Networks, pages I:31-34, Washington D.C., INNS Press , 1995.
9. Sharkey A.J.C. Combining Artificial Neural Nets : Ensemble and Modular Multi-Net Systems, Springer-Verlag Press, 1999.
10. Tumer K. and Ghosh J. Limits to performance gains in combined neural classifiers. In Proceedings of the Artificial Neural Networks in Engineering '95, pages 419-424, St. Louis, 1995.
11. Tumer K. and Ghosh J. Error correlation and error reduction in ensemble classifiers. Connection Science, Special Issue on Combining Artificial Neural Networks: Ensemble Approaches, volume 8, number 3-5, pages 385-404, 1996.
12. Alpaydin E. Voting over multiple condensed nearest neighbour subsets. Artificial Intelligence Review, volume 11, pages 115-132, 1997.
13. Breiman L. Bagging predictors. Technical report, no. 421, Department of Statistics, University of California, Berkeley, 1994.
14. Chan P.K. and Stolfo S.J. A comparative evaluation of voting and meta-learning on partitioned data. In Proceedings of the Twelfth International Machine Learning Conference, pages 90-98, Morgan Kaufmann, San Mateo, CA, 1995.
15. Kintzios S., E. Pistola, P. Panagiotopoulos, M. Bomsel, N. Alexandropoulos, F. Bem, I. Biselis, R. Levin. Bioelectric recognition assay (BERA). Biosensors and Bioelectronics 16¨325-336, 2001.
16. Kintzios S., E. Pistola, J. Konstas, F. Bem, T. Matakiadis, N. Alexandropoulos, I. Biselis, R. Levin. Application of the Bioelectric recognition assay (BERA) for the detection of human and plant viruses: definition of operational parameters. Biosensors and Bioelectronics 16: 467-480, 2001.
17. Kintzios, S., Bem, F., Mangana, O., Nomikou, K., Markoulatos, P., Alexandropoulos, N., Fasseas, C., Arakelyan, V., Petrou, A-L., Soukouli, K., Moschopoulou, G., Yialouris, C., Simonian, A. Study on the mechanism of Bioelectric Recognition Assay: evidence for immobilized cell membrane interactions with viral fragments. Biosensors & Bioelectronics 20: 907-916, 2004.
18. Kintzios S., Makrygianni Ef., Pistola E., Panagiotopoulos P., Economou G. Effect of amino acids and amino acid analogues on the in vitro expreesion of glyphosate tolerance in johnsongrass (Sorghum halepense L. pers.) J. Food, Agriculture and Environment 3: 180-184, 2003.
19. Kintzios S., J. Goldstein, A. Perdikaris, G. Moschopoulou, I. Marinopoulou, , O. Mangana, K. Nomikou, I. Papanastasiou, A-L. Petrou V. Arakelyan, A. Economou, A. Simonian. The BERA Diagnostic System: An all-purpose cell biosensor for the 21th Century. 5th Biodetection Conference, Baltimore, MD, USA, 9-10/06/05, 2005.

20. Moschopoulou G., Kintzios S. (2005): Membrane engineered Bioelectric Recognition Cell sensors for the detection of subnanomolar concentrations of superoxide: A novel biosensor principle. International Conference on Instrumental Methods of Analysis (IMA) 2005, Crete, Greece, 1-5/10/2005.
21. Kintzios S., I. Marinopoulou, G. Moschopoulou, O. Mangana, K. Nomikou, K. Endo, I. Papanastasiou, A. Simonian. Construction of a novel, multi-analyte biosensor system for assaying cell division. Biosensors and Bioelectronics.(in press).
22. Tzafestas G.S., Anthopoulos Y., Neural Networks Based Sensorial Signal Fusion: An Application to Material Identification', DSP'97, Santorini, Greece, July 2-4 1997.
23. Dennis J.E., Schnabel R.B., Numerical methods for unconstrained optimization and nonlinear equations. Englewood Cliffs, NJ: Prentice-Hall, 1983.

A Novel Connectionist-Oriented Feature Normalization Technique

Edmondo Trentin

Dipartimento di Ingegneria dell'Informazione
Università di Siena, V. Roma, 56 - Siena, Italy

Abstract. Feature normalization is a topic of practical relevance in real-world applications of neural networks. Although the topic is sometimes overlooked, the success of connectionist models in difficult tasks may depend on a proper normalization of input features. As a matter of fact, the relevance of normalization is pointed out in classic pattern recognition literature. In addition, neural nets require input values that do not compromise numerical stability during the computation of partial derivatives of the nonlinearities. For instance, inputs to connectionist models should not exceed certain ranges, in order to avoid the phenomenon of "saturation" of sigmoids. This paper introduces a novel feature normalization technique that ensures values that are distributed over the $(0, 1)$ interval in a uniform manner. The normalization is obtained starting from an estimation of the probabilistic distribution of input features, followed by an evaluation (over the feature that has to be normalized) of a "mixture of Logistics" approximation of the cumulative distribution. The approach turns out to be compliant with the very nature of the neural network (it is realized via a mixture of sigmoids, that can be encapsulated within the network itself). Experiments on a real-world continuous speech recognition task show that the technique is effective, comparing favorably with some standard feature normalizations.

1 Introduction

Feature normalization, as pointed out in classic pattern recognition literature [4,5,7], is a topic of practical relevance in real-world applications of artificial neural networks (ANN). Although the topic is sometimes overlooked, the success of connectionist models in difficult tasks may depend on a proper normalization of input features. Let us assume that input (or output) patterns are in the form $\mathbf{x} = (x_1, \ldots, x_d)$, and that they are extracted from a d-dimensional, real-valued feature space X. Individual feature values x_i, $i = 1, \ldots, d$, are measurements of certain attributes, according to a problem-specific feature extraction process. Such measurements are expressed, in general, in terms of different units, and the latter ones may span different possible ranges of values. Major motivations for applying a normalization method include the following:

1. Reducing all features $x_1, \ldots, x_d$ to a common range (a, b), where $a, b \in calR$. In so doing, increased homogeneity of values is gained, yielding a common

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 410–416, 2006.

(e.g., Euclidean) "distance measure" over patterns along the different axis. Furthermore, all features are given the same credit, or weight: unnormalized features that span a wider numerical range would otherwise overpower features defined over smaller intervals.

2. Tackling, or reducing, numerical stability problems of the learning algorithms in the ANN (i.e., during the computation of partial derivatives of the nonlinearities of the model). In particular, input values should not exceed a certain (a, b) interval, in order to avoid the phenomenon of "saturation" of sigmoids. As a matter of fact, saturation occurs when the activation value a (input argument) of a sigmoid $f(a)$ is along the tails of $f(.)$, where the partial derivative $\frac{\delta f(a)}{\delta a}$ is numerically null. In case of saturation, the sigmoid is basically "stuck" and it cannot provide any further contribution to the gradient-driven learning of connection weights.
3. Stabilizing the numerical behavior of the delta-rule in the backpropagation algorithm [12]. Since $\Delta w_{ij} = \eta \delta_i f_j(a_j)$ for a generic hidden or output weight, while $\Delta w_{jk} = \eta \delta_j x_k$ for weights in the first (input) layer, and given a common learning rate η, it is seen that unnormalized large-value features x_k would overpower the learning process for input weights w_{jk} w.r.t. the other weights of the ANN.
4. Allowing application of a nonlinear output layer of the ANN to model outputs in a wider range. Actually, sigmoids in the form $\frac{1}{1+e^{-a}}$ are limited to the $(0, 1)$ interval, and hyperbolic-tangent sigmoids range over the $(-1, 1)$ interval, while target outputs may exceed these ranges.
5. Leading to data distributions that are basically invariant to rigid displacements of the coordinates.

Classic normalization techniques mostly rely on the following approaches: (i) for each feature $i = 1, \ldots, d$, find the maximum absolute value M_i (i.e., $M_i \in \mathcal{R}^+$) over the training sample, and normalize each pattern $\mathbf{x}$ to obtain a new pattern $\mathbf{x}'$ defined as $\mathbf{x}' = (x_1/M_1, \ldots, x_d/M_d)$. This ensures features within the $(-1, 1)$ range. A similar technique is described in [2]; (ii) compute the sample mean m_i and the sample variance s_i for each feature $i = 1, \ldots, d$, and normalize $\mathbf{x}$ to obtain $\mathbf{x}' = (\frac{x_1 - m_1}{s_1}, \ldots, \frac{x_d - m_d}{s_d})$. This ensures zero mean and unit variance along all coordinate axis of the normalized feature space [7]. Approaches (i) and (ii), i.e. mean subtraction and division by maximum, are sometimes combined.

Other (often similar) approaches can be found in the literature. For instance, [6] presents an algorithm based on a heterogeneity measure, while [8] proposes a combined normalization/clustering procedure. Different methods rely on linear projections, e.g. the eigenvector projection or Karhunen-Loeve method [7], where the original features are linearly transformed to a lower dimensionality space. These transformations imply a certain loss of information w.r.t. the original feature space representation of patterns.

This paper introduces a novel feature normalization technique that ensures values that are distributed over the $(0, 1)$ interval in a uniform manner. The technique is inspired by an approach suggested by Yoshua Bengio[1], who used the

[1] Y. Bengio, personal communication to the author (Como, Italy, 2000).

rank of discrete observations as their numeric feature value. The normalization is obtained starting from a maximum-likelihood estimation of the probabilistic distribution of input features, according to a particular parametric model. The technique is described in detail in Section 2. In addition to the above-listed benefits, the technique turns out to be compliant with the very nature of the ANN (it is realized via a mixture of sigmoids, that can be encapsulated within the ANN itself). The uniform distribution obtained for the normalized data is basically sample-invariant, i.e. the ANN architecture and training parameters (e.g. the learning rate) are expected to fit different datasets. An experimental evaluation on a real-world, continuous speech recognition task (Section 3) shows that the technique is effective, and more suitable to the task than classic normalization approaches.

2 The Proposed Normalization Method

Normalization is accomplished by transforming individual components of each input pattern into the corresponding value of the cumulative distribution function (*cdf*) of the inputs, estimated on a feature-by-feature basis. A mixture of Logistics is assumed as a model of the *cdf* for a given component of the feature space. Maximum-likelihood estimation of the mixture parameters is performed from the available samples [4]. Each feature is eventually normalized by replacing it with the value of the corresponding *cdf* evaluated over the feature itself. In so doing, the normalization step can be encapsulated within the ANN, in a suitable and straightforward manner, by adding an extra (pre)input layer with sigmoid activation functions. The proposed approach leads to (potentially) sample-invariant ANN topologies/learning parameters, since different data sets are reduced to the same distribution.

More precisely, let $\mathcal{T} = \{\mathbf{x}\}$ be the data sample, where $\mathbf{x} \in \mathcal{R}^d$ are the patterns to be normalized. The i-th feature x_i is drawn from a certain (generally unknown) probability distribution, having probability density function (*pdf*) $p_i(x)$. We assume a parametric model for $p_i(x)$ in the form of a mixture of Normal components, i.e. $p_i(x) = \sum_{j=1}^{c} P_i(j) N(x; \mu_{ij}, \sigma_{ij}^2)$, where c Normal densities $N(x; \mu_{ij}, \sigma_{ij}^2)$ (with mean μ_{ij} and variance σ_{ij}^2) are considered, along with their corresponding mixing parameters $P_i(j)$. The condition $\sum_{j=1}^{c} P_i(j) = 1$ holds. Although the assumption of a specific parametric from for $p_i(x)$ may look strong, the mixture of Normal components is popular in statistical inference. In fact, it may approximate arbitrarily well any given continuous *pdf* if c is sufficiently large [4].

Once the parametric model has been fixed, any given statistical technique for parameter estimation may be applied in order to estimate $P_i(j)$, μ_{ij}, and σ_{ij}^2 from the sample $\mathcal{T}$, for $i = 1, \ldots, d$ and $j = 1, \ldots, c$. Parameter estimation approaches include the maximum-likelihood (ML) technique [4], Bayesian learning [4], and the minimax procedure [10,9]. ML is used in the experiments presented in this paper.

At this point, let us assume that the parameters of the model $p_i(x) = \sum_{j=1}^{c} P_i(j) N(x; \mu_{ij}, \sigma_{ij}^2)$ have been determined from the data in a suitable manner. The corresponding *cdf* $f_{p_i}(x)$ is defined as $f_{p_i}(x) = \int_{-\infty}^{x} p_i(u) du$, hence

$f_{p_i}(x) = \sum_{j=1}^{c} P_i(j) \int_{-\infty}^{x} N(u; \mu_{ij}, \sigma_{ij}^2) du$. It is easily seen that, according to [10], the indefinite integral cannot be expressed in a simple functional form, but the *cdf* of a Normal *pdf* $N(x; \mu, \sigma^2)$ has a sigmoid-like shape that is close to a logistic function $F(x) = 1/(1 + e^{-(x-\alpha)/\beta})$ having mean $\alpha = \mu$ and variance $\sigma^2 = \frac{\beta^2 \pi^2}{3}$. The ML estimates for μ and σ^2 can be used to compute α and β accordingly. Thus, we can obtain an approximation of $f_{p_i}(x)$ by assuming a mixture of logistics, namely $f_{p_i}(x) \simeq \sum_{j=1}^{c} P_i(j)\{1/(1 + e^{-(x-\mu_{ij})/\sqrt{3\sigma_{ij}^2/\pi^2}})\}$.

Finally, transformation of the unnormalized feature vector $\mathbf{x} = (x_1, \ldots, x_d)$ into a normalized feature vector $\mathbf{x}'$ is accomplished as follows:
$\mathbf{x}' = (f_{p_1}(x_1), f_{p_2}(x_2), \ldots, f_{p_d}(x_d))$. Due to the properties of *cdfs*, it is immediately seen that the normalized features are uniformly distributed over the $(0, 1)$ interval.

It is worth pointing out that the logistic $F(x)$ is basically a standard ANN sigmoid, with bias α and smoothness β. As a consequence, the normalization step can be encapsulated within the ANN by adding an extra (pre)input layer with sigmoid activation functions $1/(1 + e^{-(x-\mu_{ij})/\sqrt{3\sigma_{ij}^2/\pi^2}})$ which feed connection weights that are set equal to the mixing parameters $P_i(j)$. This further emphasizes the numerical stability of the proposed approach, since all nonlinearities in the overall model are the same in nature and in numerical properties.

In so doing, the normalization transformation may be also refined by means of a few, further learning epochs via backpropagation. This is accomplished through a 3-step procedure: (1) estimate the *cdfs* as above, and apply normalization; (2) train the ANN over the normalized data; (3) keeping the weights of the ANN fixed, apply encapsulation of the mixture of logistics within the connectionist architecture, and apply backpropagation to improve the values of the weights $P_i(j)$, as well as of the bias μ_{ij} and smoothness σ_{ij} (i.e., learn a gradient-driven normalization transformation that better fits the data and the overall training criterion of the ANN).

3 Experimental Evaluation

We evaluate performance of the proposed technique in a real-world continuous speech recognition task. The ANN/hidden Markov model hybrid that we introduced in [13] is used for carrying out the recognition task. It relies on a feed-forward ANN that learns to estimate the emission probabilities of a hidden Markov model (HMM) [11] according to the ML criterion [14]. Performance obtained with/without different feature normalization techniques is reported, and compared with a baseline yielded by a standard Gaussian-based HMM, and with the results obtained using a classic Bourlard and Morgan's ANN/HMM paradigm [1], where the ANN is heuristically trained to estimate the conditional transition probabilities.

The recognition task is the same that we discussed in [14]. Speech signals from the `cdigits` part of the *SPK* database[2], collected in laboratory conditions, were

[2] SPK is available from the European Language Resources Association (ELRA).

considered. It is a continuous speech task, namely 1000 utterances of connected Italian digit strings having length 8 (for a total of 8000 words), acquired over 40 different speakers (21 male and 19 female). The whole dataset was divided into two equally-sized subsets, to be used for training and test, respectively. A close-talk microphone was used for the recordings, under quiet laboratory conditions. Spectral analysis of the speech signals (acquired at a sampling rate of 16kHz) was accomplished over 20ms Hamming windows having an overlap of 10ms, in order to extract 8 *Mel Frequency Scaled Cepstral Coefficients* (MFSCCs) [3] and the signal log-energy as acoustic features (i.e., $d = 9$). The unnormalized values of this feature space roughly range in the $(-198, 246)$ interval.

Words in the dictionary were modeled using individual, left-to-right HMMs having 3 to 6 states, according to the phonetic transcription of each Italian digit, plus a 1-state model for the "silence" (or "background noise", *@bg*) for a total of 40 states. Mixtures of 8 Gaussian components were used to model emission probabilities for each state of the standard HMM, that was initialized via *Segmental k-Means* [11] and trained by applying the Baum-Welch algorithm [11]. After a preliminary cross-validation step, the topology of the feed-forward ANN included 9 inputs, 93 sigmoids in the hidden layer, and 40 output sigmoids (one per each state of the underlying HMM). This architecture was kept fixed during all the following experiments.

Table 1. Word recognition rate (WRR) on test set of the SPK connected digits, speaker-independent problem. 9-dim acoustic space: 8 MFSCCs and signal log-energy.

Architecture/normalization technique	*WRR (%)*
HMM with 8-Gaussian mixtures, no norm	90.03
Bourlard and Morgan's, no norm	46.75
Bourlard and Morgan's, division by max	88.01
Bourlard and Morgan's, mean-variance norm	89.16
Bourlard and Morgan's, proposed norm	90.20
ANN/HMM hybrid, no norm	52.16
ANN/HMM hybrid, division by max	89.92
ANN/HMM hybrid, mean-variance norm	92.05
ANN/HMM hybrid, proposed norm	94.65

Experimental results with/without feature normalization are reported in Table 1 in terms of *Word Recognition Rate* (WRR). In the Table, "no norm" means that no normalization was applied to the original features; "division by max" means that each feature value was divided by the maximum absolute value (for that specific feature) that was met on the training sample; "mean-variance norm" refers to the mean-subtraction and division-by-variance normalization scheme (as described in Section 1); "proposed norm" is the normalization technique presented in this paper.

The Gaussian-based HMM is best suited to unnormalized data (results obtained using normalized data are worse, and they are not reported in the Table).

A comparison with Bourlard and Morgan's architecture is provided. The variant "Case 4" of the algorithm, described in [1] on page 164, was applied. Normalization of ANN outputs, i.e. division by the corresponding state-priors (estimated from the training set) in order to reduce to "likelihoods", was accomplished at recognition time, as recommended in [1], Chapter 7. Finally, the ANN/HMM hybrid proposed by [14] was applied. As already reported in [14], the ANN/HMM hybrid (trained with the *MAP* version of the gradient-ascent, global optimization algorithm [14]) improves performance over the other recognition paradigms. In any case, it is seen that the proposed normalization technique is particularly effective for ANN training in the speech recognition task.

4 Conclusion

The success of connectionist models in difficult tasks may depend on a proper normalization of input features. Moreover, ANNs require input values that do not compromise numerical stability during the computation of partial derivatives of the nonlinearities, as well as inputs that do not exceed certain ranges in order to avoid the phenomenon of "saturation" of sigmoids. This paper introduced a novel feature normalization technique that ensures values that are distributed over the $(0, 1)$ interval in a uniform manner. The normalization is obtained starting from a ML estimation of the probabilistic distribution of input features according to a parametric mixture density model, turning out to be compliant with the very nature of the data. A mixture of Logistics is then used to approximate the corresponding cumulative distribution $f_{p_i}(x)$ (it is realized via a mixture of sigmoids, that can be encapsulated within the ANN itself). Each feature x_i is then normalized by taking $x'_i = f_{p_i}(x_i)$.

Experiments were accomplished on a speaker-independent, continuous speech recognition task from the *SPK* database. Hybrid ANN/HMM models were applied, including ANNs that estimate probabilistic quantities involved in the underlying HMM. Comparison of the results obtained with the different models with/without major feature normalization methods show that the proposed technique is effective.

The proposed approach leads to (potentially) sample-invariant ANN topologies/learning parameters, since different data sets are reduced to the same distribution.

References

1. H. Bourlard and N. Morgan. *Connectionist Speech Recognition. A Hybrid Approach*, volume 247. Kluwer Academic Publishers, Boston, 1994.
2. J.W. Carmichael, J.A. George, and R.S. Julius. Finding natural clusters. *Systematic Zoology*, 17:144–150, 1968.
3. S. B. Davis and P. Mermelstein. Comparison of parametric representations of monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 28(4):357–366, 1980.

4. R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.
5. K. Fukunaga. *Statistical Pattern Recognition*. Academic Press, San Diego, second edition, 1990.
6. A.V. Hall. Group forming and discrimination with homogeneity functions. In A.J. Cole, editor, *Numerical Taxonomy*, pages 53–67. Academic Press, New York, 1969.
7. A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, 1988.
8. V. Lumelsky. A combined algorithm for weighting the variables and clustering in the clustering problem. *Pattern Recognition*, 15:53–60, 1982.
9. N. Merhav and C. H. Lee. A minimax classification approach with application to robust speech recognition. *IEEE Transactions on Speech and Audio Processing*, 1:90–100, January 1993.
10. A.M. Mood, F.A. Graybill, and D.C. Boes. *Introduction to the Theory of Statistics*. McGraw-Hill International, Singapore, 3rd edition, 1974.
11. La. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
12. D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing*, volume 1, chapter 8, pages 318–362. MIT Press, Cambridge, 1986.
13. E. Trentin and M. Gori. Continuous speech recognition with a robust connectionist/markovian hybrid model. In *Proceedings of ICANN2001, International Conference on Artificial Neural Networks*, Vienna, Austria, August 2001.
14. E. Trentin and M. Gori. Robust combination of neural networks and hidden Markov models for speech recognition. *IEEE Transactions on Neural Networks*, 14(6), november 2003.

An Evolutionary Approach to Automatic Kernel Construction

Tom Howley and Michael G. Madden

National University of Ireland, Galway
thowley@vega.it.nuigalway.ie, michael.madden@nuigalway.ie

Abstract. Kernel-based learning presents a unified approach to machine learning problems such as classification and regression. The selection of a kernel and associated parameters is a critical step in the application of any kernel-based method to a problem. This paper presents a data-driven evolutionary approach for constructing kernels, named KTree. An application of KTree to the Support Vector Machine (SVM) classifier is described. Experiments on a synthetic dataset are used to determine the best evolutionary strategy, e.g. what fitness function to use for kernel evaluation. The performance of an SVM based on KTree is compared with that of standard kernel SVMs on a synthetic dataset and on a number of real-world datasets. KTree is shown to outperform or match the best performance of all the standard kernels tested.

1 Introduction

A major advance in recent research into pattern analysis has been the emergence of an approach known as kernel-based learning. This unified approach to problems, such as classification, regression and clustering, is based on a kernel that defines how two objects of a dataset are related. Kernel-based learning first appeared in the form of support vector machines, a powerful classification algorithm that is capable of representing non-linear relationships (via kernels) and producing models that generalise well to unseen data. A key decision in the use of any kernel-based method is the choice of kernel. In the case of SVMs, the performance exhibited by different kernels may differ considerably. Generally, kernel method practitioners will pick from a set of standard kernels, the Radial Basis Function (RBF) and Polynomial kernel being two widely used examples. An alternative to using one of these pre-defined kernels is to construct a custom kernel especially for a particular problem domain, e.g. the string kernel used for text classification [1]. This approach can yield good results, but obviously depends on the availability of expert knowledge of a particular domain.

This paper presents an approach, named KTree, that uses the evolutionary method of Genetic Programming (GP) to find a kernel for a particular data domain. KTree is a modified and extended version of the Genetic Kernel SVM (GKSVM) developed by the authors [2]: it uses a more sophisticated kernel representation that can represent standard kernels, such as RBF; a Mercer filter is used to improve performance; it uses a different fitness function (based on cross-validation), which results have shown to be superior. This study also includes a more extensive evaluation, using both a synthetic dataset and wider range of real-world datasets. KTree allows for the generation of non-standard kernels; the objective is to provide for the automatic discovery of kernels that

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 417–426, 2006.

achieve good classification accuracy when tested on unknown data. The major goal of this research is to determine the best strategy in the use of GP to evolve kernels; key issues include choice of fitness function and the filtering of non-Mercer kernels.

Kernel methods are described in Section 2, with particular emphasis on kernel functions. Section 3 describes KTree. Experimental results and analyses are presented in Section 4. Section 5 evaluates research related to this work and Section 6 presents the main conclusions.

2 Kernel Methods and Classification

In kernel methods, the kernel function is used to recode the data into a new feature space that reveals regularities in the data that were not detectable in the original representation. This allows the use of algorithms based on linear functions in the feature space; such linear methods are both well understood and computationally efficient. With kernel functions, no explicit mapping of the data to the new feature space is carried out – this is known as the "kernel trick". It enables the use of feature spaces whose dimensionality is more than polynomial in the original set of features, even though the computational cost remains polynomial. This unified kernel approach approach can be applied to a number of machine learning problems, such as supervised classification and regression, semi-supervised learning and unsupervised methods, such as clustering. The classic example of this kernel approach is found in the SVM classifier.

2.1 Kernel Functions

One key aspect of the SVM model is that the data enters both the optimisation problem and the decision function only in the form of the dot product of pairs. This enables SVMs to handle non-linear data. The dot product is replaced by a kernel function, $K(x, z) = \langle \phi(x), \phi(z) \rangle$, that computes the dot product of two samples in a feature space, where $\phi(x)$ represents the mapping to this feature space. The SVM finds the maximum margin separating hyperplane in the feature space defined by this kernel, thus yielding a non-linear decision boundary in the original input space. With the use of kernel functions, it is possible to compute the separating hyperplane in a high dimensional feature space without explicity carrying out the mapping, ϕ, into that feature space [3]. Typical choices for kernels are the Linear, Polynomial, RBF and Sigmoid kernels. Note that using a Linear kernel is equivalent to working in the original input space. Apart from this kernel, all of the above kernels require the setting of one or more parameters, such as σ, the kernel width of the RBF kernel. One alternative to using these standard kernels is to employ a kernel that has been customised for a particular application domain, e.g. the string kernel of Lodhi *et al.* [1].

Whether building complex kernels from simpler kernels, or designing custom kernels, there are conditions that the kernel must satisfy before it can be said to correspond to some feature space. Firstly, the kernel must be symmetric, i.e. $K(x, z) = \langle \phi(x), \phi(z) \rangle = \langle \phi(z), \phi(x) \rangle = K(z, x)$. Typically, kernels are also required to satisfy Mercer's theorem, which states that the matrix $K = (K(x_i, x_j))_{i,j=1}^{n}$ must be positive semi-definite, i.e. it has no negative eigenvalues [4]. In SVM classification, this condition ensures that the solution of the optimisation problem produces a global optimum.

However, good results have been achieved with non-Mercer kernels, and convergence is expected when the SMO algorithm is used, despite no guarantee of optimality when non-Mercer kernels are used [5]. Furthermore, despite its wide use, the Sigmoid kernel matrix is not positive semi-definite for certain values of the parameters γ and θ [6].

3 KTree and SVM Classification

A critical stage in the use of kernel-based algorithms is kernel selection, as this can be shown to correspond to the encoding of prior knowledge about the data [7]. SVM users typically employ one of the standard kernels listed in Section 2.1. Kernels can also be constructed by using simpler kernels as building blocks, e.g. the kernel, $K(x, z) = K_1(x, z)+K_2(x, z)$ or by using the custom kernel approach. Ideally, a kernel is selected based on prior knowledge of the problem domain, but it is not always possible to make the right of choice of kernel *a priori*.

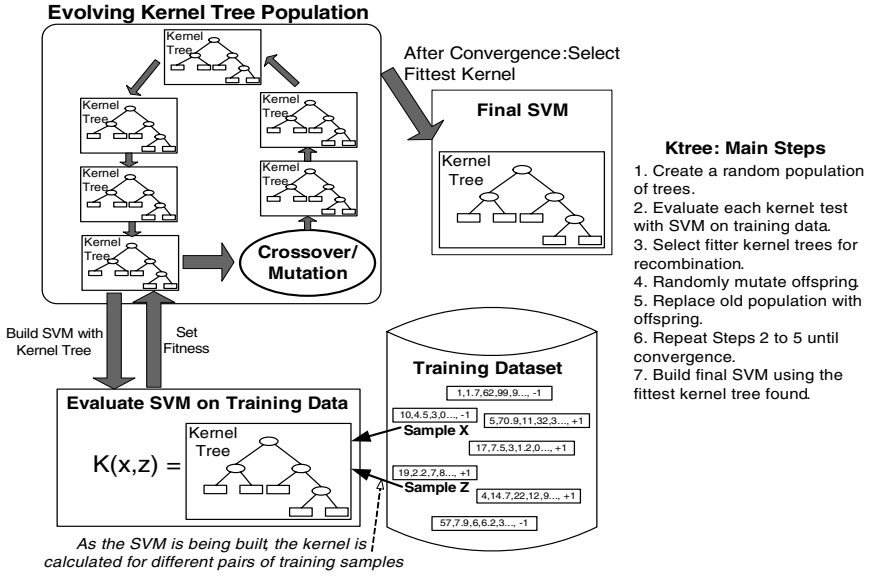


Fig. 1. Application of KTree to the SVM

The approach presented here uses an evolutionary technique to discover a suitable kernel for a particular problem. In this case, KTree is used to evolve kernels specifically for SVM classifiers, but this approach can be used with other kernelised pattern analysis algorithms. The aim of KTree is to eliminate the need for testing various kernels and parameter settings, while also allowing for the discovery of new non-standard kernels. With KTree, a tree structure, known as a *kernel tree* (see Figure 2) is used to represent a kernel function. The objective of KTree is to find a kernel tree that best represents the data. An overview of the application of KTree to the SVM is shown in Figure 1, which also includes the main steps in the building of a SVM using KTree.

3.1 Kernel Tree Representation

The kernel tree used to represent a kernel function must take two data samples as inputs and provide a scalar value as output. An example of a kernel tree is shown in Figure 2.

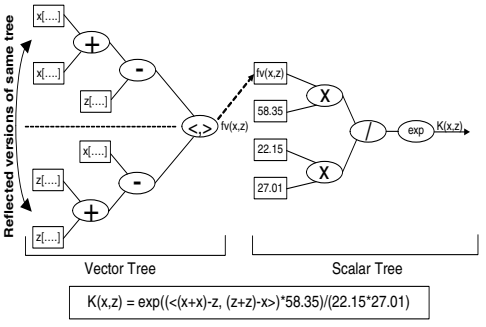


Fig. 2. Kernel Tree generated for Ionosphere Data

This particular kernel tree was generated from experiments on the Ionosphere dataset. The diagram shows that the kernel tree is split into two parts, the vector and the scalar tree. The inputs to the vector tree are the two samples, x and z, for which the kernel is being evaluated. These inputs are passed through vector operators, such as *add* or *subtract*, which in turn pass vectors onto the next node. To ensure that the output of this tree is symmetric, the entire vector tree is evaluated twice, swapping the inputs x and z for the second evaluation. The final output of the vector tree, $f_v(x, z)$, is the dot product of these two evaluations. This output becomes an input, along with randomly generated constant terminals, for the scalar tree. This design was chosen to allow for the use of complex mathematical operators, such as exp and $tanh$, in the scalar tree. Applying these operators directly to the vector inputs could result in overly complex and unusable kernels. A second motivation for this design is that it is also capable of representing the standard kernels, e.g. the RBF kernel and Polynomial kernel. Although symmetry is satisfied, this kernel tree design is not guaranteed to produce Mercer kernels. However, non-Mercer kernels can be filtered out (see Section 3.2).

For the initial population, each kernel tree (both vector and scalar parts) is generated by randomly creating a root node and by growing a tree from this node until either no more leaves can be expanded (i.e. all leaves are terminals) or until a preset initial maximum depth has been reached (2 for the experiments reported here). The evolutionary process shown in Figure 1 involves the application of mutation and crossover operators on selected kernel trees. For mutation, a point in either the vector or scalar tree is randomly chosen and the sub-tree at that point is replaced with a newly generated tree (vector or scalar, depending on where mutation occurred). Mutation of individual nodes (e.g. constant terminals) is not employed. Crossover between two kernel trees begins with the selection of a random point from either the vector or scalar part of the first kernel tree. The location of the crossover point on the second kernel tree is constrained so that crossover does not occur between the scalar part of one kernel tree and the vector

part of another. Rank-based selection was employed for the selection of the candidates for crossover. To prevent the proliferation of massive tree structures, pruning is carried out on kernel trees after mutation, maintaining a maximum depth of 12 (for either the vector or scalar part). A population of 500 kernel trees was used for all experiments, each being evolved over 32 generations, on average.

3.2 Fitness Function

Another key element of KTree is the choice of fitness function. Three different fitness functions were tested in experiments on a synthetic dataset (see Section 4.1). Two of the fitness functions are based on training set classification error in combination with a different tiebreaker fitness (to limit overfitting on the training set). The first tiebreaker fitness is based on kernel tree size, favouring smaller trees, in the spirit of *Ockham's Razor*. The second tiebreaker fitness is based on the sum of the support vector values, $\sum \alpha_i$ (where $\alpha_i = 0$ for non-support vectors). It favours kernels with a smaller sum and also incorporates a penalty corresponding to the radius of the smallest hypersphere, centred at the origin, that encloses the training data in feature space. The third fitness function employed is based on a 3-fold cross-validation test on the training data and also uses tree size as a tiebreaker fitness. In this case, the same kernel is used to build an SVM three times over the course of one fitness evaluation. The experimental analysis of Section 4.1 details the results of applying the above fitness functions on a synthetic dataset.

In addition to the above fitness evaluations, the use of a filter for non-Mercer kernels (referred to as the *Mercer filter*) was investigated. To estimate the Mercer condition of a kernel, the eigenvalues of the kernel matrix over the training data are calculated; if any negative eigenvalues are discovered, the kernel is marked as non-Mercer and is assigned the worst possible fitness, e.g. a cross-validation error of 100%. To reduce the computational cost when dealing with larger datasets, the kernel matrix is based on only a subset of the training data. This approach was to found to be effective in the experiments (detailed in Section 4). The kernel matrix was limited to a maximum size of 250x250.

4 Experimental Results

4.1 Synthetic Dataset

To determine the best strategy for evolving kernels for use in SVM classifiers, a number of experiments were carried out on a synthetic dataset, the checkerboard dataset, shown in Figure 3(a). A checkerboard dataset (similar to that used by Mangasarian *et al.* [8]) of 10,000 samples was generated with an equal distribution of both classes. This synthetic dataset allows for the creation of a large test set that is suitable for comparing different kernel classifiers and is also useful for visually comparing kernel performance. In addition to finding a strategy that generates kernels with good classification accuracy, this research is concerned with issues such as the selection of fitness function, the effect of using non-Mercer kernels, and the contribution of genetic operators.

Table 1. Results on Checkerboard Dataset

(a) Standard Kernels

Standard Kernel	**Fitness**	**Error**
Linear (C=1)	43.3%	48.4%
Poly (C=32,d=13)	19.6%	27.48%
RBF (C=16,σ=8)	14.8%	11.67%
Sigmoid (C=0.1, (γ=10,θ=1E-6)	40.8%	48.92%

(b) KTree

KTree	**Fitness**	**Error**
Default	9.6%	10.74%
Training + No. Nodes	–	14.19%
Training + $\sum \alpha_i$	–	41.26%
No Mercer Filter	8.00%	7.43%
No Crossover	11.6%	7.71%
No Mutation	11.2%	12.57%

Table 1 shows the results on the checkerboard dataset for the standard kernels and KTree. In both cases, the SVM was trained on a subset of 250 samples from the checkerboard dataset and then tested on the full dataset. For each standard kernel, a simple technique is employed for choosing parameters: an SVM with the standard kernel is tested on the training dataset over a range of settings for both kernel and SVM (C parameter). The degree parameter, d, ws tested with the values: $1, 2, \ldots, 19, 20$. C and σ were tested with the values $2^{-20}, 2^{-19}, \ldots, 2^{19}, 2^{20}$, except for the Sigmoid kernel, in which case C and the two other parameters (γ and θ) were tested with the following values: $10^{-6}, 10^{-5}, \ldots, 10^{5}, 10^{6}$. For each kernel type, the kernel setting of the best fitness (based on 3-fold cross-validation error) is chosen and used to build an SVM on the entire training dataset, the resulting model used for the test dataset. Table 1 shows the fitness of the final selected kernel (for both standard and KTree) along with its test error. Table 1(a) shows the RBF kernel outperforming all other standard kernels. The KTree results are based on different variations of KTree, depending on choice of fitness estimate, use of Mercer filter and crossover/mutation rates. The default KTree of Table 1(b) uses a fitness function based on 3-fold cross-validation error, employs a Mercer filter and uses both mutation and crossover. The next two KTree variations use the other two fitness estimates (based on training error with either number of nodes or $\sum \alpha_i$ as tiebreaker) outlined in Section 3.2. The results show that the default setting achieves the best results out of the three, with KTree using training error with α-radius estimate performing very badly. Further analysis of fitness vs. test error showed the fitness based on 3-fold cross-validation to be more stable; this fitness estimate is used as the default in the remaining experiments (both synthetic and UCI datasets).

This study is also concerned with the behaviour of the genetic operators used in KTree. The traditional view of Genetic Algorithms (GAs) is that crossover is primarily responsible for improvements in fitness, and that mutation serves a secondary role of reintroducing individuals that have been lost from the population. However, an important difference with GPs (compared with GAs) is that crossover in GP swaps trees of varying sizes, shape, and position, whereas the typical GA swaps alleles at exactly the same locus [9]. Furthermore, changing a function or a terminal in a GP can have a dramatic effect on the operations of other functions and terminals not only within its own subtree, but throughout an individual. The default setting for KTree shown in Table 1 adopts the classical approach, i.e. a high crossover rate (0.8) relative to the mutation

rate (0.2). The results for KTree with two other different settings are shown in the last two rows of this table: one without crossover and the other without mutation. KTree without crossover achieved a very good test error, but the actual fitness of its best individual is worse than that produced by the default KTree. In terms of final kernel fitness, there is very little difference between KTree based on crossover alone and that based on mutation alone. This is in agreement with Luke & Spector's conclusion that there is often no significant difference between the performance obtained by an all-crossover strategy or an all-mutation strategy. As selecting a very high mutation rate can have adverse effects on convergence and also result in a significant increase in the number of kernel evaluations required in one run, KTree used for tests on the UCI datasets uses a higher crossover rate (0.8) than mutation rate (0.2).[1]

In addition to these results, the output for four different kernels, shown in Figure 3, was used to compare the performance of KTree with that of standard kernels. Two variations of the KTree are shown: the default KTree with Mercer filter and the same KTree, except without a filter for Mercer kernels. It can be seen from these figures that both kernels achieve an output that is much closer to the original checkerboard pattern than the standard kernels' output. A comparison of the fitness versus test error of the kernels produced during the non-Mercer KTree run shows a reasonable trend, but does indicate a greater danger for finding highly fit kernels with poor test performance.

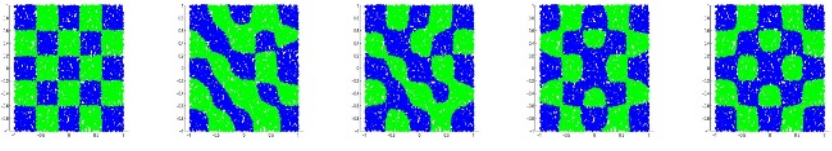


Fig. 3. Output of standard and KTree kernels

4.2 UCI Datasets

The overall conclusion from the experiments on the synthetic dataset is that KTree is capable of producing kernels that outperform the best standard kernels. A further test on nine binary classification datasets from the UCI machine learning repository [10] was carried out. The results of Table 2 show the average error from a single 10-fold cross validation test on each dataset. For each dataset, the lowest average error is highlighted in bold. A pairwise comparison between kernels over all datasets (see Table 3) was carried out using a two-tailed Wilcoxon Signed Rank test [11] at a confidence level of 5%. Table 3 shows that KTree (with Mercer filter) significantly outperforms all other kernels, with the exception of the RBF kernel (no significant difference found). These results show that KTree is capable of outperforming or matching the best results of the most widely used standard kernels. Further tests on the UCI data showed that KTree without Mercer filter yielded poor results (not shown) in comparison with the KTree that incorporates the Mercer filter.

[1] We note that this is still a reasonably high mutation rate.

Table 2. Classifier 10-fold Error Rates(%): see Table 3 for pairwise comparisons of kernels over all datasets

Dataset	Linear	Polynomial	RBF	Sigmoid	KTree
Ionosphere	13.66±3.62	8.30±4.12	5.72±2.68	9.49±5.02	**5.70±2.32**
Heart	**17.78±7.96**	**17.78±7.96**	18.15±8.27	18.52±5.52	**17.78±8.69**
Hepatitis	17.91±9.84	25.33±22.00	18.66±12.18	21.09±14.05	**14.08±8.82**
Sonar	21.72±9.56	16.84±10.61	14.43±8.98	18.25±9.18	**11.58±7.25**
Glass2	29.55±12.89	27.36±13.66	**15.72±13.06**	27.12±13.08	16.31±11.42
Pima	25.91±11.15	23.44±3.44	23.05±3.96	22.66±4.44	**22.53±4.48**
WBCP	26.85±9.20	32.09±17.99	**22.62±5.44**	30.93±9.33	24.2±2.72
Liver	31.54±7.71	30.63±9.31	29.18±8.30	**27.15±8.41**	27.73±8.93
Tic-Tac-Toe	1.67±1.12	0.10±0.33	0.21±0.65	**0.00±0.00**	0.42±0.72

Table 3. Performance on Independent Test Sets: pairwise comparison of kernels using Wilcoxon Test (W=Win, L=Loss, D=Draw–no sig. difference). Overall, KTree exhibits the best results.

Kernel	Lin	Poly	RBF	Sig	KTree
Lin	-	D	W	D	W
Poly	D	-	W	D	W
RBF	L	L	-	D	D
Sig	D	D	D	-	W
KTree	L	L	D	L	-
W/L/D	0/2/2	0/2/2	2/0/2	0/1/3	3/0/1

Table 4. Average kernel fitness (based on 3-fold error) on the Training Sets: this shows that KTree kernels achieve the best fitness

Dataset	Linear	Polynomial	RBF	Sigmoid	KTree
Ionosphere	12.38±0.64	8.00±0.58	4.72±0.47	11.05±3.6	**4.12±0.77**
Heart	16.13±0.77	16.09±0.79	15.56±0.93	15.68±0.94	**13.91±0.84**
Hepatitis	15.63±1.49	14.27±1.79	14.70±1.88	17.06±1.64	**12.76±2.03**
Sonar	22.54±2.32	14.69±2.06	12.82±1.77	19.66±2.70	**8.01±1.70**
Glass2	28.09±2.58	20.31±2.04	15.13±2.47	27.06±2.33	**13.98±2.82**
Pima	22.51±0.45	22.02±0.43	22.05±0.50	22.18±0.51	**21.76±0.52**
WBCP	23.63±0.16	23.23±0.53	**21.44±1.24**	23.12±0.66	22.17±0.99
Liver	30.08±1.61	24.80±1.24	24.83±1.17	25.60±1.16	**23.41±1.17**
Tic-Tac-Toe	1.67±0.12	0.94±0.21	0.51±0.25	0.59±0.28	**0.38±0.19**

All methods compared in Table 2 use the same basic fitness evaluation for selecting the best model for a given training set, namely 3-fold cross-validation error. Therefore, ten different kernels are selected over the course of a 10-fold cross-validation run. Table 4 shows the average fitness (or 3-fold error rate) of the ten models selected for each

kernel type. The best fitness (or lowest error) is highlighted in bold. It was found that KTree significantly outperformed (using the same Wilcoxon test as before) all of the standard kernels in terms of the average fitness of the final kernels selected. This result suggests that with with a better fitness function (i.e. one that follows the actual test error more closely), KTree may be able improve its performance on test data. On the other hand, the datasets used in these tests may be the cause of some of the problems; the presence of noise in these datasets may be adversely affecting the usefulness of this particular fitness estimate. Although the 3-fold error fitness results in good performance, further investigation is required to find a more suitable (and possibly more efficient) fitness measure. For example, it may be possible to use the training error (which is quicker to compute) as a fitness estimate when dealing with larger datasets, where there is less danger of overfitting.

5 Related Research

Some research has been carried out on the use of evolutionary approaches in tandem with SVMs. Frohlich *et al.* use GAs for feature selection and train SVMs on the reduced data [12]. The novelty of this approach is in its use of a fitness function based on the calculation of the theoretical bounds on the generalisation error of the SVM. This approach was found to achieve better results than when a fitness function based on cross-validation error was used. A RBF kernel was used in all reported experiments. Other work has used evolutionary algorithms to optimise a single kernel, typically the RBF Kernel [13,14]. Similarly, Lessmann *et al.* [15] used a GA to optimise a set of parameters for five kernel types and the SVM C parameter, and is also used to determine how the result of each kernel is combined (addition or multiplication) to give the final kernel output. A separate hold-out validation set is used to assess the fitness of each kernel candidate.

6 Conclusions

This paper has described an evolutionary method for constructing the kernel of a kernel-based classifier, in this case the SVM. KTree is a data-driven approach that uses GP to evolve a suitable kernel for a particular problem. Experiments on a synthetic dataset were carried out to determine suitable settings for KTree. Using a fitness function based on an internal cross-validation test was found to yield the best result. In addition, both mutation and crossover operators were found to be useful for the discovery of better kernels. Tests on a number standard datasets show that KTree is capable of matching or beating the best performance of any of the standard kernels tested. When compared using the fitness measure, the kernels produced with KTree clearly outperform the best standard kernels. The results also highlight the need for future work into finding a more effective fitness estimate, with which the performance of KTree could be improved.Future work will also involve testing on more datasets and using the KTree approach for regression problems and cluster analysis.

References

1. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.: Text classification using string kernels. Journal of Machine Learning Research **2** (2002)
2. Howley, T., Madden, M.G.: The Genetic Kernel Support Vector Machine: Description and Evaluation. Artificial Intelligence Review **24** (2005)
3. Scholkopf, B.: Statistical Learning and Kernel Methods. Technical Report MSR-TR-2000-23, Microsoft Research, Microsoft Corporation (2000)
4. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines. Cambridge University Press (2000)
5. Bahlmann, C., Haasdonk, B., Burkhardt, H.: On-line Handwriting Recognition with Support Vector Machines - A Kernel Approach. In: Proc. of the 8th Intl. Workshop on Frontiers in Handwriting Recognition. (2002)
6. Lin, H., Lin, C.: A Study on Sigmoid Kernels for SVM and the Training of non-PSD Kernels by SMO-type Methods. Technical report, Dept. of Computer Science and Information Engineering, National Taiwan University (2003)
7. Cristianini, N., Shawe-Taylor, J.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)
8. Mangasarian, O., Musicant, D.: Lagrangian Support Vector Machines. Journal of Machine Learning Research **1** (2001)
9. Luke, S., Spector, L.: A Comparison of Crossover and Mutation in Genetic Programming. In: Genetic Programming: Proc. of the 2nd Annual Conference, Morgan Kaufmann (1997)
10. Newman, D., Hettich, S., Blake, C., Merz, C.: UCI Repository of machine learning databases (1998)
11. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics **1** (1945) 80–83
12. Frolich, H., Chapelle, O., Scholkopf, B.: Feature Selection for SVMs by Means of Genetic Algorithms. In: Proc. of the Intl. IEEE Conference on Tools with AI. (2003) 142–148
13. Runarsson, T., Sigurdsson, S.: Asynchronous Parallel Evolutionary Model Selection for Support Vector Machines. Neural Information Processing - Letters and Reviews **3** (2004)
14. Friedrichs, F., Igel, C.: Evolutionary Tuning of Multiple SVM Parameters. In: Proc. of the 12th European Symposium on Artificial Neural Network. (2004) 519–524
15. Lessmann, S., Stahlbock, R., Crone, S.: Genetically constructed kernels for support vector machines. In: Proc. of German Operations Research (GOR). (2005)

A Leave-K-Out Cross-Validation Scheme for Unsupervised Kernel Regression

Stefan Klanke and Helge Ritter

Neuroinformatics Group
Faculty of Technology
University of Bielefeld
P.O. Box 10 01 31
33501 Bielefeld, Germany
{sklanke, helge}@techfak.uni-bielefeld.de

Abstract. We show how to employ leave-K-out cross-validation in Unsupervised Kernel Regression, a recent method for learning of nonlinear manifolds. We thereby generalize an already present regularization method, yielding more flexibility without additional computational cost. We demonstrate our method on both toy and real data.

1 Introduction

Unsupervised Kernel Regression (UKR) is a recent approach for the learning of principal manifolds. It has been introduced as an unsupervised counterpart of the Nadaraya-Watson kernel regression estimator in [1]. Probably the most important feature of UKR is the ability to include leave-one-out cross-validation (LOO-CV) at no additional cost. In this work, we show how extending LOO-CV to leave-K-out cross-validation (LKO-CV) gives rise to a more flexible regularization approach, while keeping the computational efficiency.

The paper is organized as follows: In the next section we recall the UKR algorithm and briefly review its already existing regularization approaches. After that, we introduce our generalization to LKO-CV as well as a simple complementary regularizer. Then, we report some results of our experiments and finally we conclude with some remarks on the method and an outlook to further work.

2 The UKR Algorithm

In classical (supervised) kernel regression, the Nadaraya-Watson estimator [2,3]

$$\mathbf{f}(\mathbf{x}) = \sum_i \mathbf{y}_i \frac{K(\mathbf{x} - \mathbf{x}_i)}{\sum_j K(\mathbf{x} - \mathbf{x}_j)} \tag{1}$$

is used to describe a smooth mapping $\mathbf{y} = \mathbf{f}(\mathbf{x})$ that generalizes the relation between available input and output data samples $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_i\}$. Here, $K(\cdot)$ is a density kernel function, e.g. the Gaussian kernel $K(\mathbf{v}) \propto \exp\left[-\frac{1}{2h^2}\|\mathbf{v}\|^2\right]$, where h is a bandwidth parameter which controls the smoothness of the mapping.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 427–436, 2006.

In unsupervised learning, one seeks both a faithful lower dimensional representation (latent variables) $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ of an observed data set $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N)$ and a corresponding functional relationship. UKR addresses this problem by using (1) as the mapping from latent space to data space, whereby the latent variables take the role of the input data and are treated as *parameters* of the regression function. By introducing a vector $\mathbf{b}(\cdot) \in \mathbb{R}^N$ of basis functions, the latter can conveniently be written as

$$\mathbf{f}(\mathbf{x}; \mathbf{X}) = \sum_i \mathbf{y}_i \frac{K(\mathbf{x} - \mathbf{x}_i)}{\sum_j K(\mathbf{x} - \mathbf{x}_j)} = \mathbf{Y}\mathbf{b}(\mathbf{x}; \mathbf{X}) \ . \tag{2}$$

While the bandwidth parameter h is crucial in classical kernel regression, here we can set h=1, because the scaling of $\mathbf{X}$ itself is free. Thus, UKR requires no additional parameters besides the choice of a density kernel[1]. This distinguishes UKR from many other algorithms (e.g. [4,5]) that, albeit using a similar form of regression function, need an a priori specification of many parameters (e.g. the number of basis functions).

Training an UKR manifold, that is, finding optimal latent variables $\mathbf{X}$, involves gradient-based minimization of the reconstruction error (or empirical risk)

$$R(\mathbf{X}) = \frac{1}{N} \sum_i \|\mathbf{y}_i - \mathbf{f}(\mathbf{x}_i; \mathbf{X})\|^2 = \frac{1}{N} \|\mathbf{Y} - \mathbf{Y}\mathbf{B}(\mathbf{X})\|_F^2, \tag{3}$$

where the $N \times N$-matrix of basis functions $\mathbf{B}(\mathbf{X})$ is given by

$$(\mathbf{B}(\mathbf{X}))_{ij} = b_i(\mathbf{x}_j) = \frac{K(\mathbf{x}_i - \mathbf{x}_j)}{\sum_k K(\mathbf{x}_k - \mathbf{x}_j)} \ . \tag{4}$$

To avoid getting stuck in poor local minima, one can incorporate nonlinear spectral embedding methods (e.g. [6,7,8]) to find good initializations.

It is easy to see that without any form of regularization, (3) can be trivially minimized to $R(\mathbf{X}) = 0$ by moving the $\mathbf{x}_i$ infinitely apart from each other. In this case, since $K(\cdot)$ is a density function, $\forall_{i \neq j} \|\mathbf{x}_i - \mathbf{x}_j\| \to \infty$ implies that $K(\mathbf{x}_i - \mathbf{x}_j) \to \delta_{ij} K(\mathbf{0})$ and thus $\mathbf{B}(\mathbf{X})$ becomes the $N \times N$ identity matrix.

2.1 Existing Regularization Approaches

Extension of latent space. A straight-forward way to prevent the aforementioned trivial interpolation solution and to control the complexity of an UKR model is to restrict the latent variables to lie within a certain allowed (finite) domain $\mathcal{X}$, e.g. a sphere of radius R. Training of the UKR model then means solving the optimization problem

$$\text{minimize} \quad R(\mathbf{X}) = \frac{1}{N} \|\mathbf{Y} - \mathbf{Y}\mathbf{B}(\mathbf{X})\|_F^2 \quad \text{subject to} \quad \forall_i \|\mathbf{x}_i\| \leq R \ . \tag{5}$$

[1] which is known to be of relatively small importance in classical kernel regression.

A closely related, but softer and numerically easier method is to add a penalty term to the reconstruction error (3) and minimize $R_e(\mathbf{X}, \lambda) = R(\mathbf{X}) + \lambda S(\mathbf{X})$ with $S(\mathbf{X}) = \sum_i \|\mathbf{x}_i\|^2$. Other penalty terms (e.g. the L_p-norm) are possible.

With the above formalism, the model complexity can be directly controlled by the pre-factor λ or the parameterization of $\mathcal{X}$. However, normally one has no information about how to choose these parameters. Bigger values of λ lead to stronger overlapping of the density kernels and thus to smoother manifolds, but it is not clear how to select λ to achieve a *certain* degree of smoothness.

Density in latent space. The denominator in (1) is proportional to the Rosenblatt-Parzen density estimator $p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} K(\mathbf{x} - \mathbf{x}_i)$. Stronger overlap of the kernel functions coincides with higher densities in latent space, which gives rise to another method for complexity control. As in the last paragraph, the density $p(\mathbf{x})$ can be used both in a constraint minimization of $R(\mathbf{X})$ subject to $\forall_i p(\mathbf{x}_i) \geq \eta$ or in form of a penalty function with some pre-factor λ. Compared to a regularization based on the extension of latent space, the density based regularization tends to work more locally and allows a clustered structure of the latent variables (non-contiguous manifolds). Again, suitable values for λ and η can be difficult to specify.

Leave-one-out cross-validation. Perhaps the strongest feature of UKR is the ability to include leave-one-out cross-validation (LOO-CV) without additional computational cost. Instead of minimizing the reconstruction error of a UKR model including the complete dataset, in LOO-CV each data vector $\mathbf{y}_i$ has to be reconstructed without using $\mathbf{y}_i$ itself:

$$R_{cv}(\mathbf{X}) = \frac{1}{N} \sum_i \|\mathbf{y}_i - \mathbf{f}_{-i}(\mathbf{x}_i; \mathbf{X})\|^2 = \frac{1}{N} \|\mathbf{Y} - \mathbf{Y}\mathbf{B}_{cv}(\mathbf{X})\|_F^2 \tag{6}$$

$$\mathbf{f}_{-i}(\mathbf{x}) = \sum_{m \neq i} \mathbf{y}_m \frac{K(\mathbf{x} - \mathbf{x}_m)}{\sum_{j \neq i} K(\mathbf{x} - \mathbf{x}_j)} \tag{7}$$

For the computation of the matrix of basis functions $\mathbf{B}_{cv}$, this just means zeroing the diagonal elements before normalizing the column sums to 1. A similar strategy works also for calculating the gradient of (6).

As long as the dataset is not totally degenerated (e.g. each $\mathbf{y}_i$ exists at least twice), LOO-CV can be used as a built-in *automatic* complexity control. However, under certain circumstances LOO-CV can severely undersmooth the manifold, particularly in the case of densely sampled noisy data. See Fig. 1 (first plot, K=1) for a UKR curve fitted to a sample of a noisy spiral distribution as a result of minimizing the LOO-CV-error (6).

Regularization by special loss functions. Recently, we showed in [9] how to regularize UKR manifolds by incorporating general loss functions instead of the squared Euclidean error in (3). In particular, the ϵ-insensitive loss is favorable if one has information about the level of noise present in the data.

3 UKR with Leave-K-Out Cross-Validation

Generally, leave-K-out cross-validation consists of forming several subsets from a dataset, each missing a different set of K patterns. These K patterns are used to validate a model that is trained with the corresponding subset. The resulting models are then combined (e.g. averaged) to create a model for the complete dataset. The special case $K = 1$ is identical to LOO-CV.

Since UKR comes with LOO-CV "for free", it is interesting to investigate if the concept is applicable for $K > 1$. Hereto, we first have to specify how to form the subsets. With the aim to both maximize and equally distribute the effect of omitting each K data vectors on how UKR fits the manifold, we opt to reconstruct each data vector without itself and its $K - 1$ nearest neighbors. Concerning this, please recall that the UKR function (2) computes a *locally weighted* average of the dataset. Therefore, normally, each data vector is mainly reconstructed from its neighbors. By omitting the immediate neighbors we shift the weight to data vectors farther away, which forces the kernel centers $\mathbf{x}_i$ to huddle closer together and thus leads to a smoother regression function.

Please note that in contrast to standard LKO-CV, this procedure yields N different subsets of size $N - K$, each being responsible for the reconstruction of *one* data vector. A corresponding objective function, which automatically combines the subset models, can be stated as

$$R_{lko}(\mathbf{X}) = \frac{1}{N} \sum_i \|\mathbf{y}_i - \mathbf{f}_i(\mathbf{x}_i; \mathbf{X})\|^2 = \frac{1}{N} \|\mathbf{Y} - \mathbf{Y}\mathbf{B}_{lko}(\mathbf{X})\|_F^2 \tag{8}$$

$$\mathbf{f}_i(\mathbf{x}) = \sum_{m \notin \mathcal{N}_i} \mathbf{y}_m \frac{K(\mathbf{x} - \mathbf{x}_m)}{\sum_{j \neq i} K(\mathbf{x} - \mathbf{x}_j)}, \tag{9}$$

where $\mathcal{N}_i$ describes the index set of neighbors excluded for reconstructing $\mathbf{y}_i$.

In principle, we may consider neighborhoods both in latent space and data space, since a good mapping will preserve the topology anyway. However, it is much simpler to regard only the original neighborhood relationships in data space, because these are *fixed*. The latent space neighborhoods may change with every training step, and thus have to be recomputed. Furthermore, convergence is not guaranteed anymore, because the latent variables $\mathbf{X}$ might jump between two "optimal" states belonging to different neighborhood structures.

As with LOO-CV, data space neighborhood LKO-CV can be implemented in UKR with nearly no additional cost. All one has to do is zeroing certain components of the matrix $\mathbf{B}_{lko}$ before normalizing its column sums to 1. In particular, set $b_{ij} = 0$, if $i \in \mathcal{N}_j$, with fixed and precomputed index sets $\mathcal{N}_j$.

One might argue that the whole idea seems somehow strange, especially if the UKR model is initialized by a spectral embedding method (e.g. LLE) which takes into account some K' nearest neighbors for constructing the lower dimensional representation. Thus, in a way, UKR with LKO-CV works against its initialization method. On the other hand, this can be viewed as being complementary. Furthermore, our experiments not only show that the idea is sound, but even indicate that selecting $K = K'$ is not a bad choice at all.

3.1 How to Get Smooth Borders

As we will show in the next section, LKO-CV does work well at the interior of a manifold, but not at its borders. This results naturally from the topology: At the borders of a 1D manifold (that is, at the ends of a curve) for example, all K neighbors lie in the same direction. Thus, the nearest data points taking part in reconstructing the end points are exceptionally far away. If, after training, the curve is sampled by evaluating the normal UKR function (2), the ends get very wiggly, especially for larger K.

To overcome this problem, we propose to employ an additional regularizer that smoothes at the borders without disturbing LKO-CV in regions that are already fine. Hereto, penalizing the extension of latent space (e.g. by using a penalty term of the form $S(\mathbf{X}) = \|\mathbf{X}\|_F^2$) is a bad choice, since this would affect the manifold as a whole and not only the borders. The same argument applies to a penalty term of the form $S(\mathbf{X}) = -\sum_i \log p(\mathbf{x}_i)$, which favors high densities and thus again smoothes the complete manifold. A possible choice, however, is to penalize the *variance* of the density in latent space. For this, we apply the following penalty term:

$$S(\mathbf{X}) = \frac{1}{N} \sum_i (p(\mathbf{x}_i) - \bar{p}(\mathbf{X}))^2 \quad , \quad \bar{p}(\mathbf{X}) = \frac{1}{N} \sum_j p(\mathbf{x}_j). \tag{10}$$

The UKR model is thus regularized by two factors: 1) the LKO parameter K determines the overall smoothness and 2) the penalty term $S(\mathbf{X})$, scaled by an appropiate pre-factor λ, ensures that the smoothness is evenly distributed.

Because these regularizers have more or less independent goals, one may hope that the results show considerable robustness towards the choice of λ. Indeed, for a UKR model of a noisy spiral (Fig. 2), there was no visual difference between results for $\lambda = 0.001$ and $\lambda = 0.0001$. Only a much smaller value ($\lambda = 10^{-6}$), led to wiggly ends, again.

4 Experiments

In all following experiments, we trained the UKR manifolds in a common way: For initialization, we calculated multiple LLE [6] solutions corresponding to different neighborhood sizes K', which we compared with respect to their LKO-CV error (8) after a coarse optimization of their overall scale. While this procedure may seem rather computationally expensive, it greatly enhances the robustness, because LLE and other nonlinear spectral embedding methods can depend critically on the choice of K'. In our experiments, the best LLE neighborhood size K' did not depend on which LKO neighborhood size K we used. Further fine-tuning was done by gradient-based minimization, applying 500 RPROP [10] steps. For simplicity, we used only the Gaussian kernel in latent space.

4.1 Noisy Spiral

As a first example, we fitted a UKR model to a 2D "noisy spiral" toy dataset, which contains 300 samples with noise distributed uniformly in the interval

$[-0.1; 0.1]$. We tried LLE neighborhood sizes $K' = 4 \ldots 12$, of which $K' = 7$ led to the best initialization. Fig. 1 shows the results for different values of the LKO-CV parameter K as indicated in the plots. Note how the manifold gets smoother for larger K, without suffering from too much bias towards the inner of the spiral. A bit problematic are the manifolds ends, which get quite wiggly for larger K. Note that $K = K' = 7$ yields a satisfactory level of smoothness.

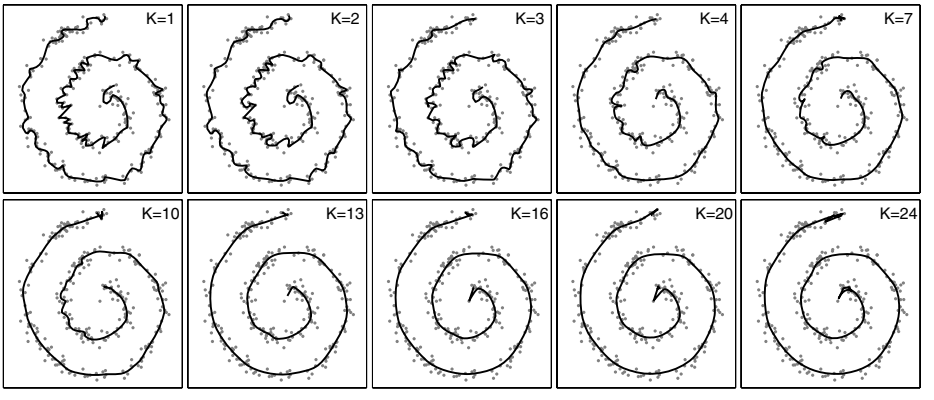


Fig. 1. UKR model of a noisy spiral using LKO-CV. The data points are depicted as grey dots, the black curve shows the manifold which results from sampling $\mathbf{f}(\mathbf{x}; \mathbf{X})$.

To show the effect of the density variance penalty term (10), we repeated the experiment adding the penalty with pre-factors $\lambda = 10^{-3}, 10^{-4}$ and 10^{-6}. Fig. 2 shows the results for $\lambda = 10^{-4}$, which are visually identical to those for $\lambda = 10^{-3}$. However, a pre-factor of only 10^{-6} turned out to be too small, resulting in wiggly ended curves similar to those in Fig. 1.

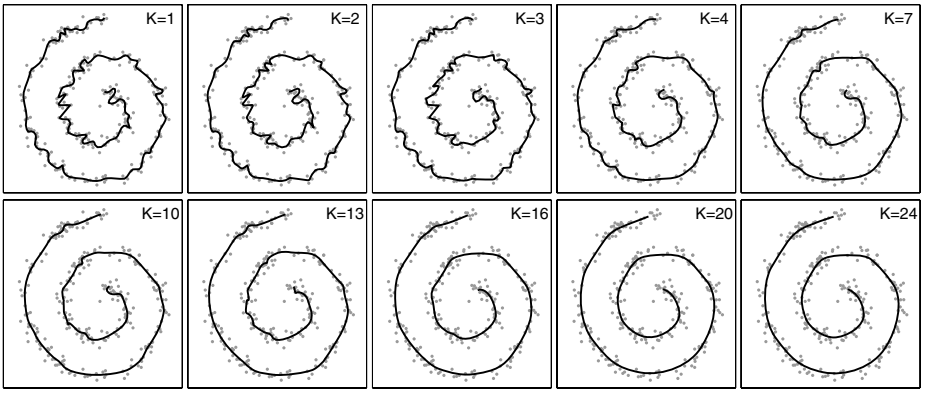


Fig. 2. UKR model of a noisy spiral using both LKO-CV and the density variance penalty term (10) scaled by a pre-factor $\lambda = 10^{-4}$

Some insight on the influence of the density variance penalty is provided by Fig. 3: Most of the latent variables stay in the same region, but the outliers (the little bumps to the far left and right) are drawn towards the center, compacting the occupied latent space. Figure 4 shows a magnified comparison of the UKR models ($K = 24$) with and without the penalty term. In addition to the original data points and the resulting curve, it also depicts the data as it is reconstructed during training, that is, using the LKO function (9). Note that these LKO reconstructions show a strong bias towards the inner of the spiral, which is not present in the final mapping (2) based on the complete data set.

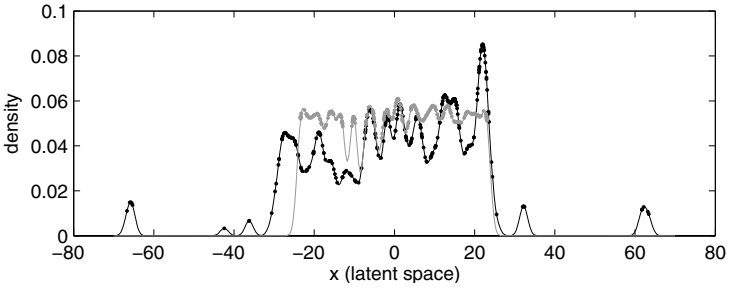


Fig. 3. Comparison of latent densities for UKR models of a noisy spiral using a) only LKO-CV ($K = 24$, depicted in black) and b) LKO-CV together with the densisty variance penalty ($K = 24$, $\lambda = 10^{-4}$, depicted in gray). The curves result from sampling $p(x)$, the dots indicate the latent variable positions.

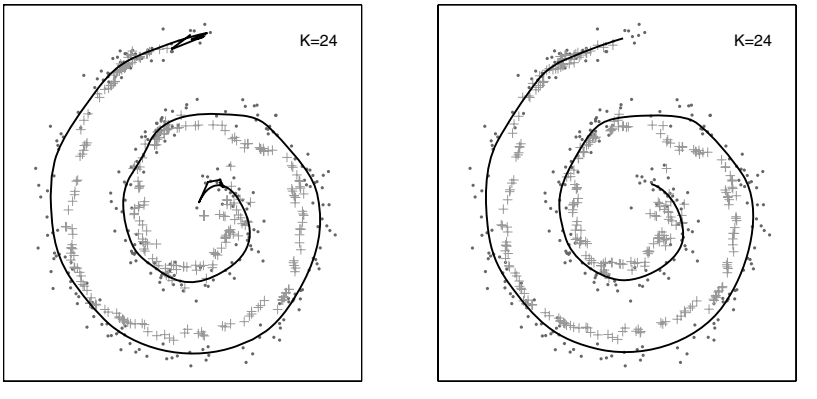


Fig. 4. Comparison of UKR models of a noisy spiral. Left: pure LKO-CV ($K = 24$). Right: with additional density variance penalty ($\lambda = 10^{-4}$). The dots depict the observed data points, the black curve depicts the manifold, and the gray pluses depict the LKO reconstructions (9).

4.2 Noisy Swiss Roll

As a second experiment, we fitted a UKR model to a noisy "Swiss Roll" dataset. We first computed LLE solutions with $K' = 3 \ldots 18$, of which $K' = 7$ was selected as the best initialization for all UKR models. Figure 5 shows the dataset as reconstructed with LOO-CV ($K = 1$) and LKO-CV ($K = 7$). Instead of comparing the results for multiple K's visually again, we projected the reconstructed datasets onto the underlying data model (i.e. the smooth continuous "Swiss Roll"). Figure 6 shows the resulting mean distance as a function of K. The minimum is at $K = 9$, with our proposed automatic choice $K = K' = 7$ being nearly as good.

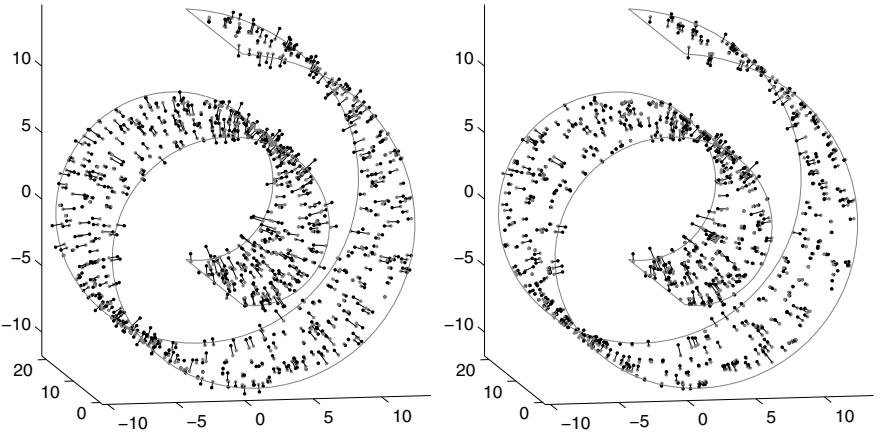


Fig. 5. UKR reconstruction of a "Swiss Roll". Left: $K = 1$ (LOO-CV). Right: $K = 7$. The black dots depict the UKR reconstructions, whereas the gray dots depict their projection (along the black lines) onto the underlying smooth data model. Note the much smaller projection error (distance to the "true" manifold) in the right plot.

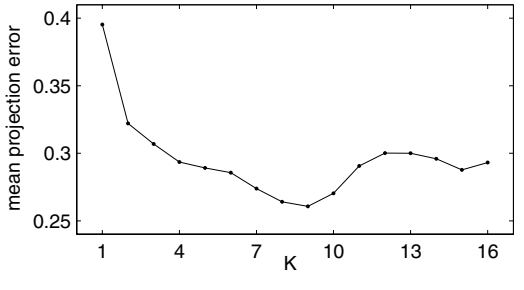


Fig. 6. Mean distance between LKO-CV-UKR reconstructions and their projections onto the underlying smooth data manifold. The corresponding projection error of the observed (noisy) data points is 0.498. Please note that the y-axis does not start at 0.

4.3 USPS Digits

To show that LKO-CV-UKR also works with higher dimensional data, our last experiment deals with the USPS handwritten digits. In particular, we work with the subset corresponding to the digit "2", which contains 731 data vectors in 256 dimensions (16x16 pixel gray-scale images). As with the "Swiss Roll", we compared the results of LOO-CV and LKO-CV with $K = K' = 12$, that is, we chose the LKO parameter to be identical to the automatically selected LLE neighborhood size. Both models use the density variance penalty with a pre-factor[2] $\lambda = 0.01$. Figure 7 visualizes the resulting manifolds (we chose a 2D embedding) by sampling $\mathbf{f}(\mathbf{x}; \mathbf{X})$ in latent space and depicting the function value as the corresponding image. Note the smaller extension in latent space and the blurrier images of the model belonging to $K = 12$ (right plot).

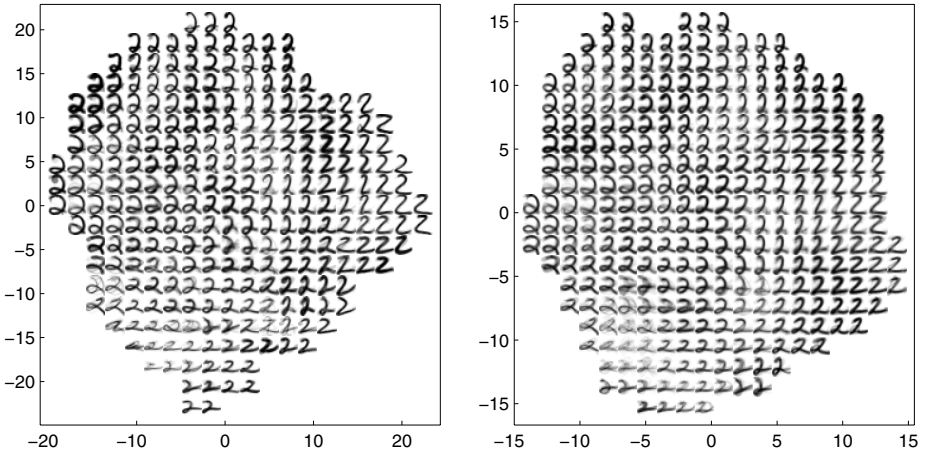


Fig. 7. UKR model of the USPS digit "2", shown by evaluating $\mathbf{f}(\mathbf{x}; \mathbf{X})$ on a 20x20 grid enclosing the latent variables. Grid positions of low density $p(\mathbf{x})$ are left blank. Left: $K = 1$ (LOO-CV). Right: $K = 12$.

5 Conclusion

In this work, we described how leave-K-out cross-validation (LKO-CV) can be employed in the manifold learning method UKR, generalizing the already present LOO-CV regularization. We demonstrated our approach on both synthetic and real data. When used with pre-calculated data space neighborhoods, LKO-CV involves nearly no additional computational cost, but can yield favorable results. This was revealed especially in the noisy "Swiss Roll" experiment, where LKO-CV significantly reduced the projection error, i.e. the mean distance between the reconstructed (de-noised) dataset and the "true" underlying manifold.

[2] Here, we used a larger λ because the data's variance is larger, too.

While we gave no final answer to the question how to choose the new regularization parameter K, our experiments indicate that simply setting $K = K'$ (the neighborhood size of the best LLE solution, which UKR can automatically detect) yields satisfactory results. In addition, we showed how a complementary regularizer, which is based on penalizing a high variance of the latent density, can further enhance the UKR models trained with LKO-CV. By promoting an even distribution of smoothness, this regularizer diminishes the problem of rather wiggly manifold borders, which otherwise may result from a pure LKO-CV regularization. When used as a penalty term, the complementary regularizer is quite robust towards the choice of an appropiate pre-factor.

Further work may address other possibilities to deal with the border problem, e.g. by a smart local adaption of the neighborhood parameter K. We also successfully experimented with leave-R-out CV, a scheme where not a fixed number of neighbors are left out, but all neighbors within a sphere of fixed size. Finally, it will be interesting to see how UKR with LKO-CV performs in real applications.

References

1. Meinicke, P., Klanke, S., Memisevic, R., Ritter, H.: Principal surfaces from Unsupervised Kernel Regression. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(9) (2005) 1379–1391
2. Nadaraya, E.A.: On estimating regression. Theory of Probability and Its Application **10** (1964) 186–190
3. Watson, G.S.: Smooth regression analysis. Sankhya Series A **26** (1964) 359–372
4. Bishop, C.M., Svensen, M., Williams, C.K.I.: GTM: The Generative Topographic Mapping. Neural Computation **10**(1) (1998) 215–234
5. Smola, A.J., Williamson, R.C., Mika, S., Schölkopf, B.: Regularized Principal Manifolds. Lecture Notes in Computer Science **1572** (1999) 214–229
6. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by Locally Linear Embedding. Science **290** (2000) 2323–2326
7. Belkin, M., Niyogi, P.: Laplacian Eigenmaps for dimensionality reduction and data representation. Neural Computation **15 (6)** (2003) 1373–1396
8. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290** (2000) 2319–2323
9. Klanke, S., Ritter, H.: Variants of Unsupervised Kernel Regression: General loss functions. In: Proc. European Symposium on Artificial Neural Networks. (2006) to appear.
10. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In: Proc. of the IEEE Intl. Conf. on Neural Networks. (1993) 586–591

Neural Network Clustering Based on Distances Between Objects

Leonid B. Litinskii and Dmitry E. Romanov

Institute of Optical-Neural Technologies Russian Academy of Sciences, Moscow
litin@iont.ru, demaroman@yandex.ru

Abstract. We present an algorithm of clustering of many-dimensional objects, where only the distances between objects are used. Centers of classes are found with the aid of neuron-like procedure with lateral inhibition. The result of clustering does not depend on starting conditions. Our algorithm makes it possible to give an idea about classes that really exist in the empirical data. The results of computer simulations are presented.

1 Introduction

Data clustering deals with the problem of classifying a set of N objects into groups so that objects within the same group are more similar than objects belonging to different groups. Each object is identified by a number m of measurable features, consequently, ith object can be represented as a point $\mathbf{x}_i \in \mathbf{R}^m$, $i = 1, 2, ..., N$. Data clustering aims at identifying clusters as more densely populated regions in the space $\mathbf{R}^m$.

This is a traditional problem of unsupervised pattern recognition. During last 40 years a lot of approaches to solve this problem were suggested. The general strategy is as follows: at first, somehow or other one finds the optimal partition of the points into K classes, and then changes the value of the parameter K from N to 1. Here the main interest is the way how small classes (relating to big values of K) are combined into bigger classes (relating to small values of K). These transformations allow us to get some idea about the structure of empirical data. They indicate mutual location of compact groups of points in many-dimensional space. They also indicate which of these groups are close and which are far from each other. Interpretation of the obtained classes in substantial terms, and it is no less important, the details of their mutual location allows the researcher to construct meaningful models of the phenomenon under consideration.

Different methods of data clustering differ from each other by the way of finding of the optimal partition of the points into K classes. It is literally to say that almost all of them own the same poor feature: the result of partition into K classes depends on arbitrary chosen initial conditions, which have to be specify to start the partition procedure. Consequently, to obtain the optimal partition, it is necessary to repeat the procedure many times, each time starting from new initial conditions. In general, it cannot be guaranteed that the optimal partition into K classes would be found. Here the situation is close to the one, which we face when founding the global minimum of multiextremal functional. The problems of such a kind exhibit a tendency to become

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 437 – 443, 2006.

NP-complete. This means that for large N only a local optimal partition can be found, but not necessary the best one.

Thus, almost all clustering methods based on the local partition of objects into K classes. Among them there are the well-known and most simple *K-means approach* [1]-[3], mathematically advanced *Super-Paramagnetic Clustering* [4] and *Maximum Likelihood Clustering* [5], popular in Russia *the FOREL-type* algorithms [3] and prevailing in the West different variants of *Hierarchical Clustering* [6]. Let us show the problem, using two last approaches as examples.

The general scheme of the *FOREL*-algorithm is as follows: 1) we specify a value T that is the radius of m-dimensional sphere, which in what follows is used as a threshold for interaction radius between points; 2) we place the center of the sphere with the radius T at an arbitrary input point; 3) we find coordinates of the center of gravity of points that find themselves inside the sphere; 4) we transfer the center of the sphere in the center of gravity and go back to item 3; 5) as far as when going from one to the next iterating the sphere remains in the same place, we suppose that the points inside it constitute a class; we move them away from the set and go back to the item 2.

It is clear that after finite number of steps we obtain a partition of the points into some classes. In each class the distances between points are less than $2T$. However, the result of partition depends on the starting point, where the center of the sphere is situated (see item 2). Since the step 2 is repeated again and again, it is evident that the number of different partitions (for fixed T) can be sufficiently large.

The Hierarchical Clustering is based on a very simple idea too. Given some partition into K classes, it merges the two closest classes into a single one. So, starting from the partition into $K = N$ classes, the algorithm generates a sequence of partitions as K varies from N to 1. The sequence of partitions and their hierarchy can be represented by a dendrogram. Applications of this approach are discussed for example in [6]. However, there is no explanation why just two closest classes have to be combined. After all, this is only one of possible reasonable recipes. This leads to local optimal partition too.

Of course, there is no reason to dramatize the situation with regard to local optimality of partitions. During 40 years of practice, approaches and methods were developed allowing one to come to correct conclusions basing on local optimal partitions of objects into classes. However, it is very attractive to construct a method, which would not depend on an arbitrary choice of initial conditions. Just such an algorithm is presented in this publication.

The same as the *FOREL*-algorithm our method is based on introduction of an effective interaction radius T between points $\mathbf{x}_i$ and the partition of the points between spheres of radius T. The points that get into a sphere belong to one class. The centers of the spheres are obtained as a result of a neuron-like procedure with lateral inhibition. As a result, the centers of the spheres are the input points, which for a given radius T interact with maximal number of surrounding points. It can be said that the centers of the spheres are located inside the regions of concentration of the input points. At the same time, we determine the number of classes K that are characteristic for the input data for a given interaction radius T. Then, the value T changes from zero to a very large value. We plot the graph $K(T)$, which shows the dependence of the number of classes on T. We can estimate the number of real classes that are in the

empirical data by the number of lengthy "plateau" on this graph. The calculation complexity of the algorithm is estimated as $O(N^2)$.

In the present publication we describe the clustering algorithm and the results of it testing with the aid of model problems and empirical data known as "Fisher's irises" [2].

2 Clustering Algorithm

For a given set of *m*-dimensional points $\{\mathbf{x}_i\}_1^N \in \mathbf{R^m}$ we calculate a quadratic $(N \times N)$-matrix of Euclidean distances between them: $\mathbf{D} = (D_{ij})_{i,j=1}^N$. In what follows we need these distances D_{ij} only. We suppose that in each point $\mathbf{x}_i$ there is a neuron with initial activity $S_i(0)$, which will be defined below.

1) For a fixed interaction threshold $T > 0$ let us set the value of a connection w_{ij} between *i*th and *j*th neurons as

$$w_{ij} = \begin{cases} \dfrac{T^2}{D_{ij}^2 + T^2} & \text{, when } \dfrac{T^2}{D_{ij}^2 + T^2} \geq 0.5, \\ 0 & \text{, when } \dfrac{T^2}{D_{ij}^2 + T^2} < 0.5. \end{cases}$$

As we see, there are no connections between neurons, if the distance between points is greater than *T*. Note, $w_{ii}(T) \equiv 1$.

2) Let us set initial activity of each neuron to be

$$S_i(0) = \sum_{j=1}^{N} w_{ij} \geq 1$$

Neurons, which are inside agglomerations of the points, have large initial activity, because they have more nonzero connections than neurons at the periphery of agglomerations.

3) We start the activities "transmitting" process:

$$S_i(t+1) = S_i(t) + \alpha \sum_{j=1}^{N} w_{ij}(S_i(t) - S_j(t))$$

where α is the parameter characterizing the transmitting speed. It is easy to see that during the transmitting process a neuron with large initial activity "takes away" activities from neurons with whom it interacts and whose activities are less. The activities of these surrounding neurons decrease steadily.

4) If during the transmitting process the activity of a neuron becomes negative $S_i(t) < 0$, we set $S_i \equiv 0$, and eliminate this neuron from the transmitting process (it has nothing to give away).

It is clear that little by little the neurons from the periphery of agglomerations shall drop out giving away their activities to neurons inside the agglomerations. This means that step by step the neurons from the periphery will leave the field. Gradually, we

shall have a situation, when only some far from each other non-interacting neurons with nonzero activities remain. Subsequent transmitting is impossible and the procedure stops.

5) Suppose as a result of the transmitting process K neurons remain far away from each other. The input points $\mathbf{x}_i$ corresponding to these neurons will be called the centers of the classes. All other input points $\mathbf{x}_j$ are distributed between classes basing on the criterion of maximal closeness to one or another center.

The items 1)-5) are carried out for a fixed interaction threshold T. It is clear that if $T \approx 0$, no one neuron interacts with another one ($w_{ij} = 0$ when $i \neq j$). All the neurons have the same activities $S_i(0) = w_{ii} = 1$. No transmitting process will have place. So we get a great number N of classes, each of which consists of one input point only. On the other hand, if the interaction threshold T is very large (for example, it is greater than $\max(D_{ij})/2$) all neurons are interacting with each other, and as the result of transmitting only one neuron remains active. We can say that it is located inside "the cloud" of the input points. In this limiting case there is only one class including all the input points.

Changing T from zero to it maximal value, we plot the dependence of the number of classes on the value of the interaction threshold, $K(T)$. It was found that for these graphs the presence of long, extensive "plateaus" is typical. In other words, the number of classes K does not change for some intervals, where T changes. These "plateaus" allows one to estimate the number of classes existing really in empirical data (for the first time this criterion was proposed by the author of [3]).

3 The Results of Computer Simulations

The simplest case is shown in Fig.1. On the left panel we see 50 points distributed on the plane into five clusters, on the right panel the obtained graph *K(T)* is shown. We see that at the initial stage of changing of T the number of classes changes very rapidly. Then it becomes stabilize at the level $K = 5$ and does not change in the some

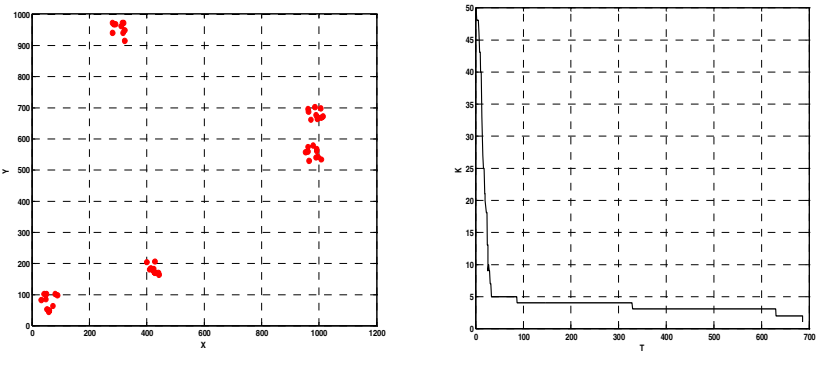


Fig. 1. On the left panel there are 5 compact clusters formed by 50 points at the plane (X,Y); on the right panel the graph $K(T)$ is presented

interval of changing of *T*. Here empirical points are distributed into 5 input clusters exactly. The next plateau corresponds to $K = 4$. In this case the points that belong to two close internal agglomerations combine into one class. The next plateau is when $K = 3$; here the points belonging to two pairs of close to each other agglomerations are combined into two classes. Finally, the last plateau is at the level $K = 2$; here all the points from the lower part of the figure are combined into one class.

In the case of 10 classes (see Fig.2) on the graph small plateaus can be observed when $K = 10, 9, 8, 7, 5, 4$. However, the widest plateau we observe when $K = 2$. This plateau corresponds to the partition of the points between two classes that are sufficiently clearly seen in the left panel of the figure.

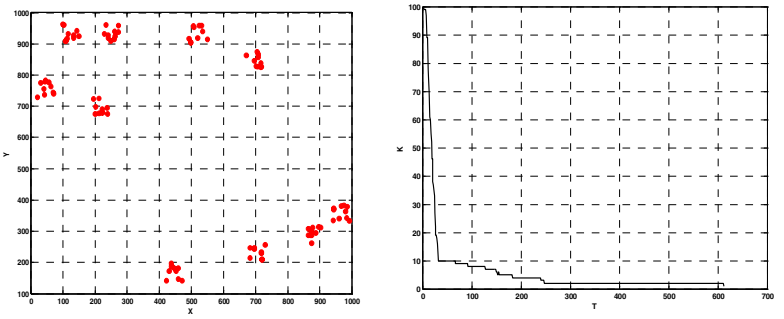


Fig. 2. The same as in the previous figure for 100 points and 10 compact clusters

The last example is the classical «Fisher's irises» clustering problem. The input data is a set of 150 four-dimensional points (four parameters characterize each iris). It is known that these 150 points are distributed between 3 compact classes. Two of them are slightly closer to each other than the third. Our algorithm showed just the same picture of irises distribution between classes (Fig.3). The first plateau is at the level K=3, and the next plateau corresponds to K=2.

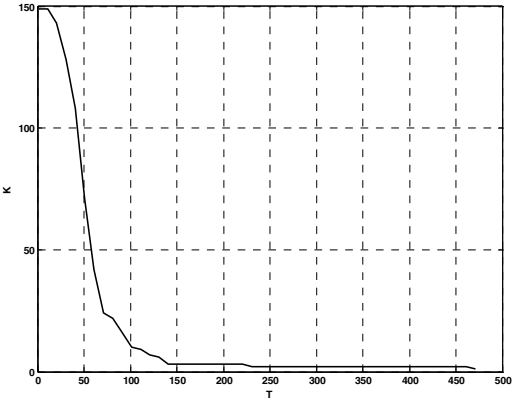


Fig. 3. The graph for Fisher's irises

These examples demonstrate that of interest is not only the separation of the objects into a certain number of classes, but also the way in which small classes join into larger ones. These transformations indicate which of small compact classes of objects are close to each other and allow one to understand the intrinsic structure of empirical data.

4 Discussion

In real empirical data in addition to compact agglomerations certainly there are "noisy" points, which can be found in the space between the agglomerations. In this case our algorithm gives plateaus when the threshold T is small. The number of classes K corresponding to these values of T is sufficiently large: $K>>1$. These classes are mostly fictitious ones. Each of them consists from one noisy point only.

The reason is clear: when the value of the threshold T is small each noisy point has a chance to make its own class consisting from this one point only. For beginning values of T some noisy points take their occasion. That is why there are short, but distinct plateaus corresponding to the large number of classes K. It is easy to ignore these fictitious classes: it is enough to take into account only classes where there are a lot of points. Thus, our algorithm is able to process noisy data sets.

The algorithm is based on the hypothesis of a compact grouping points in the classes and is oriented on the unsupervised pattern recognition. In a general case it is unable to process overlapping sets of data belonging to different classes. This is a problem of supervised pattern recognition. It is possible, however, that different classes overlap slightly, only by their periphery. Then our algorithm can be modified, so that it can separate cores of the classes. Such an approach can be of considerable use, and we plan to examine it in details.

Our algorithm has a usefull property: it works not only for points $\{\mathbf{x}_i\}_1^N$ in the coordinate presentation, but also when we know the distances between points only. These two types of input data, the coordinate presentation of the points, on the one hand, and the distances between the points, on the other, are not equivalent. Indeed, if only the distances between the points are known, it is not so simple to reconstruct the coordinates of the points themselves. As a rule, this can be done only under some additional assumptions, and these assumptions have an influence on the solution of the problem.

Still more general is the clustering problem, when the input data is an arbitrary symmetrical matrix (not necessary the matrix of distances between the points). In this case matrix elements can be negative. As a rule the diagonal elements of such a matrix are equal to zero. Then (ij)th matrix element is treated as a measure of connection between jth and ith objects. The clustering of this matrix aims at finding of strongly connected groups of objects. Of course, matrix elements can be normalized in such a way that they can be treated as "scalar products" of vectors. However, the problem is that with the aid of these "scalar products" it is impossible to reconstruct uniquely even the distances between the vectors. Consequently, to solve the most general clustering problems our algorithm has to be modified. We plan to do this in the following publication.

The work was supported by Russian Basic Research Foundation (grants 05-07-90049 and 06-01-00109).

References

[1] J.T. Tou, R.C.Gonzales. Pattern recognition principles. Addison-Wesley, London-Amsterdam, 1974.
[2] S.Ayvazyan, V.Bukhstaber, I.Enyukov, L.Meshalkin. Applied statistics. Clustering and Reduction of Dimension. Moscow: Finansy i Satitistika, 1989 (in Russian).
[3] N.Zagoruyko. Applied Methods of Data and Knowledge Analysis. Novosibirsk: Institute of Mathematics Press, 1999 (in Russian).
[4] M.Blatt, S.Wiseman, E.Domany. Super-paramagnetic Clustering of Data. Phys. Rev. Letters (1996) v.76, pp.3251-3255.
[5] L.Giada, M.Marsili. Algorithms of maximal likelihood data clustering with applications. Cond-mat/0204202.
[6] R.N. Mantegna. Hierarchical structure in financial markets. European Phys. Journal B (1999) v. 11, pp.193-197.

Rotation-Invariant Pattern Recognition: A Procedure Slightly Inspired on Olfactory System and Based on Kohonen Network

M.B. Palermo[1] and L.H.A. Monteiro[1,2]

[1] Universidade Presbiteriana Mackenzie, Pós-graduação em Engenharia Elétrica, Escola de Engenharia, Rua da Consolação, n.896, 01302-907, São Paulo, SP, Brazil
[2] Universidade de São Paulo, Departamento de Engenharia de Telecomunicações e Controle, Escola Politécnica, Av Prof. Luciano Gualberto, travessa 3, n.380, 05508-900, São Paulo, SP, Brazil
luizm@mackenzie.br, luizm@usp.br

Abstract. A computational scheme for rotation-invariant pattern recognition based on Kohonen neural network is developed. This scheme is slightly inspired on the vertebrate olfactory system, and its goal is to recognize spatiotemporal patterns produced in a two-dimensional cellular automaton that would represent the olfactory bulb activity when submitted to odor stimuli. The recognition occurs through a multi-layer Kohonen network that would represent the olfactory cortex. The recognition is invariant to rotations of the patterns, even when a noise lower than 1% is added.

1 Introduction

Olfactory systems are responsible for discriminating, memorizing and recognizing odors and are crucial for the survival of animals, because foods, friends, sexual partners, predators and territories can be identified by characteristic smells[9,13]. There are models of olfactory systems based on differential equations[8], coupled maps[3] and cellular automata[6]. Usually, the main goal of these models is to reproduce the neural responses occurring after an odor stimulus[9,13]. In vertebrates, the processing of odor information is a multi-level task and, at each level, a modified representation of the odor is generated[12,13]. The processing begins in the olfactory epithelium, where odor molecules are detected by specific neural receptors. This detection stimulates neurons composing the olfactory bulb, and a spatiotemporal pattern of neural activity appears. Each odor produces a distinct activity pattern. Then, the odor is recognized or learned due to the interaction with the olfactory cortex[12,13].

There are several rotation-invariant patter recognition systems using classical signal processing techniques [4,5,17], and Hopfield[2,7] and multilayer perceptron neural networks[1,15].

Here, we use a cellular automaton (CA) for simulating the neural activity of a two-dimensional olfactory bulb and a multi-layer Kohonen neural network

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 444–450, 2006.

(KNN) for playing the role of the olfactory cortex, where the recognition of the spatiotemporal patterns produced by the CA takes place. Our identification scheme is intended to be invariant to rotation of the spatial patterns. Such a scheme and the results obtained in numerical experiments are presented in the next sections.

2 Cellular Automaton

Cellular automata have been employed for representing living systems because it is possible to propose rules of state transitions that are biologically motivated and easily programmable in digital computers[16]. Here, we use a CA to model the olfactory bulb. Assume that each cell (each neuron) forming a two-dimensional lattice is connected to its eight surrounding cells, and that each one can be at rest, excited (firing) or refractory. For simplicity, there are only excitatory synapses. Thus, a resting cell becomes excited if the number of excited cells connected to it (its neighborhood) exceeds or is equal to the threshold L. The connections are local and regular; therefore, the medium is spatially homogeneous. An excited cell spends T_1 time steps firing and then becomes refractory. After T_2 iterations, a refractory cell returns to rest. This model was already used in studies about travelling waves in excitable media[10]. Here, we take $L = 3$, $T_1 = 5$, $T_2 = 4$ and a lattice composed by 100×100 cells (that is, the lattice is a 100×100 matrix).

3 Kohonen Neural Network

The self-organizing map, known as KNN[11], uses an unsupervised learning algorithm in order to translate the similarities of the input data (here produced by the CA) into distance relationships among the neurons composing its output layer. Thus, after appropriately training the network, two input data with similar statistical features stimulate either the same neuron or a neighbor. The learning algorithm is given by:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)h(t)(x_j(t) - w_{ij}(t)) \quad \text{if} \;\; i \in V(i^*) \tag{1}$$

$$w_{ij}(t+1) = w_{ij}(t) \quad \text{if} \;\; i \notin V(i^*) \tag{2}$$

The integer number j labels the position of a neuron in the input matrix. Each neuron in this matrix is connected with all neurons of the output layer by synaptic weights w_{ij}, where the index i expresses the position in the output matrix. The value of the input corresponding to the neuron j is given by x_j. Thus, the vector $\boldsymbol{x}(t)$ represents the complete input at the learning step t. For an input $\boldsymbol{x}$, the output neurons compete among themselves for being activated. The winner is the one presenting the maximum value of the dot product $\boldsymbol{x}.\boldsymbol{w}_i$ and it becomes labeled in this step by i^*. Such a winning neuron defines a neighborhood $V(i^*)$ of radius $R(t)$ centered around it. At each step t, the weights of all neurons pertaining to $V(i^*)$ are updated according to the expressions (1) and (2); the weights of the neurons outside $V(i^*)$ are not altered. This adaptation

rule modifies the weight vector of i^* to more closely resemble the input vector that just stimulated it[11]. And the weight vectors of the other neurons in $V(i^*)$ are modified in order to be stimulated by similar vectors in the following steps, leading to the formation of a topographic map of the input data[11]. The neighborhood function $h(t)$ attains its maximum value at i^* and decays along the distance r from i^*. The function $\eta(t)$ is a learning-rate factor and is taken in the range $0 < \eta(t) < 1$. The values of $R(t)$ and $\eta(t)$ usually decrease as the learning process progresses.

4 Our Scheme

The three different patterns (P1, P2 e P3) obtained from the CA temporal evolution and used in our numerical experiments are shown in Fig. 1, where the black cells represent the excited ones; the white background is formed by resting cells; and refractory gray cells shadow the excited ones. Each pattern is supposed to represent a different odor. These patterns are similar to recordings of neural activity in the olfactory system of, for instance, a rabbit stimulated by smell of banana[9] or a honeybee stimulated by nonanol ($C_9H_{20}O$)[14] (vertebrate and insect olfactory systems present considerable similarities[12]).

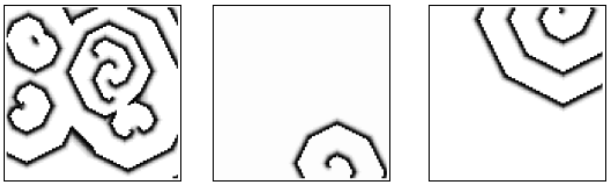


Fig. 1. Pattern P1 (left), P2 (centre) and P3 (right)

These patterns are presented to a multi-layer KNN in order to train it. We wish to develop an identification scheme invariant to the rotation of the spatial patterns. Thus, P1 and P1 rotated by 90^o, by 180^o, and by 270^o should be identified as the same pattern. In our scheme, firstly each pattern (100×100) is splitted into four parts and arranged as shown in Fig. 2, where each part corresponds to a 50×50 matrix; and each one is used as input to four KNN with 10×10 output matrix. After training these networks (see details below) using P1, P2 and P3, then each pattern is rotated by 90^o, and the training algorithm is applied again, until all patterns are rotated three times. The initial values of the synaptic weights w_{ij} for a network in the second layer are randomly picked; however, these initial values are the same for all four networks. In the second layer, the winning neurons are obtained after applying the input patterns. Therefore, P1 determines four winning neurons, one neuron in each 10×10 matrix; P1 rotated by 90^o corresponds to other four neurons; P1 rotated by 180^o to other four neurons; and P1 rotated by 270^o to more four neurons. These

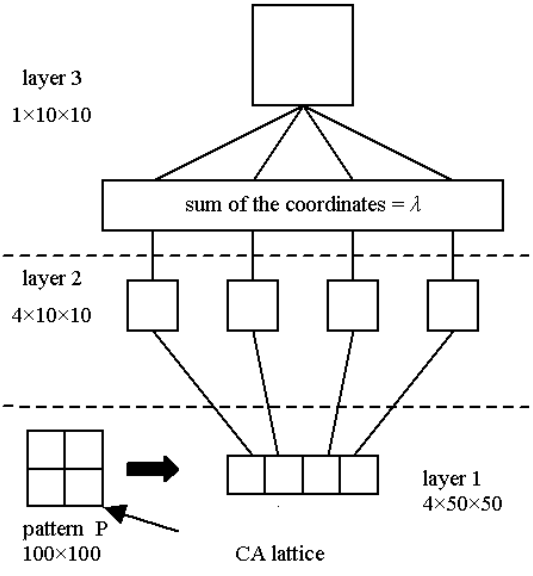


Fig. 2. Proposed scheme for pattern recognition invariant to rotation

16 numbers (the coordinates of the 16 winning neurons in the second layer) identify P1 and its three rotations. Then, these 16 numbers are added, and this sum is called λ. Thus, P1 is related to the sum λ_1; P2 is related to the sum λ_2 and P3 to the sum λ_3. These numbers λ_i $(i = 1, 2, 3)$ are used to form the vectors $\boldsymbol{a}_i = (\lambda_i, \lambda_i, \lambda_i, \lambda_i)$, which are the inputs for the third layer. The output of the third layer is also a 10×10 matrix. It is in this third layer that the pattern identification is accomplished. We find that this scheme is able of identifying a pattern and its three rotations as the same pattern.

For training the second layer, each pattern is presented T times and the functions $h(t)$ and $\eta(t)$ are chosen as:

$$h(t) = 1 - r(t)/R(t) \tag{3}$$

$$\eta(t) = 1 - 2t/(3TR(t)) \tag{4}$$

where $0 \leq r(t) \leq R(t)$ and $1 \leq t \leq T$, with $T = 44$. The neighborhood radius $R(t)$ is initially taken as 10 and is kept constant during $T/11$ steps of training; then $R(t)$ is diminished of 1 and kept constant for more $T/11$ steps and so on. The learning factor $\eta(t)$ is taken as a linearly decreasing function of t.

Finally, the vectors $\boldsymbol{a}_i$ $(i = 1, 2, 3)$ are used as the inputs for the third layer. After training it following a similar procedure, we obtain that each pattern and the corresponding rotations are represented by the same neuron in the 10×10 output matrix of the third layer. The functions $h(t)$, $\eta(t)$ and $R(t)$, the number of components composing the vectors $\boldsymbol{a}_i$, and the matrix dimensions were chosen after a lot of numerical tests.

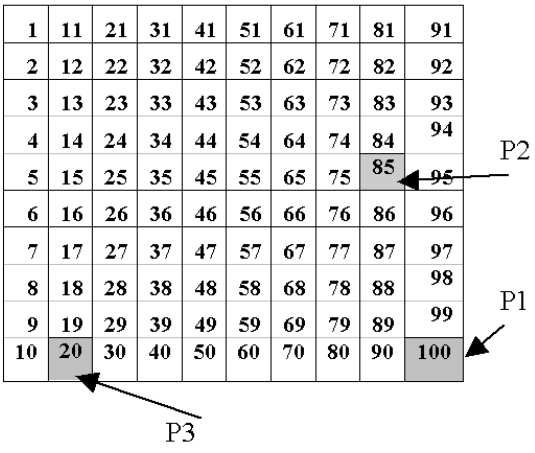


Fig. 3. Winning neurons corresponding to P1, P2 and P3

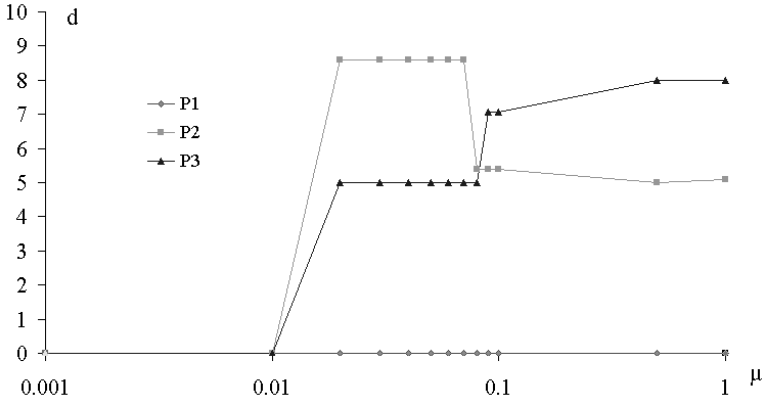


Fig. 4. The distance d as a function of μ

We also employed this scheme for training dynamic patterns. Hence, instead of P1 and its three rotations, we used as inputs the spatial patterns obtained from the CA representing the temporal evolution of this pattern for five consecutive time steps. Thus, there are $3 \times 4 \times 5$ distinct inputs (for each one of the five time steps, there are 3 patterns at 0^o, 90^o, 180^o, 270^o). The winning neurons in the output matrix of the third layer related to the patterns P1, P2 and P3 are shown in Fig. 3. Thus, this identification scheme is able of recognizing a pattern, its rotations, and the similar patterns obtained from its temporal evolution as the same pattern.

The performance of this identification scheme was tested when a noise D is added to the original patterns. Thus, new patterns were created following the expression:

$$\mathrm{Pi}_{noise} = \mathrm{Pi} + \mu \mathrm{D} \tag{5}$$

where Pi ($i = 1, 2, 3$) are the original patterns shown in Fig. 1; D is an uniform noise represented by a 100×100 matrix, and $0 \leq \mu \leq 1$. In order to quantify the performance when noise is present, the distances d between the winning neurons for the original patterns Pi and the winning neurons for the noise images Pi_{noise} were calculated in function of μ. The results are exhibited in Fig. 4. Notice that if $\mu > 1\%$, then the network starts to fail.

5 Conclusions

We developed a scheme slightly inspired on the vertebrate olfactory system for identifying spatiotemporal activity patterns. In this biological system, the representation of odor information is modified at each processing level[12,13]. Here, the representation of the spatiotemporal patterns are also modified at each level (layer): a pattern and its three rotations are firstly translated in 16 numbers (the coordinates of the winning neurons in the second output layer), that are represented by a unique number λ (the sum of the 16 coordinates), that is used for forming the vector $\boldsymbol{a}$ employed for training the last neural network where the recognition will be actually performed. Such a scheme is based on a multi-layer KNN, which is able of identifying dynamic patterns produced by the temporal evolution of the CA. The identification is invariant to the rotation of the patterns and presents good performance, even when a noise lower than 1% is added. Notice that this scheme can fail if two distinct patterns give the same number λ. The chance of occurring this coincidence can be reduced by splitting the original 100×100 pattern into more parts, because in this case there are more coordinates to be summed in order to obtain λ.

Acknowledgments

LHAM is partially supported by CNPq.

References

1. Agarwal, S., Chaudhuri, S.: Determination of aircraft orientation for a vision-based system using artificial neural networks. J. Math. Imaging Vis. **8** (1998) 255–269.
2. Antonucci, M., Tirozzi, B., Yarunin. N.D., Dotsenko, V.S.: Numerical-simulation of neural networks with translation and rotation-invariant pattern-recognition. Int. J. Mod. Phys. B **8** (1994) 1529–1541.
3. Breakspear, M.: Perception of odors by a nonlinear model of the olfactory bulb. Int. J. Bifurcat. Chaos **11** (2001) 101–124.
4. Chen, G.Y., Bui, T.D., Krzyzak, A.: Rotation invariant pattern recognition using ridgelets, wavelet cycle-spinning and Fourier features. Pattern Recognition **38** (2005) 2314–2322.

5. Chiu, C.F., Wu, C.Y.: The design of rotation-invariant pattern recognition using the silicon retina. IEEE J. Solid-State Circuits **32** (1997) 526–534.
6. Claverol, E.T., Brown, A.D., Chad, J.E.: A large-scale simulation of the piriform cortex by a cell automaton-based network model. IEEE Trans. Biomed. Eng. **49** (2002) 921–935.
7. Dotsenko V.S.: Neural networks: translation-, rotation- and scale-invariant pattern recognition. J. Phys. A: Math. Gen. **21** (1988) L783-L787.
8. Freeman, W.J.: Simulation of chaotic EEG patterns with a dynamic model of the olfactory system. Biol. Cybern. **56** (1987) 139–150.
9. Freeman, W.J.: The physiology of perception. Sci. Am. **264**(2) (1991) 78–85.
10. Greenberg, J.M., Hassard, B.D., Hasting, S.P.: Pattern formation and periodic structures in systems modeled by reaction-diffusion equations. Bull. Math. Soc. **84** (1978) 1296–1327.
11. Kohonen, T.: The self-organizing map. Proc. IEEE **78** (1990) 1464–1480.
12. Korsching, S.: Olfactory maps and odor images. Curr. Opin. Neurobiol. **12** (2002) 387–392.
13. Lledo, P.M., Gheusi, G., Vincent, J.D.: Information processing in the mammalian olfactory system. Physiol. Rev. **85** (2005) 281–317.
14. Sachse, S., Galizia, C.G.: Role of inhibition for temporal and spatial odor representation in olfactory output neurons: a calcium imaging study. J. Neurophysiol. **87** (2002) 1106–1117.
15. Sheng, Y., Lejeune, C.: Invariant pattern recogniton using Fourier-Mellin transforms and neural networks. J. Optics (Paris) **22** (1991) 223-228.
16. Wolfram, S.: Theory and Applications of Cellular Automata. Singapore: World Scientific (1986).
17. Zalevsky, Z., Mendlovic, D., Garcia, J.: Invariant pattern recognition by use of wavelength multiplexing. Appl. Opt. **36** (1997) 1059–1063.

Pattern Classification Using Composite Features

Chunghoon Kim and Chong-Ho Choi

School of Electrical Engineering and Computer Science, Seoul National University,
#047, San 56-1, Sillim-dong, Gwanak-gu, Seoul 151-744, Korea
{spinoz, chchoi}@csl.snu.ac.kr

Abstract. In this paper, we propose a new classification method using composite features, each of which consists of a number of primitive features. The covariance of two composite features contains information on statistical dependency among multiple primitive features. A new discriminant analysis (C-LDA) using the covariance of composite features is a generalization of the linear discriminant analysis (LDA). Unlike LDA, the number of extracted features can be larger than the number of classes in C-LDA. Experimental results on several data sets indicate that C-LDA provides better classification results than other methods.

1 Introduction

In pattern classification, an input pattern is represented by a set of features. For better classification, feature extraction has been widely used to construct new features from input features [1]. This reduces the number of features while preserving as much discriminative information as possible. Among the various existing methods, the linear discriminant analysis (LDA) is a well-known method for feature extraction. The objective of LDA is to find an optimal transform that maximizes the ratio of the between-class scatter and the within-class scatter [2]. It is very effective in classifying patterns when the within-class samples are concentrated in a small area and the between-class samples are located far apart.

However, there are two limitations in LDA. If the number of features is larger than the number of training samples, the within-class scatter matrix becomes singular and LDA cannot be applied directly. This problem is called the small sample size (SSS) problem [2]. The other limitation of LDA is that the number of features that can be extracted is at most one less than the number of classes. This becomes a serious problem in binary classification problems, in which only a single feature can be extracted by LDA. These two limitations are derived from the rank deficiencies of the within-class and between-class scatter matrices. In order to solve the SSS problem, several approaches such as the PCA preprocessing [3], null-space method [4], and QR decomposition [5] have been introduced. There are also several approaches that can increase the number of extracted features by modifying the between-class scatter matrix. Fukunaga and Mantock proposed the nonparametric discriminant analysis which uses the nonparametric between-class scatter matrix [2], [6]. Brunzell and Eriksson proposed the Mahalanobis distance-based method [7]. Recently, Loog and Duin [8] used the Chernoff distance [9] between two classes to generalize the between-class scatter matrix.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 451–460, 2006.

On the other hand, Yang et al. proposed 2DLDA using an image covariance matrix for face recognition [10]. An image is represented as a matrix and its transpose is multiplied by itself to obtain an image covariance matrix [11]. Each element in an image covariance matrix is obtained from the covariance of vertical line-shaped input features, and so the size of the image covariance matrix is determined by the number of columns in the image matrix. In their study, an input feature is composed of pixels inside a vertical strip of an image and the dimension of input space is equal to the number of columns in the image matrix.

In this paper, we propose a new feature extraction method using composite features for pattern classification. A composite feature consists of a number of primitive features, each of which corresponds to an input feature in LDA. The covariance of two composite features is obtained from the inner product of two composite features and can be considered as a generalized form of the covariance of two primitive features. It contains information on statistical dependency among multiple primitive features. A new discriminant analysis (C-LDA) using the covariance of composite features is derived, which is a generalization of LDA. Unlike LDA, the SSS problem rarely occurs and the number of extracted features can be larger than the number of classes. In this method, each extracted feature is a vector, whose dimension is equal to that of composite features. Hence, the L1, L2, and Mahalanobis distance metrics are redefined in order to measure the similarity between two samples in the new feature space. In order to evaluate the effectiveness of C-LDA, comparative experiments have been performed using several data sets from the UCI machine learning repository [12]. The experimental results show that the proposed C-LDA provides better classification results than other methods.

In the following section, we explain how patterns are represented and derive C-LDA using composite features. Experimental results are described in Section 3, and the conclusions follow in Section 4.

2 New Classification Method Using Composite Features

2.1 Pattern Representation

An input pattern is usually represented by a vector, whose elements are primitive features. Let U denote a set of primitive features $\{u_1, u_2, \ldots, u_p\}$. Traditional methods such as PCA and LDA use the covariance of two primitive features, which contains second-order statistical information. Now, let us consider a composite feature and the covariance of two composite features. First, u_i's are ordered by some method and we denote the jth primitive feature as s_j. Then, a composite feature $\mathbf{x}_i \in \mathbb{R}^l$ $(i = 1, \ldots, n)$ consists of l primitive features as shown in Fig. 1. Note that the number of composite features n is $p - l + 1$.

In LDA, which uses the covariance of two primitive features, it is irrelevant how those primitive features are ordered. However, it becomes important in the case where composite features are used. Among the many ways of ordering primitive features, we can order the features so that the sum of correlations between neighboring features is maximized. This means that each primitive feature has more correlation with its neighboring features and so a composite feature is composed of primitive features with high

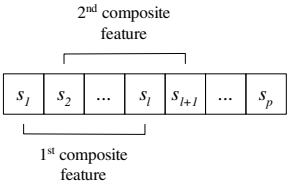


Fig. 1. The composite feature in a pattern, whose size is l

correlation. However, finding such an order is a combinatorial optimization problem and requires large computational effort for large values of p [13]. Instead of finding the optimal solution, we order the primitive features using the greedy algorithm [14]. The greedy ordering algorithm using the correlation coefficient is realized as follows:

1. (Initialization) set $S \longleftarrow \{\ \}$, $U \longleftarrow \{u_1, u_2, \ldots, u_p\}$.
2. (Selection of the first primitive feature) select u_1 from U, and set $s_1 \longleftarrow u_1$, $S \longleftarrow \{s_1\}$, $U \longleftarrow U \backslash \{u_1\}$.
3. (Greedy ordering) for $j = 2$ to $j = p$,
   (a) $\forall u_i \in U$, compute the correlation coefficient between u_i and s_{j-1}.
   (b) choose the primitive feature u_a that has the maximum correlation coefficient, and set $s_j \longleftarrow u_a$, $S \longleftarrow S \cup \{s_j\}$, $U \longleftarrow U \backslash \{u_a\}$.
4. Output the set S.

2.2 Linear Discriminant Analysis Using Composite Features (C-LDA)

Let the set of composite features X be $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_1 = [s_1 \ \ldots \ s_l]^T$, $\mathbf{x}_2 = [s_2 \ \ldots \ s_{l+1}]^T$, and so on. We first consider the covariance matrix C based on the composite features. The element c_{ij} of C is defined as

$$c_{ij} = E[(\mathbf{x}_i - \bar{\mathbf{x}}_i)^T(\mathbf{x}_j - \bar{\mathbf{x}}_j)], \qquad i, j = 1, 2, \ldots, n. \tag{1}$$

where $\bar{\mathbf{x}}_i$ and $\bar{\mathbf{x}}_j$ are the mean vectors of $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively, Note that c_{ij} corresponds to the total sum of covariances between the corresponding elements of $\mathbf{x}_i$ and $\mathbf{x}_j$. The covariance matrix C is computed as [15]

$$C = \frac{1}{N}\sum_{k=1}^{N}(X^{(k)} - M)(X^{(k)} - M)^T, \tag{2}$$

where $X^{(k)} = [\mathbf{x}_1^{(k)} \ \ldots \ \mathbf{x}_n^{(k)}]^T$ for the kth sample, $M = [\bar{\mathbf{x}}_1 \ \ldots \ \bar{\mathbf{x}}_n]^T$, and N is the total number of samples. Note that $X^{(k)} \in \mathbb{R}^{n \times l}$ and $C \in \mathbb{R}^{n \times n}$.

Let us consider the rank of C. Let $\boldsymbol{\chi}_i^{(k)}$, $\bar{\boldsymbol{\chi}}_i \in \mathbb{R}^n$ denote the column vectors of $X^{(k)}$ and M, respectively. Then $X^{(k)} = [\boldsymbol{\chi}_1^{(k)} \ \ldots \ \boldsymbol{\chi}_l^{(k)}]$ and $M = [\bar{\boldsymbol{\chi}}_1 \ \ldots \ \bar{\boldsymbol{\chi}}_l]$. We rewrite (2) as

$$C = \frac{1}{N}\sum_{i=1}^{l}\sum_{k=1}^{N}(\boldsymbol{\chi}_i^{(k)} - \bar{\boldsymbol{\chi}}_i)(\boldsymbol{\chi}_i^{(k)} - \bar{\boldsymbol{\chi}}_i)^T. \tag{3}$$

There are at most Nl linearly independent vectors in (3), and consequently the rank of C is at most Nl. Also, $(\boldsymbol{\chi}_i^{(k)} - \bar{\boldsymbol{\chi}}_i)$'s are not linearly independent because they are related by $\sum_{k=1}^{N}(\boldsymbol{\chi}_i^{(k)} - \bar{\boldsymbol{\chi}}_i) = 0$ for $i = 1, \ldots, l$. Therefore, the rank of C is

$$rank(C) \leq \min(n, (N-1)l). \tag{4}$$

Now, a new discriminant analysis (C-LDA) using the composite features is derived from the within-class scatter matrix C_W and between-class scatter matrix C_B. Assume that each training sample belongs to one of D classes, $c_1, c_2, \ldots, c_D$, and that there are N_i samples for class c_i. As in the covariance matrix C, $C_W \in \mathbb{R}^{n\times n}$ is defined as

$$C_W = \sum_{i=1}^{D} p(c_i)\{\frac{1}{N_i}\sum_{k\in I_i}(X^{(k)} - M_i)(X^{(k)} - M_i)^T\}, \tag{5}$$

where

$$M_i = \frac{1}{N_i}\sum_{k\in I_i} X^{(k)}. \tag{6}$$

Here $p(c_i)$ is a prior probability that a sample belongs to class c_i, and I_i is the set of indices of the training samples belonging to class c_i. $C_B \in \mathbb{R}^{n\times n}$ is also defined as

$$C_B = \sum_{i=1}^{D} p(c_i)\{(M_i - M)(M_i - M)^T\}. \tag{7}$$

As in (4), the rank of C_W is

$$rank(C_W) \leq \min(n, (N-D)l). \tag{8}$$

If $l \geq \frac{p+1}{N-D+1}$, then $rank(C_W) = n$ and the SSS problem does not occur. And the rank of C_B is

$$rank(C_B) \leq \min(n, (D-1)l). \tag{9}$$

In LDA ($l = 1$), the rank is smaller than or equal to $\min(p, (D-1))$, which is the maximum number of features that can be extracted. However, one can extract features up to rank(C_B), which is larger than $D-1$, in C-LDA. It is important to emphasize that we can avoid the problems due to rank deficiencies of C_W and C_B by using composite features.

In order to extract m features in C-LDA, the set of projection vectors $W_L \in \mathbb{R}^{n\times m}$ is obtained as follows:

$$W_L = \arg\max_W \frac{|W^T C_B W|}{|W^T C_W W|}. \tag{10}$$

This can be computed in two steps as in LDA [15]. First, C_W is transformed to an identity matrix by $\Psi\Theta^{-\frac{1}{2}}$, where Ψ and Θ are the eigenvector and diagonal eigenvalue matrices of C_W, respectively. Let C'_W and C'_B denote the within-class and between-class scatter matrices after whitening, respectively. Now $C'_W = I$ and $C'_B = (\Psi\Theta^{-\frac{1}{2}})^T C_B (\Psi\Theta^{-\frac{1}{2}})$. Second, C'_B is diagonalized by Φ, which is the eigenvector matrix of C'_B.

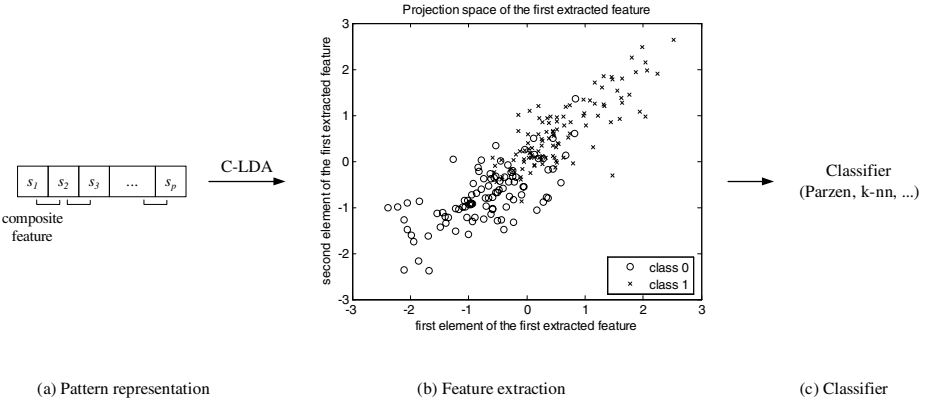


Fig. 2. Classification process using the composite features

Then, W_L consists of m column vectors of $\Psi\Theta^{-\frac{1}{2}}\Phi$, corresponding to the m largest eigenvalues of C'_B. The set of feature vectors $Y^{(k)}$ is obtained from $X^{(k)}$ as

$$Y^{(k)} = {W_L}^T X^{(k)}, \qquad k = 1, 2, \ldots, N, \tag{11}$$

where $Y^{(k)} \in \mathbb{R}^{m \times l}$ has m feature vectors $[\mathbf{y}_1^{(k)}\ \mathbf{y}_2^{(k)}\ \ldots\ \mathbf{y}_m^{(k)}]^T$.

As an example, Fig. 2(b) shows the first extracted feature $\mathbf{y}_1^{(k)}$ obtained from the Sonar data set [12]. For the purpose of visualization, the size of the composite feature is set to 2. Note that $\mathbf{y}_1^{(k)}$ is a vector of dimension 2, which is equal to the size of the composite feature. As can be seen in the figure, there are strong correlations between the two elements of the first extracted feature. It is because each composite feature is composed of two primitive features which are strongly correlated. Figure 2 shows a schematic diagram of the classification process using the composite features.

2.3 Distance Metrics and Classification

In C-LDA, the set of extracted features consists of m vectors of dimension l. Therefore, we need to define the distance metrics in this new subspace. The Manhattan (L1), Euclidean (L2), and Mahalanobis (Mah) distances between $Y^{(j)} = [\mathbf{y}_1^{(j)}\ \ldots\ \mathbf{y}_m^{(j)}]^T$ and $Y^{(k)} = [\mathbf{y}_1^{(k)}\ \ldots\ \mathbf{y}_m^{(k)}]^T$ are defined as

$$\begin{aligned}
d_{L1}(Y^{(j)}, Y^{(k)}) &= \sum_{i=1}^{m} \| \mathbf{y}_i^{(j)} - \mathbf{y}_i^{(k)} \|_2, \\
d_{L2}(Y^{(j)}, Y^{(k)}) &= \{\sum_{i=1}^{m} (\| \mathbf{y}_i^{(j)} - \mathbf{y}_i^{(k)} \|_2)^2\}^{1/2}, \\
d_{Mah}(Y^{(j)}, Y^{(k)}) &= \{\sum_{i=1}^{m} (\frac{\lambda_i}{l})^{-1} (\| \mathbf{y}_i^{(j)} - \mathbf{y}_i^{(k)} \|_2)^2\}^{1/2},
\end{aligned} \tag{12}$$

where $\|\cdot\|_2$ is the 2-norm, and λ_i is the ith largest eigenvalue of a covariance matrix. In (12), the distance between $\mathbf{y}_i^{(j)}$ and $\mathbf{y}_i^{(k)}$ is obtained from the Euclidean distance in the l-dimensional space, irrespective of the metric. The L1 distance is calculated by taking the sum of these between-feature distances, and the L2 distance is calculated by taking the square root of the squared sum of these distances. The Mahalanobis distance can be defined as (12) [16] because the covariance matrix becomes a diagonal matrix with the diagonal elements λ_i's and λ_i corresponds to the total variation of l elements in the ith projection space. In the case of C-LDA, λ_i corresponds to the ith eigenvalue of C', where C' is a covariance matrix after whitening. Since $C' = C'_B + I$ [15], λ_i can be calculated from $\gamma_i + 1$, where γ_i is the ith eigenvalue of C'_B.

For classification, any good classifier can be used but we choose the Parzen classifier and the k-nearest neighbor classifier which are well-known nonparametric methods [1], [17]. Let us consider the Parzen classifier in this subspace. The Parzen classifier is based on the Bayes decision rule and assigns a pattern to the class with the maximum posterior probability [15]. We first derive the posterior probability in the vector space obtained by LDA. Let $\mathbf{y}^{(k)}$ and $\mathbf{z}$ denote the extracted feature vector of the kth training sample and the extracted feature vector of a test sample, respectively. By using the Parzen window density estimation with the Gaussian kernel [18], [19], the posterior probability $\hat{p}(c_j|\mathbf{z})$ can be defined as

$$\begin{aligned}\hat{p}(c_j|\mathbf{z}) &= \frac{\sum_{k\in I_j} exp(-(\mathbf{z}-\mathbf{y}^{(k)})^T \Sigma^{-1}(\mathbf{z}-\mathbf{y}^{(k)})/2h^2)}{\sum_{k=1}^{N} exp(-(\mathbf{z}-\mathbf{y}^{(k)})^T \Sigma^{-1}(\mathbf{z}-\mathbf{y}^{(k)})/2h^2)} \\ &= \frac{\sum_{k\in I_j} exp(-d_M^2(\mathbf{z},\mathbf{y}^{(k)})/2h^2)}{\sum_{k=1}^{N} exp(-d_M^2(\mathbf{z},\mathbf{y}^{(k)})/2h^2)},\end{aligned} \tag{13}$$

where Σ is a covariance matrix, h is a window width parameter, and $d_M(\cdot)$ is the Mahalanobis distance in the vector space.

Now, let us derive the posterior probability in C-LDA. Let $Z = [\mathbf{z}_1\, \mathbf{z}_2\, \ldots\, \mathbf{z}_m]^T$ be the set of extracted features of a test sample, where $\mathbf{z}_i$'s are l-dimensional vectors. As in (13), the posterior probability $\hat{p}(c_j|Z)$ can be defined as

$$\hat{p}(c_j|Z) = \frac{\sum_{k\in I_j} exp(-d_{Mah}^2(Z,Y^{(k)})/2h^2)}{\sum_{k=1}^{N} exp(-d_{Mah}^2(Z,Y^{(k)})/2h^2)}. \tag{14}$$

In (14), we use $h = 1.0$ [18]. Then, the Parzen classifier is to assign Z to class c_t if

$$t = \arg\max_j \hat{p}(c_j|Z), \qquad j = 1, 2, \ldots, D. \tag{15}$$

On the other hand, the k-nearest neighbor classifier is also implemented, where patterns are assigned to the majority class among k nearest neighbors. We use $k = 3$, which is often considered to be a reasonably regularized version of the 1-nearest neighbor classifier [20]. And, the L1 metric is used to measure the distance between two samples.

Table 1. Data Sets Used in the Experiments

Data set	# of classes	# of primitive f.	# of instances
Pima	2	8	768
Breast cancer	2	9	683
Heart disease	2	13	297
Ionosphere	2	34	351
Sonar	2	60	208
Iris	3	4	150
Wine	3	13	178
Glass	6	9	214

3 Experimental Results

In this section, we evaluate the performance of C-LDA. We used eight data sets from the UCI machine learning repository [12] as shown in Table 1. These data sets have been used in many other studies [8], [20], [21].

3.1 Experimental Setup

In order to extract features in classification problems, we implemented two types of C-LDA, i.e., C-LDA(g) and C-LDA(r). In C-LDA(g), the primitive features were ordered by using the greedy algorithm, as explained in Section 2.1. On the other hand, the primitive features were ordered at random in C-LDA(r). In C-LDA, there are two parameters, the size of the composite feature (l) and the number of extracted features (m). For the purpose of comparison, we implemented the principal component analysis (PCA), the linear discriminant analysis (LDA), and the linear discriminant analysis using the Chernoff distance (Cher-LDA) [8]. Here, the design parameter for PCA, LDA, and Cher-LDA is the number of extracted features (m).

For each data set and each classification method, the experiments were conducted in the following way:

1. Ten-fold cross validation was performed in each experiment. We performed 10-fold cross validation 10 times and computed the average classification rate and its standard deviation. Additionally, the Ionosphere and Sonar data sets were split into training and test sets as described in [12], and the classification rates for test sets were computed.
2. Each primitive feature in the training set was normalized to have zero mean and unit variance, and the primitive features in the test set were also normalized using the means and variances of the training set.
3. For classifiers with the extracted features, we used the Parzen classifier (Parzen) and the k-nearest neighbor classifier with $k = 3$ (3-nn) as described in Section 2.3.
4. The optimal parameter values (l^*, m^*), with which each classification method showed the best performance, were recorded.

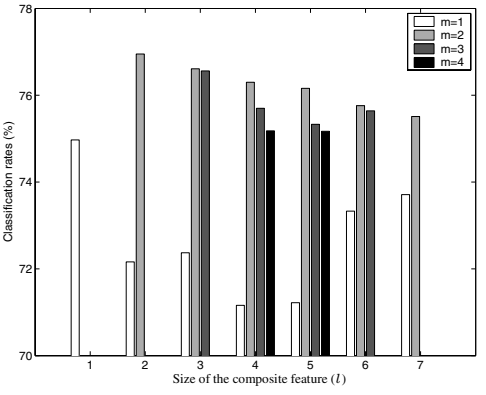


Fig. 3. Classification rates of C-LDA(g) for various values of l and m (Pima)

3.2 Classification Results

We first used the Pima data set for the performance evaluation. Figure 3 shows the classification results of C-LDA(g) using the Parzen classifier. Since there are 8 primitive features in the data set, l was varied from 1 to 7. When $l = 1$, only one feature can be extracted because the rank of C_B is 1, and C-LDA becomes LDA. As l increases, m can be larger. If l is larger than 5, C_B has full rank. In the figure, we observed the following; 1)The classification rates of C-LDA(g) are 75$\sim$77% for $l \geq 2$ and $m \geq 2$. This implies that the classification rates are not very sensitive to l and m, which is a desirable property. 2)Larger values of l and m do not necessarily show better performance. When both l and m are 2, the best classification rate of C-LDA(g) is 77.0%.

The classification results for the Pima data set by the other methods are displayed in Table 2. The results show that C-LDA(g) with the Parzen classifier performed best.

We also tested all the methods on the other seven data sets. The best result for each data set is indicated in boldface. As can be seen in Table 2, C-LDA shows better performance than PCA and LDA. In case of relatively easy problems such as the Iris and Wine data sets, C-LDA shows the best performance with $l = 1$. It is noted that C-LDA with $l = 1$ corresponds to LDA. When C-LDA gives good results with small l, LDA also performs well. However, in other cases, LDA gives poor results. Especially in the case of Sonar data set, LDA performs even worse than PCA. This means that $D-1$ extracted features do not contain sufficient information when D is small. On the other hand, we can extract more than $D - 1$ features in the case of C-LDA, as explained in Section 2.2. This seems to make C-LDA outperform LDA. For the Breast cancer, Ionosphere, Iris, and Wine data sets, Cher-LDA also shows good performance. However, Cher-LDA gives poor results for the Sonar and Glass data sets. The last rows in Table 2(a) and Table 2(b) show the average classification rates for all data sets, in which C-LDA(g) with the Parzen classifier shows the best result of 89.6%. On average, C-LDA(r) performs comparably to C-LDA(g).

Table 2 also shows the optimal parameter value, with which each classification method showed the best performance. We can see that l^* of C-LDA(r) is larger than that

Table 2. Classification Rates and Optimal Parameters

(a) Parzen classifier

Data set	C-LDA(g)		C-LDA(r)		PCA		LDA		Cher-LDA	
	rate (%)	l^*/m^*	rate (%)	l^*/m^*	rate (%)	m^*	rate (%)	m^*	rate (%)	m^*
Pima	**77.0±0.2**	2 / 2	76.7±0.4	4 / 4	73.5±0.4	7	75.0±0.4	1	75.1±0.3	1
Breast	96.2±0.1	8 / 1	**96.4±0.1**	8 / 1	93.3±0.4	7	94.1±0.2	1	94.0±0.1	1
Heart	84.2±0.7	1 / 1	**84.5±0.4**	2 / 2	83.7±0.6	4	84.2±0.7	1	83.9±0.5	1
Iono.	**90.3±0.7**	5 / 5	89.3±0.8	14 / 8	85.8±0.5	5	83.2±0.4	1	85.8±0.5	33
Iono. $^+$	96.0	5 / 5	95.4	10 / 3	96.0	6	91.4	1	**96.7**	16
Sonar	**87.8±1.5**	28 / 12	86.8±0.5	57 / 2	85.6±1.3	15	75.6±1.9	1	79.4±2.0	33
Sonar $^+$	**97.1**	28 / 8	**97.1**	58 / 3	93.3	17	77.9	1	82.7	59
Iris	97.6±0.3	1 / 1	97.6±0.3	1 / 1	92.1±0.5	1	97.6±0.3	1	**97.9±0.5**	1
Wine	99.3±0.4	1 / 2	99.3±0.4	1 / 2	98.0±0.5	5	99.3±0.4	2	**99.6±0.4**	2
Glass	**70.8±0.7**	5 / 3	69.3±1.3	3 / 4	65.7±0.6	7	60.6±1.4	5	64.0±1.1	9
Average	**89.6**		89.2		86.7		83.9		85.9	

(b) 3-nn classifier

Data set	C-LDA(g)		C-LDA(r)		PCA		LDA		Cher-LDA	
	rate (%)	l^*/m^*	rate (%)	l^*/m^*	rate (%)	m^*	rate (%)	m^*	rate (%)	m^*
Pima	73.5±0.8	4 / 2	**74.5±0.6**	4 / 4	72.0±1.1	8	72.4±0.7	1	73.0±1.8	1
Breast	96.9±0.3	6 / 1	96.9±0.3	4 / 2	96.8±0.5	2	96.2±0.5	1	**97.1±0.2**	5
Heart	**83.7±1.1**	9 / 2	83.0±1.4	2 / 2	81.4±1.5	9	80.7±1.2	1	81.1±1.2	1
Iono.	91.4±0.8	3 / 3	89.3±0.6	3 / 3	88.2±0.8	5	85.6±1.0	1	**93.1±0.7**	3
Iono. $^+$	96.7	3 / 2	95.4	9 / 5	95.4	33	79.5	1	**98.0**	7
Sonar	**86.3±0.8**	58 / 1	85.5±0.8	57 / 2	**86.3±0.9**	16	73.2±2.4	1	79.5±2.1	15
Sonar $^+$	**95.2**	16 / 14	91.3	58 / 1	87.5	10	77.9	1	84.6	39
Iris	97.3±0.6	1 / 1	97.3±0.6	1 / 1	95.1±0.5	3	97.3±0.6	1	**97.5±0.4**	2
Wine	98.9±0.5	1 / 2	98.9±0.5	1 / 2	96.4±0.9	7	98.9±0.5	2	**99.3±0.3**	2
Glass	**72.1±1.9**	5 / 3	71.5±1.7	3 / 6	72.0±1.4	8	63.2±3.0	5	68.0±2.0	9
Average	**89.2**		88.4		87.1		82.5		87.1	

($+$: Experimental results are for the given training and test sets instead of 10-fold cross validation.)

of C-LDA(g), especially in the case of Ionosphere and Sonar data sets. In C-LDA(r), it seems that the size of the composite feature should be large.

4 Conclusions

In this paper, we proposed a new linear discriminant analysis using composite features. A composite feature is composed of a number of neighboring primitive features. The covariance of two composite features is obtained from the inner product of two composite features and can be considered as a generalized form of the covariance of two primitive features. The proposed C-LDA has several advantages over LDA. First, more information on statistical dependency among multiple primitive features can be obtained by using the covariance of composite features. Second, the number of extracted features can be larger than the number of classes. Third, C-LDA is expected to provide similar or better performance compared to the other methods in most cases as shown in the previous section. This indicates that the covariance of composite features is able to capture discriminative information better than the covariance of two primitive features.

Acknowledgment

This work was supported by the Korea Research Foundation Grant (KRF-2005-041-D00491) funded by the Korean Government(MOEHRD).

References

1. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical Pattern Recognition: A Review. IEEE Trans. Pattern Analysis and Machine Intelligence. **22** (2000) 4-37
2. Fukunaga, K.: Introduction to Statistical Pattern Recognition. 2nd ed. Academic Press, New York (1990)
3. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Trans. Pattern Analysis and Machine Intelligence. **19** (1997) 711-720
4. Cevikalp, H., Neamtu, M., Wilkes, M., Barkana, A.: Discriminative Common Vectors for Face Recognition. IEEE Trans. Pattern Analysis and Machine Intelligence. **27** (2005) 4-13
5. Ye, J., Li, Q.: A Two-Stage Linear Discriminant Analysis via QR-Decomposition. IEEE Trans. Pattern Analysis and Machine Intelligence. **27** (2005) 929-941
6. Fukunaga, K., Mantock, J.M.: Nonparametric Discriminant Analysis. IEEE Trans. Pattern Analysis and Machine Intelligence. **5** (1983) 671-678
7. Brunzell, H., Eriksson, J.: Feature Reduction for Classification of Multidimensional Data. Pattern Recognition. **33** (2000) 1741-1748
8. Loog, M., Duin, R.P.W.: Linear Dimensionality Reduction via a Heteroscedastic Extension of LDA: The Chernoff Criterion. IEEE Trans. Pattern Analysis and Machine Intelligence. **26** (2004) 732-739
9. Chen, C.H.: On Information and Distance Measures, Error Bounds, and Feature Selection. Information Sciences. **10** (1976) 159-173
10. Yang, J., Zhang, D., Yong, X., Yang, J.-y.: Two-dimensional Discriminant Transform for Face Recognition. Pattern Recognition. **38** (2005) 1125-1129
11. Yang, J., Yang, J.-y.: From image vector to matrix: a straightforward image projection technique-IMPCA vs. PCA. Pattern Recognition. **35** (2002) 1997-1999
12. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases. http://www.ics.uci.edu/ mlearn/MLRepository.html (1998)
13. Papadimitriou, C.H., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall, Englewood Cliffs (1982)
14. Kwak, N., Choi, C.-H.: Input Feature Selection for Classification Problems. IEEE Trans. Neural Networks. **13** (2002) 143-159
15. Webb, A.: Statistical Pattern Recognition. 2nd ed. Wiley, West Sussex (2002)
16. Moghaddam, B., Pentland, A.: Probabilistic Visual Learning for Object Representation. IEEE Trans. Pattern Analysis and Machine Intelligence. **19** (1997) 696-710
17. Fukunaga, K., Hummels, D.M.: Bayes Error Estimation Using Parzen and k-NN Procedures. IEEE Trans. Pattern Analysis and Machine Intelligence. **9** (1987) 634-643
18. Parzen, E.: On Estimation of a Probability Density Function and Mode. The Annals of Mathematical Statistics. **33** (1962) 1065-1076
19. Kim, C., Oh, J., Choi, C.-H.: Combined Subspace Method Using Global and Local Features for Face Recognition. Proc. Int'l Joint Conf. Neural Networks. (2005) 2030-2035
20. Veenman, C.J., Reinders, M.J.T.: The Nearest Subclass Classifier: A Compromise Between the Nearest Mean and Nearest Neighbor Classifier. IEEE Trans. Pattern Analysis and Machine Intelligence. **27** (2005) 1417-1429
21. Toh, K.-A., Tran, Q.-L., Srinivasan, D.: Benchmarking a Reduced Multivariate Polynomial Pattern Classifier. IEEE Trans. Pattern Analysis and Machine Intelligence. **26** (2004) 740-755

Towards a Control Theory of Attention

John G. Taylor

King's College, Dept. of Mathematics,
Strand, London WC2R 2LS, UK
john.g.taylor@kcl.ac.uk

Abstract. An engineering control approach to attention is developed here, based on the original CODAM (COrollary Discharge of Attention Movement) model. Support for the existence in the brain of the various modules thereby introduced is presented, especially those components involving an observer. The manner in which the model can be extended to executive functions involving the prefrontal cortices is then outlined, Finally the manner in which conscious experience may be supported by the architecture is described.

1 Introduction

Attention, claimed William James, is understood by everybody. But it is still unclear how it works in detail and it is still trying to be understood by attention researchers in a variety of ways. This is partly because most of the processes carried out by the brain involve attention in one way or another, but are complex in their overall use of the many different modules present in the brain. This complexity has delayed the separation of these active networks of modules into those most closely involved in attention and those which are lesser so. However considerable progress has now occurred using brain imaging and it has been shown convincingly that there are two regions of brain tissue involved in attention: those carrying activity being attended to and those doing the attending [1, 2].

The modules observed as being controlled by attention are relatively easy to understand: they function so as to have attended activity being amplified by attention and unattended activity reduced. This is a filter process, so that only the attended activity becomes activated enough to become of note for higher level processing. It is the higher level stage that is of concern in this paper. That is now thought to occur by some sort of threshold process on the attended lower-level activity. Attended activity above the threshold is thought to gain access to one of various working memory buffers in posterior sites (mainly parietal). The resulting buffered activity is then accessible to manipulation by various executive function sites in prefrontal cortex.

It is how these executive functions work that is presently becoming of great interest. Numerous studies are showing how such functions as rehearsal, memory encoding and retrieval and others depend heavily on attention control sites. This is to be expected if the executive functions themselves are under the control of attention, which enables the singling out, by the attention amplification/inhibition process, of

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 461 – 470, 2006.

suitable manipulations of posterior buffered activity (and its related lower level components). Thus rehearsal itself could be achieved by having attention drawn to the decay below a threshold of buffered activity, thereby amplifying it, and so rescuing it from oblivion.

At the same time awareness of stimuli is only achieved if they are attended to. Given a good model of attention, is it possible to begin to understand how the model might begin to explain the most important aspects of awareness?

In this paper I present a brief review of the earlier CODAM engineering control model of attention, and consider its recent support from brain science. I then extend the model so as to be able to handle some of the executive processes involved in higher order cognitive processes. I conclude the paper with a discussion of the way that CODAM and its extensions can help begin to explain how consciousness, and especially the pre-reflective self, can be understood in CODAM terms.

2 The CODAM Engineering Control Model of Attention

Attention, as mentioned in section 1, arises from a control system in higher order cortex (parietal and prefrontal) which initially generates a signal which amplifies a specific target representation in posterior cortex, at the same time inhibiting those of distracters. We apply the language of engineering control theory to this process, so assume the existence in higher cortical sites of an inverse model for attention movement, as an IMC (inverse model controller), the signal being created by use of a bias signal from prefrontal goal sites. The resulting IMC signal amplifies (by contrast gain singling out the synapses from lower order attended stimulus representations) posterior activity in semantic memory sites (early occipital, temporal and parietal cortices). This leads to the following ballistic model of attention control:

Goal bias (PFC) → Inverse model controller IMC (Parietal lobe) → Amplified lower level representation of attended stimulus (in various modalities in posterior CX) *(1)*

We denote the state of the lower level representation as **x**(, t), where the unwritten internal variable denotes a set of co-ordinate positions of the component neurons in a set of lower level modules in posterior cortex. Also we take the states of the goal and IMC modules to be x(, t; goal), x(,t; IMC).

The set of equations representing the processes in equation (1) are

$$\tau dx(goal)/dt = -x(goal) + bias \quad (2a)$$

$$\tau dx(IMC)/dt = - x(IMC) + x(goal) \quad (2b)$$

$$\tau dx(,t)/dt = -x(, t) + w*x((IMC) + w'**x(IMC)I(t) \quad (2c)$$

In (2c) the single-starred quantity w*x denotes the standard convolution product $\int w(r, r')IMC(r')dr'$ and w**x(IMC)I(t) denotes the double convolution product $\int w(r, r', r'')\, x(r'; IMC)I(r'')$, where I® is the external input at r. These two terms involving the weights w and w' and single and double convolution products correspond to the additive feedback and contrast gain suggested by various researchers.

Equation (2a) indicates how a bias signal (from lower level cortex) as in exogenous attention, an already present continued bias as in endogenous attention, or in both a form of value bias as is known to arise from orbito-frontal cortex and amygdala. The goal signal is then used in (2b) to guide the direction of the IMC signal (which may be a spatial direction or in object feature space). Finally this IMC signal is sent back to lower level cortices in either a contrast gain manner (modulating the weights arising from a particular stimulus, as determined by the goal bias, to amplify relevant inputs) or in an additive manner. Which of these two is relevant is presently controversial, so we delay that choice by taking both possibilities. That may indeed be the case.

The amplified target activity in the lower sites is then able to access a buffer working memory site in posterior cortices (temporal and parietal) which acts as an attended state estimator. The access to this buffer has been modelled in the more extended CODAM model [2, 3] as a threshold process, arising possibly from two-state neurons being sent from the down to the up-state (more specifically by two reciprocally coupled neurons almost in bifurcation, so possessing long lifetime against decay of activity). Such a process of threshold access to a buffer site corresponds to the equation

$$x(WM) = xY[x - \text{threshold}] \tag{3}$$

where Y is the step function or hard threshold function. Such a threshold process has been shown to occur by means of modelling of experiments on priming [4] as well as in detailed analysis of the temporal flow of activity in the attentional blink (AB) [5]; the activity in the buffer only arises from input activity above the threshold. Several mechanisms for this threshold process have been suggested but will not occupy us further here, in spite of their importance.

The resulting threshold model of attended state access to the buffer working memory site is different from that usual in control theory. State estimation usually involves a form of corollary discharge of the control signal so as to allow for rapid updating of the control signal if any error occurs. But the state being estimated is usually that of the whole plant being controlled. In attention it is only the attended stimulus whose internal activity representation is being estimated by its being allowed to access the relevant working memory buffer. This is a big difference from standard control theory and embodying the filtration process being carried out by attention. Indeed in modern control theory partial measurement on a state leads to the requirement of state reconstruction for the remainder of the state. This is so-called reduced-order estimation [6]. In attention control it is not the missing component that is important but that which is present as the attended component.

The access to the sensory buffer, as noted above, is aided by an efference copy of the attention movement control signal generated by the inverse attention model. The existence of an efference copy of attention was predicted as being observable by its effect on the sensory buffer signal (as represented by its P3) [3]; this has just been observed in an experiment on the Attentional Blink, where the N2 of the second target is observed to inhibit the P3 of the first when T2 is detected. [3, 4, 5].

The ballistic model of (1) is extended by addition of a copy signal – termed corollary discharge - of the attention movement control signal (from the IMC), and used to help speed up the attention movement and reduce error in attention control [2, 3]. The corollary discharge activity can be represented as

$$x(CD) = x(IMC) \tag{4}$$

The presence of this copy signal modifies the manner in which updates are made to the IMC and to the Monitor

$$\tau dx(IMC)/dt = - x(IMC) + x(goal) + w''*x(CD) \tag{5a}$$

$$\tau dx(,t)/dt = -x(, t) + w*x((IMC) + w'**x(IMC)I(t) + x(MON) \tag{5b}$$

$$x(MON) = |x(goal) – x(CD)| + |x(goal) – x(WM)| \quad 5(c)$$

where the monitor is set up so as to take whichever is first of the error signals from the corollary discharge and the buffer activations, but then discard the first for the latter when it arrives (the first having died away in the meantime).

It is the corollary discharge of the attention control model that is beyond that of earlier models, such as of 'biased competition' [7]. It is important to appreciate this as acting in two different ways (as emphasized in [2, 3]):

1) As a contributor to the threshold 'battle' ongoing in the posterior buffer in order to gain access by the attended lower level stimulus. In [2, 3] it was conjectured that this amplification occurred by a direct feedback of a corollary discharge copy of the IMC signal to the buffer (at the same time with inhibition of any distracter activity arriving there).

2) Used as a temporally early proxy for the attention-amplified stimulus activity, being used in a monitor module to determine how close the resulting attended stimulus achieves the pre-frontally held goal.

Both of these processes were shown to be important in a simulation of the attentional blink [2]; the spatially separated and temporally detailed EEG data of [4] required especially the first of these as the interaction of the N2 of the second target T2 and the P3 of the first target T1.

The resulting CODAM model [1, 2, 8] takes the form of figure 1.

Visual input enters by the INPUT module and feeds to the object map. At the same time this input alerts exogenous goals which alert the attention movement generator IMC so as to amplify the input to the object map. The corollary discharge of the IMC signal is sent to a corollary discharge short-term buffer, which is then used either to aid the access to the WM buffer site of the object map activity, or to update the error monitor (by comparison of the corollary discharge signal with an endogenous goal) so as to boost the attention signal in the IMC so as to better achieve the access of the posterior activation in the object map of the attended stimulus so it achieves access to the WM buffer.

Numerous other features have been added to the CODAM model:

a) More detailed perception/concept processing system (GNOSYS)
b) Addition of emotional evaluation modules, especially modeled on the amygdala [9]
c) Addition of a value-learning system similar to the OFC [10]

The relation of this approach contained in equations to standard engineering control theory is summarised in table 1.

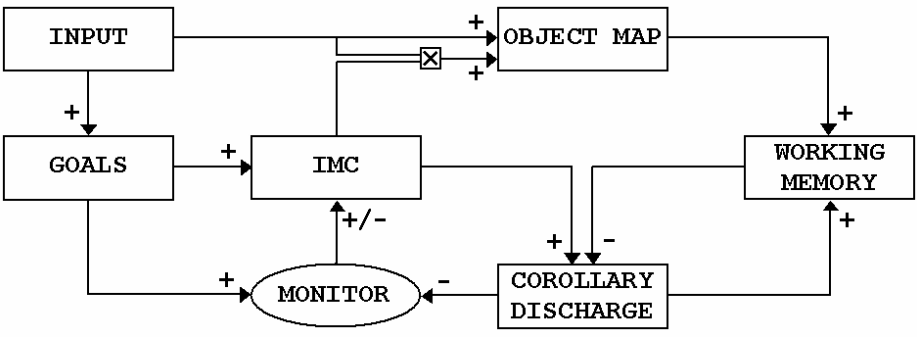


Fig. 1. The CODAM Model

Table 1. Comparison of Variables in Engineering Control Theory and Attetion

Variable	In Engineering control	In Attention
x(, t)	State of plant	State of lower level cortical activity
x(IMC)	Control signal to control plant in some manner	Control signal to move attention to a spatial position or to object features
x(goal)	Desired state of plant	Desired goal causing attention to move
x(CD)	Corollary discharge signal to be used for control speed-up	Corollary discharge to speed-up attention movement
x(WM)	Estimated state of plant (as at present time or as predictor for future use) often termed an observer	Estimated state of attended lower level activity (at present time or as predictor for future use)

We note in table 1 that the main difference between the two columns is in the entries in the lowest row, where the buffer working memory in attention control contains an estimate of only the state of the attended activity in lower level cortex; this is clearly distinguished from that for standard engineering control theory, where the estimated state in the equivalent site is that of the total plant and not just a component of it. There may in control theory be an estimate of the unobserved state of the plant only [6], but that is even more different from attention, where the estimate is only of the attended – so observed – state of lower level brain activity.

3 Executive Functions Under Attention Control

There are numerous executive functions of interest. These arise in reasoning, thinking and planning, including:

1) Storage and retrieval of memories in hippocampus (HC) and related areas;
2) Rehearsal of desired inputs in working memory;
3) Comparison of goals with new posterior activity;.
4) Transformation of buffered material into a new, goal-directed form (such as spatial rotation of an image held in the mind);
5) Inhibition of pre-potent responses [11];
6) The development of forward maps of attention in both sensory and motor modalities, so that possibly consequences of attended actions on the world can be imagined, and used in reasoning and planning;
7) Determination of the value of elements of sequences of sensory-motor states as they are being activated in forward model recurrence;
8) Learning of automatic sequences (chunks) so as to speed up the cognitive process

The rehearsal, transformation, inhibition and retrieval processes are those that can be carried out already by a CODAM model [2, 3] (with additional hippocampus for encoding & retrieval). CODAM can be used to set up a goal, such as the transformed state of the buffered image, or its preserved level of activity on the buffer, and transform what is presently on the buffer by the inverse attention controller into the desired goal state. Such transformations arise by use of the monitor in CODAM to enable the original image to be transformed or preserved under an attention feedback signal, generated by an error signal from the monitor and returning to the inverse model generating the attention movement control signal so as to modify (or preserve) attention and hence what is changed (or held) in the buffer, for later report. Longer term storage of material for much later use would proceed in the HC, under attention control. The comparison process involves yet again the monitor of CODAM. The use of forward models mentioned in (6) allows for careful planning of actions and the realization and possible valuation of the consequences. Multiple recurrence through forward models and associated inverse model controllers allow further look-ahead, and prediction of consequences of several further action steps. Automatic processing is created by sequence learning in the frontal cortex, using FCX → basal ganglia → Thalamus → FCX, as well as with Cerebellum involvement, so as to obtain the recurrent architecture needed for learning chunks (although shorter chunks are also learnt in hippocampus). Attention agents have been constructed {12], and most recently combined with reward learning [13].

Cognitive Architecture: A possible architecture is a) CODAM as an attention controller (with both sensory and motor forms and containing forward models) b) Extension of CODAM by inclusion of value maps and the reward error prediction delta; c) Extension of CODAM to include a HC able to be attended to and to learn short sequences d) Further extension of CODAM by addition of cerebellum to act as an error learner for 'glueing' chunked sequences together, with further extension to addition of basal ganglia (especially SNc) so as to have the requisite automated chunks embedded in attended control of sequential progression. The goal systems in PFC are composed of basal ganglia/thalamus architecture, in addition to prefrontal cortex, as in [14], [15], and observed in [16]. This can allow both for cortico-cortico recurrence as well as cortico-basal ganglia-thalamo-cortical recurrence as a source of long-lifetime activity (as well as through possible dopaminergic modulation of prefrontal neuron activity).

4 Modelling the Cognitive Task of Rehearsal

Rehearsal is a crucial aspect of executive function. It allows the holding of various forms of activity buffered in posterior sites to have its lifetime extended for as long as some form of rehearsal continues. As such, delayed response can be made to achieve remembered goals using activity held long past its usual sell-by date. Such a sell-by date is know to occur for the posterior buffered activity by numerous experimental paradigms [17]. The central executive was introduced by Baddeley as a crucial component of his distributed theory of working memory. The modes of action of this rehearsal process were conjectured as being based in prefrontal sites. More recent brain imaging ([18] & earlier references) have shown that there is a network of parietal and prefrontal sites involved in rehearsal. Let us consider the possible mechanism for such rehearsal to occur.

One of the natural processes to use is that of setting up a goal whose purpose is to refresh the posterior buffered activity at the attended site or object if this activity drops below a certain level. Thus if the buffered posterior activity satisfies equations (3) and (5b) then when the activity drops below a threshold then the monitor is turned on and there is the driving of attention back to the appropriate place or object needing to be preserved.

There are a number of components of this overall process which are still unexplored, so lead to ambiguities. These are as follows:

1) Is the rehearsal signal directed back from the rehearsal goal site to the IMC, and thence to boost the decaying but required input activity, or is there a direct refreshing of activity on the posterior WM buffer?

2) This question leads to the further question: is there a distinction between the posterior WM buffer site and that coding for the inputs at semantic level? It is known, for spatial maps, that there is a visuo-spatial sketchpad buffering spatial representations; is this distinct from a shorter-decaying representation of space? Also is there a separate set of object representations from that of an object WMM buffer?

3) A further question is that, if there is refreshment attention directed to the WM buffer, how does this act? Is it by contrast gain on recurrent synaptic weights that are generators of the buffering property? Or does it occur by an additive bias so as to directly boost activity in the buffer WM

The difference between the answers to question 3) above – how attention is fed back to the WM buffered activity to refresh it – can be seen by analysis. For the membrane potential u of a recurrent neuron (with recurrent weight w) in the WM buffer there is the graded simple dynamic equation:

$$du/dt = -u + wf(u) \tag{6}$$

If attention feedback is by contrast gain then the effect in equation (6) is to increase the weight w by a multiplicative factor, as in the double convolution term in equation (2a). This will have one of two possible effects on the steady-state solution to 9^):

a) The bifurcating value (the non-zero steady state solution to u = wf(u)) is increased, so amplifying the final steady state value of the WM buffer activity;
b) If there is no bifurcation, then the decay lifetime of activity in (6) will increase.

But either case a) or b) above will not necessarily help. If bifurcation has occurred, so case a) applies, then there will not be any decay of WM buffer activity, so there is no need for refreshing in the first place. If the system has not bifurcated but has a long lifetime for decay, as in case b) above, an increase in the lifetime may not help boost back the original activity, but only prolong its decay. Thus it would appear that, barring a more complex picture than present in (6), it will be necessary to have some additive feedback to the WM buffer. This would then boost the threshold activity of the WM buffer, and so help prevent its loss (by keeping it above noise). Thus both a contrast gain and an additive feedback mode of action of refreshment attention would enable the WM buffer to hold onto its activity, and so allow later use of the continued activity in the WM buffer.

5 Discussion

We discussed in section 2 the CODAM mode of attention, based on engineering control. We briefly reviewed the CODAM model, and then considered some details of the comparison between attention and standard engineering controls. An important distinction was that the estimated state of the plant (in engineering control terms usually called the observer) was replaced in attention control by the estimated state of the attended activity in lower cortical sites. This difference is crucial, since on this attention-filtered estimate is based the higher-level processing of activity going under the terms of thinking reasoning and planning. Moreover the initial stage of creating these working memory activations involves a process of taking a threshold on incoming attended activity form lower level sites, and this is regarded as a crucial component of the creation of consciousness, A further crucial component is the presence of a 'pre-signal' – the corollary discharge – suggested in CODAM as the basis of the experience of the pre-reflective self.. In section 3 there was a development of the manner in which executive control might be achieved in terms of this attention control architecture.

In section 4 the details of how rehearsal might occur was suggested in terms of an earlier monitoring model [8], together with a refreshment attention rehearsal process. Various alternatives for the way this refreshment attention could function were considered. It is necessary to wait for further data, updating that form [18], [19], [20] in order to be able to distinguish between these various possibilities. In particular the brain site where the refreshment attention signal is created, as well as the site where such refreshment actually occurs, need to be determined. The analysis of solutions of equation (6) showed a variety of mechanisms could lead to quite different dynamical and steady state activity; these could be part of the clue to how these processes occur in specific sites, so allowing regression techniques to be extended to such refreshment processing.

There is much more to be done, especially by the implementation of language, for developing such high-level cognitive processing, beyond the simple outlines of cognitive processing discussed here.

Acknowledgement

The author would like to acknowledge the support of the European Union through the FP6 IST GNOSYS project (FP6-003835) of the Cognitive Systems Initiative, and more recently of the FP6 IST MATHESIS project.

References

[1] Taylor JG (2000) Attentional Movement: the control basis for Consciousness. Soc Neurosci Abstr 26:2231 #839.3
[2] Taylor JG (2003) Paying Attention to Consciousness. Progress in Neurobiology 71:305-335
[3] Fragopanagos N, Kockelkoren S & Taylor JG (2005) Modelling the Attentional Blink Cogn Brain Research (in press)
[4] Taylor JG (1999) Race for Consciousness. Cambridge MA: MIT Press
[5] Sergent C, Baillet S, and Dehaene S (2005). Timing of the brain events underlying access to consciousness during the attentional blink.. Nat Neurosci, September 2005.
[6] Phillips CL & Harbour RD (2000) feedback Control Systems. New Jersey USA: Prentice Hall
[7] Desimone & Duncan (1995) Neural mechanisms of selective visual attention Ann. Rev. Neurosci., 18:193-222
[8] Taylor JG (2005) From Matter to Consciousness: Towards a Final Solution? Physics of Life Reviews 2:1-44
[9] Taylor JG & Fragopanagos N (2005) The interaction of attention and emotion. Neural Networks 18(4) (in press)
[10] Barto A (1995) Adaptive Critics and the basal ganglia. In: Models of Information Processing n the Basal Ganglia. JC Houk, J Davis & DC Beiser (ditors). Cambridge MA: MIT Press.
[11] Houde O & Tzourio-Mazayer N (2003) Neural foundations of logical and mathematical cognition. Nat Rev Neuroscience 4:507-514
[12] Kasderidis S & Taylor JG (2004) Attentional Agents and Robot Control. International Journal of Knowledge-based & Intelligent Systems 8:69-89
[13] Kasderidis S & Taylor JG (2005) Combining Attention and Value Maps. Proc ICANN05 to appear)
[14] Taylor N & Taylor JG (2000) Analysis of Recurrent Cortico-Basal-Ganglia-Thalamic Loops for Working Memory. Biological Cybernetics 82_415-432
[15] Monchi O, Taylor JG & Dagher A (2000) A neural model of working memory processes in normal subjects, Parkinson's disease and schizophrenia for fMRI design and predictions. Neural Networks 13(8-9):963-973
[16] Monchi O, Petrides M, Doyon J, Postuma RB, Worsley K & Dagher A (2004) Neural Bases of Set Shifting Deficits in Parkinson's Disease. Journal of Neuroscience 24:702-710
[17] Baddeley A (1986) Working Memory. Oxford: Oxford University Press

[18] Lepstein J, Griffin IC, Devlin JT & Nobre AC (2005) Directing spatial attention in mental representations: Interactions between attention orienting and working-memory load. NeuroImage 26:733-743
[19] Yoon JH, Curtis CE & D'Esposito MD (2006) Differential effects if distraction during working memory on delay-period activity in the prefrontal cortex and the visual association cortex. NeuroImage 29:1117-1126
[20] Xu Y & Chun MM (2006) Dissociable neural mechanisms supporting visual shot-term memory for objects. Nature 440:91-95

Localization of Attended Multi-feature Stimuli: Tracing Back Feed-Forward Activation Using Localized Saliency Computations

John K. Tsotsos

Dept. of Computer Science & Engineering, and, Centre for Vision Research
York University, Toronto, Ontario, Canada
tsotsos@cs.yorku.ca

Abstract. This paper demonstrates how attended stimuli may be localized even if they are complex items composed of elements from several different feature maps and from different locations within the Selective Tuning (ST) model. As such, this provides a step towards the solution of the 'binding problem' in vision. The solution relies on a region-based winner-take-all algorithm, a definition of a featural receptive field for neurons where several representations provide input from different spatial areas, and a localized, distributed saliency computation specialized for each featural receptive field depending on its inputs. A top-down attentive mechanism traces back the connections activated by feed-forward stimuli to localize and bind features into coherent wholes.

1 Introduction

Many models have been proposed to explain biological attentive behavior. Some have found utility in computer vision applications for region of interest detection. This article focuses on the Selective Tuning model (ST). Its 'first principles' foundations [1] provided the first formal justification for attention by focusing on computational complexity arguments on the nature of attention, the capacity of neural processes and on strategies for overcoming capacity limitations. The 'first principles' arise because vision is formulated as a search problem (given an image, which subset of neurons best represent image content?). This foundation suggests a specific biologically plausible architecture and its processing stages [1,2]. The architecture includes pyramid representations, hierarchical search and attentive selection.

This contribution focuses on how ST addresses the visual feature binding problem. This is a long-standing problem in cognitive science, first described by Rosenblatt [3]. In vision, as well as in other cognitive tasks, features such as an object's shape, must be correctly associated with other features to provide a unified representation of the object. This is important when more than one object is present in order to avoid incorrect combinations of features. Using the classical view of the binding problem, one can show that for a purely data-directed strategy the problem of finding the subsets of each feature map that correspond to the parts of an object has exponential complexity. It is an instance of the NP-Complete visual matching problem [4] so search is over the powerset of features and locations. In simple detection problems,

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 471 – 480, 2006.

the complexity is manageable by a data-directed strategy because there are few features. In the general case, attentional selection is needed to limit the search.

Part of the difficulty facing research on binding is the confusion over definitions. For example, in Feature Integration Theory [5], location is a feature because it is assumed faithfully represented in a master map of locations. But, this cannot be true; location precision changes layer to layer in any pyramid representation. In any case, an object's edges do not share the same location with its interior. In the cortex, it is not accurate in a Euclidean sense almost anywhere, although the topography is qualitatively preserved [6]. The wiring pattern matters in order to get the right image bits to the right neurons. Thus binding needs to occur layer to layer because location coding changes layer to layer; it is not simply a high-level problem. In addition, features from different representations with different location coding properties converge onto single cells. The resulting abstraction of location information was shown to play an important role in the solution to complexity [1]. It also means that a binding solution requires recovery of location, as opposed to assuming it is a feature.

We define the binding task to involve the solution of three sub-problems: 1) detection (is a given object/event present?); 2) localization (location and spatial extent of detected object/event); and, 3) attachment (explicit object/event links to its constituent components). Further, binding is not a problem in simple situations and only appears when there is sufficient complexity in the image. Specifically, images must contain more than one copy of a given feature, each at different locations, contain more than one object/event each at different locations, and, contain objects/events composed of multiple features and sharing at least one feature type.

Others have proposed solutions to the feature binding problem. The Temporal Synchrony hypothesis proposes recognition of synchronized neural firing patterns [7, 8]. The Biased Competition model proposes task-biased inhibitory competition, plus the responses of higher-order neurons that encode only the attended stimuli, implicitly binds features [9]. The Saliency Map model proposes that feedback modulation of neural activity for visual attributes at the location of selected targets will suffice [10]. The difficulty with these proposals is that none present a mechanism to accomplish binding. There is even the view that recognition does not need attention for binding, and that attention is needed only for task priming and cluttered scenes [11]. This view can be rejected based on the timing observed in attentive tasks among different visual areas. As predicted by ST, higher-level areas show attentive effects before early ones. This is demonstrated in [12, 13] who show this latency pattern and show that attentive effects are mostly after 150ms from stimulus onset, the time period ignored in [11] and by those who study detection tasks exclusively.

The remainder of the paper will briefly present the ST model, and then overview the solution to binding. An example, a discussion of the limitations and behavioural predictions of the model, and a concluding discussion round out the paper.

2 The Selective Tuning Model

The details of the model have been presented previously ([1, 2, 14, 15, 16]) and thus only an overview sufficient to lead into the new work will be presented here.

2.1 The Model

The processing architecture is pyramidal, units receiving both feed-forward and feedback connections from overlapping space-limited regions. It is assumed that response of units is a measure of goodness-of-match of stimulus to a neuron's selectivity. Task-specific bias, when available, allows the response to also reflect the relative importance of the contents of the corresponding receptive field in the scene.

The first stage of processing is a feed-forward pass. When a stimulus is applied to the input layer of the pyramid, it activates all of the units within the pyramid to which it is connected. The result is a feed-forward, diverging cone of activity within the pyramid. The second stage is a feedback pass embodying a hierarchical winner-take-all (WTA) process. The WTA can accept task guidance for areas or stimulus qualities if available but operates independently otherwise. The global winner at the top of the pyramid activates a WTA that operates only over its direct inputs. This localizes the largest response units within the top-level winning receptive field. All of the connections of the visual pyramid that do not contribute to the winner are inhibited. This refines unit responses and improves signal-to-noise ratio. The top layer is not inhibited by this mechanism. The strategy of finding the winners within successively smaller receptive fields, layer by layer, and then pruning away irrelevant connections is applied recursively. The result is the cause of the largest response is localized in the sensory field. The paths remaining may be considered the pass zone while the pruned paths form the inhibitory zone of an attentional beam.

2.2 ST's Winner-Take-All Process

ST's WTA is an iterative process realizable in a biologically plausible manner. The basis for its distinguishing characteristic is that it implicitly creates a partitioning of the set of unit responses into bins of width determined by a task-specific parameter, θ. The partitioning arises because inhibition between units is not based on the value of a single unit but rather on the difference between pairs of unit values.

Competition depends linearly on the difference between unit strengths. Unit A inhibits unit B if the response of A, denoted by $r(A)$, satisfies $r(A) - r(B) > \theta$. Otherwise, A will not inhibit B. The inhibition on unit B is the weighted sum of all inhibitory inputs, each of whose magnitude is determined by $r(A) - r(B)$. It has been shown that this WTA is guaranteed to converge, has well-defined properties with respect to finding largest items, and has well-defined convergence characteristics [2]. The time to convergence is specified by a simple relationship involving θ and the maximum possible value Z across all unit responses. The is because the partitioning procedure uses differences of values, and the smallest units will be inhibited by all other units while the largest valued units will not be inhibited by any unit. As a result, small units are reduced to zero quickly and the time to convergence is determined by the values of the largest and second largest units.

The WTA process has two stages: the first is to inhibit all responses except those in the largest θ-bin; and, the second is to find the largest, strongest responding region represented by a subset of those surviving the first stage. The general form is:

$$G_i(t+1) = G_i(t) - \sum_{j=1,\, j \neq i}^{n} w_{ij}\Delta_{ij} \tag{1}$$

where $G_i(t)$ is the response of neuron i at time t, w_{ij} is the connection strength between neurons i and j, (the default is that all weights are equal; task information may provide different settings), n is the number of competing neurons, and Δ_{ij} is given by:

$$\Delta_{ij} = G_j(t) - G_i(t), \quad \begin{cases} \text{if } 0 < \theta < G_j(t) - G_i(t) \\ 0 \qquad \text{otherwise} \end{cases} . \tag{2}$$

$G_i(0)$ is the feed-forward input to neuron i. Stage 2 applies a second form of inhibition among the winners of the stage 1 process. The larger the spatial distance between units the greater is the inhibition. A large region will inhibit a region of similar response strengths but of smaller spatial extent on a unit-by-unit basis. Equation (1) governs this stage of competition also with two changes: the number of survivors from stage 1 is m, replacing n everywhere, and Δ_{ij} is replaced by:

$$\Phi_{ij} = \mu\left(G_j(t) - G_i(t)\right)\left(1 - e^{-\frac{d_{ij}^2}{d_r^2}}\right), \begin{cases} \text{if} \quad 0 < \theta < \mu\left(G_j(t) - G_i(t)\right)\left(1 - e^{-\frac{d_{ij}^2}{d_r^2}}\right) \\ 0 \qquad \text{otherwise} \end{cases} . \tag{3}$$

μ controls the amount of influence of this processing stage (the effect increases as μ increases from a value of 1), d_{ij} is the retinotopic distance between the two neurons i and j, and d_r controls the spatial variation of the competition.

3 The Selective Tuning Approach to Visual Feature Binding

The binding strategy depends on the hierarchical WTA method to trace back the connections in the network along which feed-forward activations traveled. This provides the solution to the localization problem and links all the component features from different representations of an object via the pass pathways of the attentional beam. The additional elements that comprise this method are now presented.

3.1 Featural Receptive Fields

For single feature maps or for the assumption of a single saliency map [5, 10] the hierarchical WTA described above will suffice. However, in our case, no such assumption is made. Saliency is not a global, homogeneous computation in this framework. A strategy for combining features from different representations and different locations is required. This requires the functionality provided by acknowledging the contributions to a neuron's response from separate locations and separate feature maps. Define the **Featural Receptive Field** (*FRF)* to be the set of all the direct inputs to a neuron. This can be specified by the union of k arbitrarily shaped, contiguous, possibly overlapping sub-fields as

$$FRF = \bigcup_{j=1,k} f_j \quad , \tag{4}$$

where $\{f_j = \{(x_{j,a}, y_{j,a}), a=1,...,b_j\}, j=1,...,k\}$, (x,y) is a location in sub-field f_j, b_j is the number of units in sub-field f_j. The f_j's may be from any feature map, and there may be more than one sub-field from a feature map. F is the set of all sub-field identifiers 1 though k. Response values at each (x,y) location within sub-field $i \in F$ are represented by $r(i,x,y)$.

The *FRF* definition applies to each level of the visual processing hierarchy, and to each neuron within each level. Suppose a hierarchical sequence of such computations defines the selectivity of a neuron. Each neuron has input from a set of neurons from different representations and each of those neurons also have a *FRF* and their own computations to combine its input features. With such a hierarchy of computations, a stimulus-driven feed-forward pass would yield the strongest responding neurons within one representation if the stimulus matches the selectivity of existing neurons, or the strongest responding component neurons in different representations if the stimulus does not match an existing pattern. The result is that the classical receptive field (the region of the visual field in which stimulation causes the neuron to fire) now has internal structure reflecting the locations of the stimulus features.

3.2 Hierarchical WTA Traces Back Feed-Forward Activations

The idea of tracing back connections in a top-down fashion was present, in part, in the Neocognitron model of Fukushima [17]; the first description of the ST hierarchical WTA method was presented in [16].

Fukushima's model included a maximum detector at the top layer to select the highest responding cell and all other cells were set to their rest state. Only afferent paths to this cell are facilitated by action from efferent signals from this cell. The differences between Neocognitron and ST are many. Neural inhibition is the only action of ST, with no facilitation. The Neocognitron competitive mechanism is lateral inhibition at the highest and intermediate levels that finds strongest single neurons thus assuming all scales are represented explicitly, while ST finds regions of neurons removing this unrealistic assumption. For ST, units losing the competition at the top are left alone and not affected at all. ST's inhibition is only within afferent sets to winning units. Finally, Fukushima assumes the top layer is populated by so-called grandmother cells whereas ST makes no such assumption. Overall, the Neocognitron model and its enhancements cannot scale and would suffer from representational and search combinatorics [1].

ST's WTA computation requires a competition among all the representations (feature maps) at the top layer of the pyramids, i.e., there can be multiple pyramids (such as ventral and dorsal stream). Task biases can weight each computation. The type of competition is determined by the relationships among the active representations. Two types are considered here. Two representations are mutually exclusive if, on a location-by-location basis, the two features they represent cannot both be part of the same object or event (eg., an object cannot have a velocity in two directions or two speeds at the same location at the same time). This also implies that the competing *FRF* sub-fields completely overlap in space. Two representations may

co-exist if the two features they represent can both be part of some object or event (eg., an edge may have colour, a line may be at some disparity, features belonging to eyes and co-exist with those from noses, etc.).

The following method is applied at the top of all pyramids at first, then recursively downwards following the *FRF* representations of the winning units. If *F* at some level of the hierarchy contains sub-fields from more than one feature map representing mutually exclusive features (call this subset *A*), then, the two WTA stages represented by Eqs. (1-3) are applied to each sub-field separately. This will yield a winning region within each sub-field, $g_f = \{(x_{i,f}, y_{i,f}) \mid i=1,2,...,n_f\}$, where n_f is the number of locations in the winning region in sub-field *f*. Call this a Type A process. Since the features are mutually exclusive, the winning feature region is the region with the largest sum of responses of its member units. This winning value V_A is given by

$$V_A = \max_{j \in F} \sum_{x,y \in g_j} r(j,x,y). \tag{5}$$

If *F* contains sub-fields representing features that can co-exist at each point (call this subset *B*), then the two stages of the WTA, represented by Eqs. (1-3), are applied to each representation separately. Here, however, the extent of the winning region is the union of all the winning regions. These winning regions are further constrained: each winning region is required to either overlap with, or to be entirely within or entirely enclose, another winning region. Call this the Type B process. The winning value is given by the sum of responses over all of the winning regions,

$$V_B = \sum_{j \in F} \sum_{x,y \in g_j} r(j,x,y) \; . \tag{6}$$

If *F* contains sub-fields representing features that are mutually exclusive (set *A*) as well as features that co-exist (set *B*), a combination of the above strategies is used. The winning value is given by the sum of Equations (5) and (6) and the extent of the winning region is the union of winning regions in sets *A* and *B*.

There is no saliency map in this model. Saliency is a dynamic and task-specific determination and one that may differ between processing layers as required. Further, this does not imply that a feature map must exist for any possible combination of features. Features are encoded separately in a set of maps and the relationships of competition or cooperation among them provide the potential for combinations. Although the above shows two forms of competition, other types can be included.

3.3 Detection, Localization and Attachment

ST seeks the best matching scene interpretations (highest response) as a default (defaults can be tailored to task). This is the set of neurons chosen by the WTA competition throughout the hierarchy. If this happens to match the target of the search, then detection is complete. If not, the second candidate region is chosen and this proceeds until a decision on detection can be made. Localization is accomplished by the downward search to identify the feed-forward connections that led to the neuron's response following the network's retinotopic topology, using the *FRF*s all the way down the hierarchy. *FRF*s provide for a distributed, localized saliency

computation appropriate for complex feature types and complex feature combinations. What is salient for each neuron is determined locally based on its *FRF*; saliency is not a global, homogeneous computation. Once localization is complete for all features, the object is attached to its components through the attention pass beams.

3.4 An Example

A very brief explanation of one example appears here. The full background for this example is available in [2, 14] and due to space limits, cannot be included here. The input is an image sequence that has three graphical objects (textured octagons) in motion; the largest item is translating and rotating, the medium sized object is rotating and the smallest object is translating. It satisfies the constraints for stimulus complexity requiring solution of the binding problem set out earlier. The visual processing hierarchy is specialized for motion alone and contains filter banks that simulate the motion selectivity of areas V1, MT, MST and 7a following experimental observations in the monkey. The V1 layer is selective for translation in 12 directions and 3 speeds (see Fig. 1a). MT, MST and 7a layers have pyramidal abstractions of this translation. MT also includes selectivity for the spatial derivative of local velocity (i.e., the representation is affine motion specific), in 12 gradient directions for each of the 12 directions and 3 speeds of translation. MST includes selectivity for generalized spiral motion (rotations, expansion, contraction and their combinations). 7a represents abstraction of generalized spiral as well as of translation. There are a total of 690 filter types in total (72 in V1, 468 in MT, 72 in MST and 78 in 7a), each operating over the visual field. Thus, there are multiple pyramids in this representation, with multiple top layers (78 top level representations). Generalized spiral neurons in MST have complex *FRF*s, building upon many features from the MT layer. However, there is no representation for the conjunction of translation with generalized spiral motion.

The feature binding process described earlier begins with a cooperative process (Type B) across the output layers: translation and generalized spiral motion can co-exist. This identifies the translation peak and the rotation peak belonging to the combined motion. The two winning regions then begin their downward search following their own *FRF*s. See Fig. 1b. The translation pyramid (on the right side of both sub-figures) needs competitive interaction (Type A) to select the best representations. The generalized spiral pyramid (on the left side of both sub-figures) also begins with competition at the top. Both pathways require competition layer by layer except for the MT layer of spatial derivatives. There, a Type B process is needed in order to find the set of common spatial gradients across a region (a rotating object has homogenous spatial derivatives to local velocity, the derivative direction being perpendicular to the direction of local motion). The winning attentional beams then split and converge on the image plane to show localization of the input stimulus (Fig. 1b). In the continuation of this example, this location would be inhibited to allow the second strongest input item to be found, and so on. The point of this example is to show how a motion not explicitly represented in the system can be found, detected and localized, involving many representations and locations bound by the attention beam.

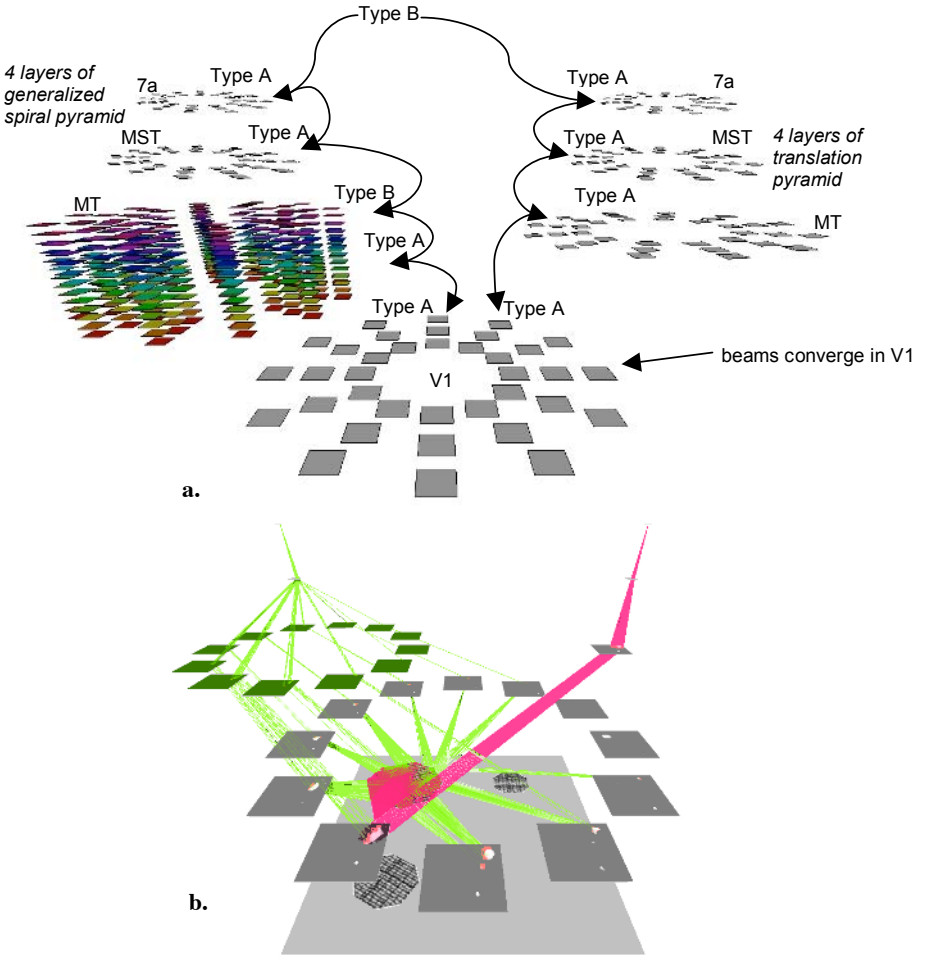


Fig. 1. a. The figure shows the full set of filter banks that are part of the motion processing system. Each small rectangle is one filter type at one pyramid layer. The rings of filters all appear at the same pyramid layer. The center rings of 36 filters show the output of V1; the translation pyramid then continues upwards on the right. The generalized spiral pyramid is on the left. V1 is their common base. The set of 432 coloured rectangles in MT depict the set of velocity gradient filters. The sequence of arrows shows the processing trail of the hierarchical WTA and the types of competition at each stage. **b.** The figure depicts the final configuration of attentive selection for the object that is translating and rotating even though no such feature conjunction has been included in the representation. The largest rectangle at the bottom represents the image plane on which are three textured octagons in motion. The other rectangles represent filter banks that contain the features of the attended stimulus. They are a subset of the full hierarchy of the left side figure with the inhibited ones removed. In some of the larger rectangles, response from the stimuli can be seen. The beams that tie together the filter bank representations are the pass zones of the attentional beam that converge on the largest of the 3 octagons. There are two roots to these beams because there is no single representation for rotating, translating objects.

4 Discussion

The solution to the feature binding problem has remained elusive for almost half a century. This paper proposes a solution and presents a very brief demonstration of the proposal. The binding problem was decomposed into three stages: detection, localization and attachment. The key element for its solution is the method for tracing connections that carry feed-forward activation downward through multiple representations so that they converge on the selected stimulus. This action links all the stimulus' component features within the pass zone of ST's attention beam.

The validation of such a model can be not only computational in the sense of performance on real images (however, see [14]). Such a model can also be validated by showing that it makes counter-intuitive predictions for biological vision that gain experimental support over time. The following predictions, among others, appeared in [1]. 1) Attention imposes a suppressive surround around attended items in space as well as in the feature dimension. 2) Selection is a top-down process where attentional guidance and control are integrated into the visual processing hierarchy. 3) The latency of attentional modulations *decreases* from lower to higher visual areas. 4) Attentional modulation appears wherever there is many-to-one, feed-forward neural convergence. 5) Topographic distance between attended items and distractors affects the amount of attentional modulation. In each of these cases, significance supporting evidence has accrued over the intervening years, recounted in [14, 15].

The binding solution has some interesting characteristics that may be considered as predictions requiring investigation in humans or non-human primates. 1) Given a group of identical items in a display, say in a visual search task, subsets of identical items can be chosen as a group if they fit within receptive fields. Thus, the slope of observed response time versus set size may be lower than expected (not a strictly serial search). 2) There is no proof that selections made at the top of several pyramids will converge to the same item in the stimulus array. Errors are possible if items are very similar, if items are spatially close, or if the strongest responses do not arise from the same stimulus item. 3) Binding errors may be detected either at the top by matching the selections against a target, or if there is no target, by the end of the binding attempt when the pass beams do not converge. The system then tries again; the prediction is that correct binding requires time that increases with stimulus density and similarity. In terms of mechanism, the ST model allows for multiple passes and these multiple passes reflect additional processing time. 4) ST's mechanism suggests that detection occurs before localization and that correct binding occurs after localization. Any interruption of any stage will result in binding errors.

The use of localized, distributed saliency within ST is precisely what the binding problem requires. Saliency is not a global, homogeneous process as in other models. Neurons in different representations that respond to different features and in different locations are selected together, the selection in location and in feature space, and are thus bound together via the pass zone of the attention mechanism. Even if there is no single neuron at the top of the pyramid that represents the concept, the WTA model allows for multiple threads bound through the spatial topology of the network wiring.

References

1. Tsotsos, J.K.: A Complexity Level Analysis of Vision. Behavioral and Brain Sciences Vol 13 (1990) 423-455
2. Tsotsos, J.K., Culhane, S., Wai, W., Lai, Y., Davis, N., Nuflo, F.: Modeling visual attention via selective tuning. Artificial Intelligence Vol 8:1-2 (1995) 507 - 547
3. Rosenblatt, F.: Principles of Neurodynamics: Perceptions and the Theory of Brain Mechanisms. Spartan Books (1961)
4. Tsotsos, J.K.: The Complexity of Perceptual Search Tasks. Proc. International Joint Conference on Artificial Intelligence Detroit (1989) 1571 - 1577
5. Treisman, A., Gelade, G.: A feature-integration theory of attention. Cognitive Psychology Vol 12 (1980) 97-136
6. Felleman, D., Van Essen, D.: Distributed Hierarchical Processing in the Primate Visual Cortex. Cerebral Cortex Vol 1 (1991) 1-47
7. von der Malsburg, C.: The correlation theory of brain function, Internal Rpt. 81-2, Dept. of Neurobiology, Max-Planck-Institute for Biophysical Chemistry, Göttingen, Germany (1981)
8. Gray, C.M.,: The Temporal Correlation Hypothesis of Visual Feature Integration, Still Alive and Well. Neuron Vol 24:1, (1999) 31-47
9. Desimone, R., Duncan, J.: Neural mechanisms of selective visual attention. Annual Reviews of Neuroscience Vol 18 (1995) 193-222
10. Itti, L. Koch, C.: Computational modelling of visual attention. Nature Reviews Neuroscience Vol 2 (2001) 194-204
11. Riesenhuber, M. Poggio, T.: Are Cortical Models Really Bound by the "Binding Problem"? Neuron 1999, Vol 24:1 (1999) 87-93
12. Mehta, A. D. Ulbert, I. Schroeder, C. E.: Intermodal Selective Attention in Monkeys. I: Distribution and Timing of Effects across Visual Areas. Cerebral Cortex Vol 10:4, (2000) 343-358
13. Connor, D.O., Fukui, M., Pinsk, M., Kastner, S.: Attention modulates responses in the human lateral geniculate nucleus, Nature Neurosci.ence Vol 5:11, (2002) 1203–1209
14. Tsotsos, J.K., Liu, Y., Martinez-Trujillo, J., Pomplun, M., Simine, E., Zhou, K.: Attending to Motion, Computer Vision and Image Understanding Vol 100:1-2, (2005) 3 - 40
15. Tsotsos, J.K., Culhane, S., Cutzu, F.: From Theoretical Foundations to a Hierarchical Circuit for Selective Attention. Visual Attention and Cortical Circuits, (2001) 285 – 306, ed. by J. Braun, C. Koch, and J. Davis, MIT Press
16. Tsotsos, J.K.: An Inhibitory Beam for Attentional Selection. in Spatial Vision in Humans and Robots, ed. by L. Harris and M. Jenkin, (1993) 313 - 331, Cambridge University Press (papers from York University International Conference on Vision, June 1991, Toronto)
17. Fukushima, K.: A neural network model for selective attention in visual pattern recognition. Biological Cybernetics Vol 55:1 (1986) 5 - 15

An Attention Based Similarity Measure for Colour Images

Li Chen and F.W.M. Stentiford

University College London, Adastral Park Campus, UK
{l.chen, f.stentiford}@adastral.ucl.ac.uk

Abstract. Much effort has been devoted to visual applications that require effective image signatures and similarity metrics. In this paper we propose an attention based similarity measure in which only very weak assumptions are imposed on the nature of the features employed. This approach generates the similarity measure on a trial and error basis; this has the significant advantage that similarity matching is based on an unrestricted competition mechanism that is not dependent upon a priori assumptions regarding the data. Efforts are expended searching for the best feature for specific region comparisons rather than expecting that a fixed feature set will perform optimally over unknown patterns. The proposed method has been tested on the BBC open news archive with promising results.

1 Introduction

Similarity matching is a basic requirement for the effective and efficient delivery of media data and for the identification of the infringement of intellectual property rights. Considerable effort has been devoted to defining and extracting image signatures, which are based on the assumption that similar images will cluster in a pre-defined feature space [1-5]. It is common for unseen patterns not to cluster in this fashion despite apparently possessing a high degree of visual similarity.

In this research we propose an attention based similarity matching method with application to colour images based on our previous work [6,7]. The approach computes a similarity measure on a trial and error basis; this has the significant advantage that features that determine similarity can match whatever image property is important in a particular region whether it is a colour, texture, shape or a combination of all three. Efforts are expended searching for the best feature for the region rather than expecting that a fixed feature set will perform optimally over unknown patterns in addition to the known patterns. In this context, the proposed method is based on the competitive evolution of matching regions between two images rather than depending on fixed features which are intuitively selected to distinguish different images or cluster similar images. In addition the proposed method can cope with different distortions of images including cropping, resizing, additive Gaussian noise, illumination shift and contrast change. These functions potentially help detect copied images.

The remainder of this paper is arranged as follows. In Section 2, the cognitive visual attention model is presented. Experiments are conducted on BBC open news archive and results are shown in Section 3. Conclusions are addressed in Section 4.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 481 – 487, 2006.

2 Visual Attention Similarity Measure

Studies in neurobiology [8] suggest that human visual attention is enhanced through a process of competing interactions among neurons representing all of the stimuli present in the visual field. The competition results in the selection of a few points of attention and the suppression of irrelevant material. In this context of visual attention, we argue that humans are able to spot anomalies in a single image or similarity between two images through a competitive comparison mechanism, where similar and dissimilar regions are identified and scored by means of a new similarity measure.

Our model of visual attention is based upon identifying areas in an image that are similar to other regions in that same image [6,7]. The salient areas are simply those that are strongly *dissimilar* to most other parts of the image. In this paper we apply the same mechanism to measure the similarity between two different images. The comparison is a flexible and dynamic procedure, which does not depend on a particular feature space which may be thought to exist in a general image database.

Let a measurement $a = (a_1, a_2, a_3)$ correspond to a pixel $x = (x_1, x_2)$ in image A and a function *F* is defined so that $a = F(x)$.

Consider a neighbourhood N of x where

$$N = \left\{ x' \in N \;\; \textit{if and only if} \;\; \left| (x_{\mathrm{i}} - x_{\mathrm{i}}') \right| \le \varepsilon_{\mathrm{i}} \right\}.$$

Select a set (called a fork) of *m* random pixels S_A from *N* where

$$S_A = \{ x_1', x_2', ..., x_{\mathrm{m}}' \}.$$

Select another random pixel *y* in image B and define the fork S_B

$$S_B = \{ y_1', y_2', ..., y_{\mathrm{m}}' \} \text{ where } x - x_{\mathrm{i}}' = y - y_{\mathrm{i}}' \;\; \forall\, i.$$

The fork S_A matches S_B if

$$\left| F_j(x_i) - F_j(x_i') \right| \le \delta_{\mathrm{j}} \quad \forall\, i, j.$$

That is, a match occurs if all colour values (suffix *j*) of corresponding pixels in S_A and S_B are close. The similarity score of a pixel *x* is incremented each time one of a set of *M* neighbourhoods S_A matches a neighbourhood S_B surrounding some *y* in pattern B. This means that pixels *x* in A that correspond to large numbers of matches between a range of *M* neighbouring pixel sets S_A and pixel neighbourhoods somewhere in B are assigned high scores. In Fig. 1, $m = 3$ pixels x' are selected in the neighbourhood of a pixel *x* in pattern A and matched with 3 pixels in the neighbourhood of pixel *y* in pattern B.

A parameter *s* is introduced to limit the area in pattern B within which the location *y* is randomly selected. $s = 2$ defines the dotted region in Fig. 1. This improves the efficiency of the algorithm in those cases where it is known that corresponding regions in the two images are shifted by no more than *s* pixels. In effect *s* represents the maximum expected mis-registration or local distortion between all parts of the two images.

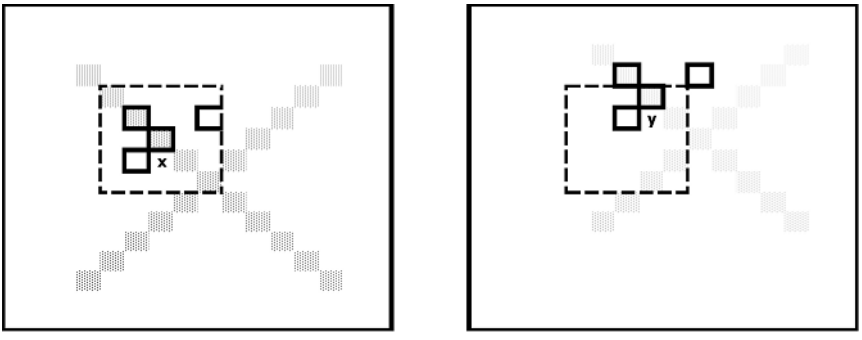


Fig. 1. Neighbourhood at location *x* matching at location *y*

The similarity contributions from all pixel regions in A are summed and normalized to give the total similarity score C_{AB} between images A and B:

$$C_{AB} = \frac{1}{M * \|A\|} \sum_{x \in A} \left(\sum_{M, y \in B} (1 \mid S_A \; matches \; S_B, \; 0 \mid otherwise) \right).$$

3 Experiments

Experiments were carried out on images downloaded from BBC open news archives [9]. 75 videos (more than 230,000 frames) cover different topics including conflicts and wars, disasters, personalities and leaders, politics, science & technology, and sports. Since many frames within a scene differ only slightly, and to utilise the diversity in the database, 2000 non-contiguous frames were extracted by taking every 100th frame from these videos to form the database for image retrieval. 21 images were randomly chosen from the database and 4 distorting transforms were applied to each image (see Fig. 2) including additional Gaussian noise, contrast change, crop and shift, and resize. These distorted images were then added to the image database making a total of 2084 images.

Fig. 3 illustrates the precision and recall performance of the proposed method with 15 queries of the database and M = 20. Recall is the ratio of the number of relevant images retrieved to the total number of relevant images in the database; and it is expressed as:

$$recall = \frac{the \;\; number \;\; of \;\; relevant \;\; images \;\; retrieved}{the \;\; number \;\; of \;\; relevant \;\; images \;\; in \;\; the \;\; database}$$

Precision is the ratio of the number of relevant images retrieved to the total number of irrelevant and relevant images retrieved, and it is defined as:

$$precision = \frac{the \;\; number \;\; of \;\; relevant \;\; images \;\; retrieved}{the \;\; number \;\; of \;\; images \;\; retrieved}$$

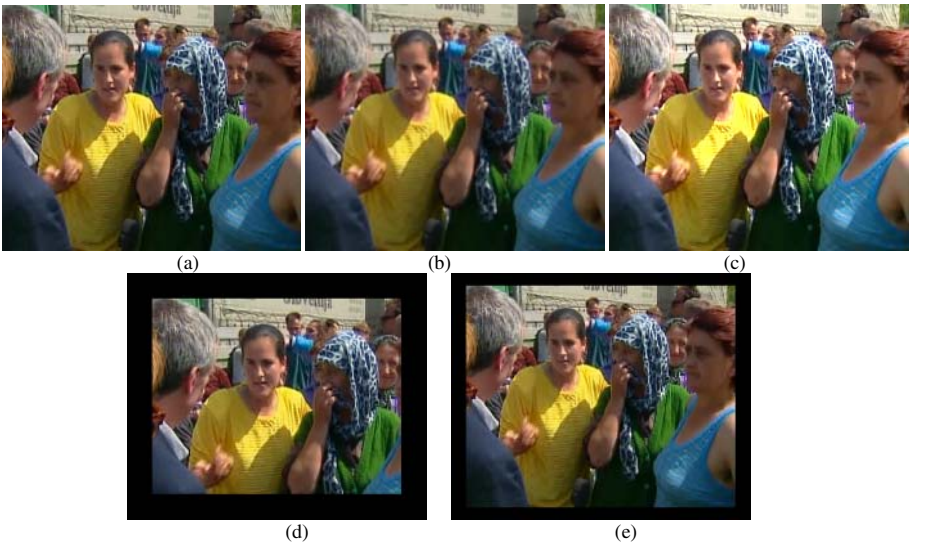


Fig. 1. An example of an image and four transforms: (a) original image (b) with additional radius-1 Gaussian blur (c) with contrast increased 25% (d) cropped and shifted to the right (e) resized to 80% of original image

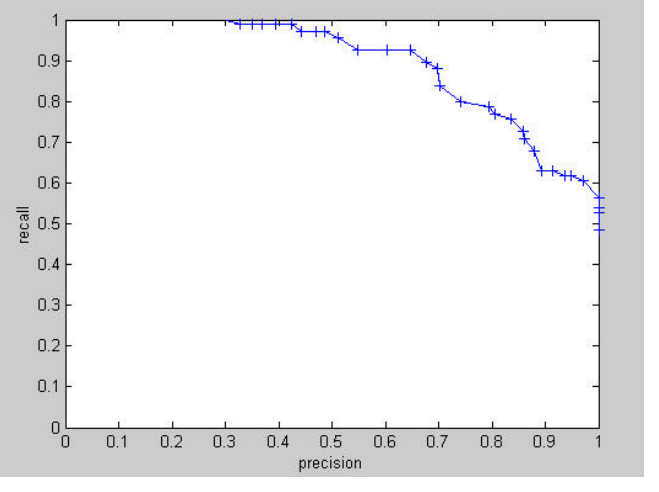


Fig. 2. Recall and precision for retrieval performance

Only very similar frames that were immediately adjacent in time to the query frame were considered to be relevant images.

Fig. 4 shows the relationship between the similarity score and the computation (M) for the original image when compared with itself, the blurred, contrast shifted, cropped and resized versions, a similar image taken from the same video ahead of example frame by 70 frame distance, and two other different images in the database.

It is interesting that for low values of *M* the original is seen to be less similar to itself than to the distorted versions. This repeated an earlier result [6] when it was found that some similarities were found more easily in blurred images.

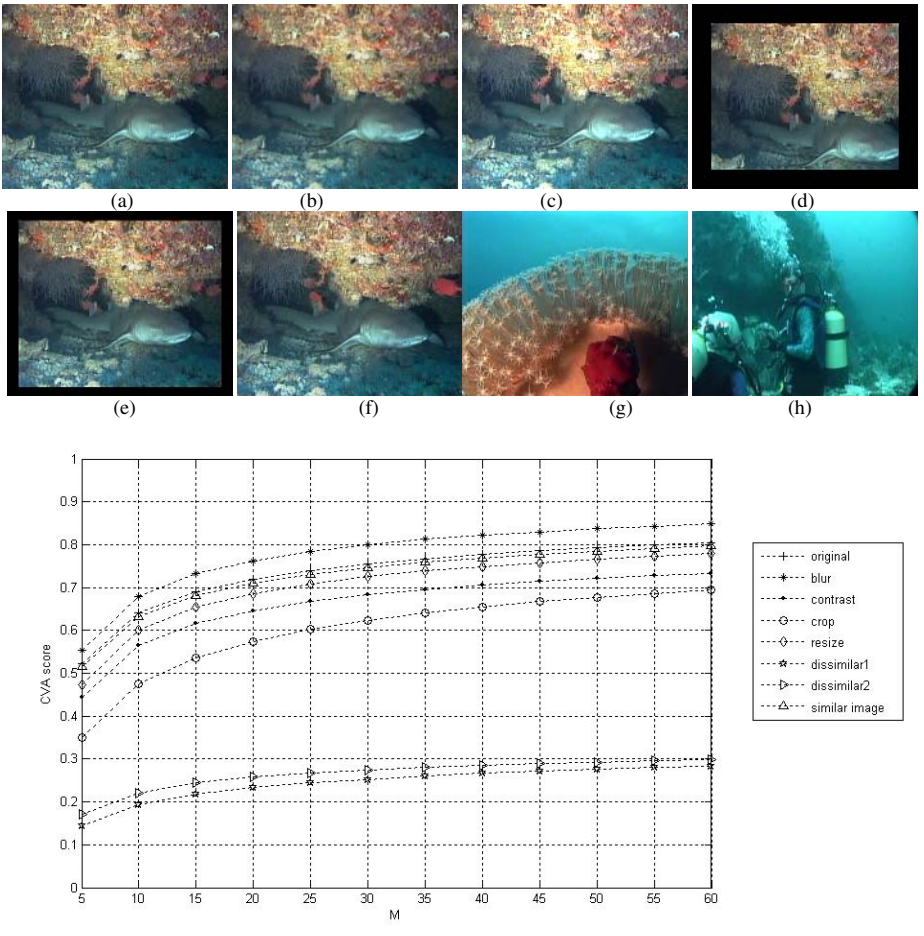


Fig. 3. CVA scores against computation (M). (a) original image (b) image with Gaussian blur (c) image with 25% contrast increase (d) cropped image (e) resize down to 80% of original image (f) similar image ahead of original frame by 70 frame distance (g) and (h) dissimilar images taking from the same video.

The approach is further illustrated in Fig. 5 where image (a) has been pasted into another image giving a composite version (b). Image (c) shows the fork pixel locations where matching has taken place during the computation of the similarity score between images (a) and (b). This indicates that the mechanism is potentially able to detect sub-images with application to copy detection.

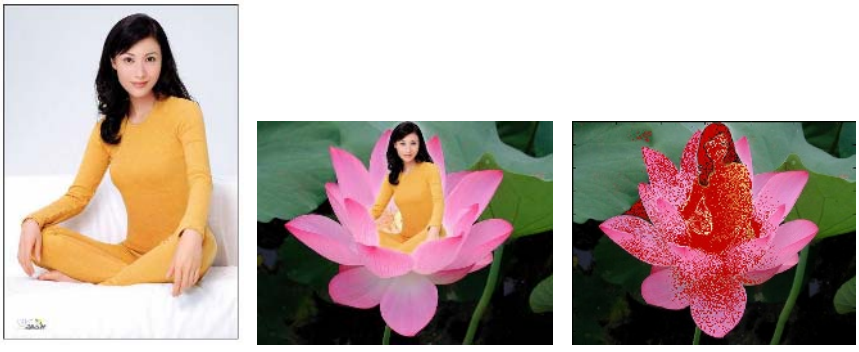


Fig. 4. (a) Image, (b) composite,(c) matching fork pixel locations

4 Conclusions

This paper has shown that a new similarity measure that is not based on pre-selected feature measurements can be used to obtain promising retrieval performance. The similarity is determined by the amount of matching structure detected in pairs of images. Such structure that is found to be in common between specific pairs of images may not be present elsewhere in the database and would be unlikely to be taken into account by a fixed set of features applied universally. The work also provides evidence in support of a mechanism that encompasses notions of both visual attention and similarity.

More results are needed to obtain statistical significance in the precision and recall performances.

Acknowledgement

This research has been conducted within the framework of the European Commission funded Network of Excellence "Multimedia Understanding through Semantics, Computation and Learning" (MUSCLE) [10].

References

[1] Zhang, D., G. Lu, G.: Review of shape representation and description techniques, Pattern Recognition, 37 (2004) 1-19

[2] Fu, H., Chi, Z., Feng, D.: Attention-driven image interpretation with application to image retrieval, Pattern Recognition, 39, no. 7 (2006)

[3] Itti, L.: Automatic foveation for video compression using a neurobiological model of visual attention, IEEE Trans. on Image Processing, 13(10) (2004) 1304-1318

[4] Treisman, A.: Preattentive processing in vision. In: Pylyshyn, Z. (ed.): Computational Processes in Human Vision: an Interdisciplinary Perspective, Ablex Publishing Corp., Norwood, New Jersey, (1988).

[5] Tsotsos, J.K., Culhane, S.M., Wai, W.Y.K., Lai, Y., Davis, N., Nuflo, F.: Modeling visual attention via selective tuning, Artificial Intelligence, 78 (1995) 507-545
[6] Stentiford, F.W.M.: An attention based similarity measure with application to content based information retrieval. In Yeung, M.M., Lienhart, R.W., Li, C-S.(eds.): Storage and Retrieval for Media Databases, Proc SPIE Vol. 5021 (2003) 221-232
[7] Stentiford, F.W.M.: Attention based similarity, Pattern Recognition, in press, 2006.
[8] Desimone, R.: Visual attention mediated by biased competition in extrastriate visual cortex, Phil. Trans. R. Soc. Lond. B, 353 (1998) 1245 – 1255
[9] http://creativearchive.bbc.co.uk/
[10] Multimedia Understanding through Semantics, Computation and Learning, EC 6th Framework Programme. FP6-507752. (2005) http://www.muscle-noe.org/

Learning by Integrating Information Within and Across Fixations

Predrag Neskovic, Liang Wu, and Leon N Cooper

Institute for Brain and Neural Systems and Department of Physics
Brown University, Providence, RI 02912, USA

Abstract. In this work we introduce a Bayesian Integrate And Shift (BIAS) model for learning object categories. The model is biologically inspired and uses Bayesian inference to integrate information within and across fixations. In our model, an object is represented as a collection of features arranged at specific locations with respect to the location of the fixation point. Even though the number of feature detectors that we use is large, we show that learning does not require a large amount of training data due to the fact that between an object and features we introduce an intermediate representation, object views, and thus reduce the dependence among the feature detectors. We tested the system on four object categories and demonstrated that it can learn a new category from only a few training examples.

1 Introduction

In this work we introduce a Bayesian Integrate And Shift (BIAS) model for learning object categories. The model is biologically inspired and uses Bayesian approach to: a) integrate information from different local regions of the scene, given a fixation point, and b) integrate information from different fixations.

Our model falls into a category of feature-based approaches [2,3,5,12]. More specifically, we represent an object as a collection of features that are arranged at specific locations with respect to the location of the fixation point. Even though the number of feature detectors that we use is large, we show that learning does not require a large amount of training data. This is due to the fact that between an object and features we introduce an intermediate representation, object views, and thus reduce the dependence among the feature detectors. In order to learn object views, the system utilizes experience from a teacher. Although this paradigm at first appears more user intensive than paradigms that provide only class information to the system, it is actually very fast since the system can learn object categories using only few training examples.

Feature-based models have become increasingly popular within the computer vision community [5,12,13]. They have been successfully used in various applications such as face recognition [11,14], car detection [1,11], and handwriting recognition [7]. Recently, two groups [3,12] have proposed models that can learn new object categories using only a few training examples. Both models use highly informative and complex features that have to be learned. While the approach

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 488–497, 2006.

proposed by Fei-Fei *et al.* introduces a new learning model based on Bayesian inference, the approach of Serre *et al.* uses standard classification approaches (SVM and gentleBoost) and derives strength from a novel set of features. Both models achieve very good performance although each model handles the number of parts differently. Whereas the system of Serre *et al.* is not affected by the number of features, and can easily use several hundreds of them, the system of Fei-Fei *et al.* can utilize only a small number of parts (e.g. up to seven) due to computational complexity that grows exponentially with the number of parts. In contrast to these two approaches, our model uses simple feature that do not have to be learned. Furthermore, unlike the model of Fei-Fei *et al.*, our system can use an arbitrarily large number of features without an increase in computational complexity.

Biologically inspired models [4,10], and models of biological vision [6,9] have been much less successful (in terms of real-world applications) compared to computer vision approaches. A model that captures some properties of human saccadic behavior and represents an object as a fixed sequence of fixations has been proposed by Keller *et al.* [4]. Similarly, Rybak *et al.* [10] presented a model that is inspired by the scanpath theory [8]. Although these models utilize many behavioral, psychological, and anatomical concepts such as separate processing and representation of "what" (object features) and "where" (spatial features: elementary eye movements) information, they still assume that an object is represented as a sequence of eye movements. In contrast to these approaches, our model does not assume any specific sequence of saccades and therefore is more general.

2 The Model

The motivation for designing the proposed model comes from human perception and the role played by saccadic eye movements during perception. The first observation is that whenever we look at an object, it is always from the point of view of a specific location that serves as the fixation point. Therefore, if an object is captured through an array of feature detectors, then each fixation elicits a different profile of activations of the feature detectors. We will call a configuration consisting of the outputs of feature detectors associated with a specific fixation point a *view*. The fixation point associated with a specific view is called the view center. The fact that any point within an object can be chosen as a fixation point means that the number of views can be very large even for objects of small sizes. In order to reduce the number of views, we will assume that some views are sufficiently similar to one another so that they can be clustered into the same view. The second observation is that all saccadic explorations that occurred prior to the current fixation influence the perception. In our model, the location of each fixation is labeled as representing a center of the specific view and this information about the locations of the centers of the previous views is supplied to the recognition system.

Distribution of the Receptive Fields (RFs). Let us assume that we are given an array of feature detectors whose RFs form a fixed grid and completely cover an input image. One RF has a special role during the recognition process. We call it the central RF and it is always positioned over the fixation point. Since the location of each feature is measured with respect to the central RF, the uncertainty associated with feature's position increases with its distance from the fixation point. In order to capture variations in feature locations, the sizes of the RFs in our model increase with their distance from the central RF. Similarly, the overlap among the RFs increases with their distance from the central location.

Notations. With symbol H we denote a random variable with values $H = (n, i)$ where n goes through all possible object classes and i goes through all possible views within the object. Instead of (n, i), we use the symbol H_i^n to denote the hypothesis that the outputs of all the feature detectors represent the i^{th} view of an object of the n^{th} (object) class. The background class, by definition, has only one view. With variable $\boldsymbol{y}$ we measure the distances of the centers of the RFs from the fixation point. The symbol D_k^r denotes a random variable that takes values from a feature detector that is positioned within the RF centered at $\boldsymbol{y}_k$ from the central location, and is selective to the feature of the r^{th} (feature) class, $D_k^r = d^r(\boldsymbol{y}_k)$. The symbol A_t denotes the outputs of all the feature detectors for a given fixation point $\boldsymbol{x}_t$ at time t. With variable $\boldsymbol{z}$ we measure the distances of the previous fixation locations (view centers) with respect to the location of the current fixation point. For example, the symbol z_{t-1}^j denotes the location of the center of the j^{th} view at time $t-1$. The collection of the locations of all the view centers, up to time t, we denote with the symbol B_t. In order to relate positions of fixation locations with respect to one another, their positions are recorded and indexed with time variable, $\boldsymbol{x}_t$.

What we want to calculate is how information coming from different feature detectors as well as information from previous fixations (the centers of the previous views) influence our hypothesis, $p(H_i^n|A_t, B_t)$. In order to gain a better insight into dependence of these influences, we will start by including the evidence coming from one feature detector and then increase the number of feature detectors and fixation locations.

Combining information within a fixation. Let us now assume that for a given fixation point $\boldsymbol{x}_0$, the feature of the r^{th} class is detected with confidence $d^r(\boldsymbol{y}_k)$ within the RF centered at $\boldsymbol{y}_k$. The influence of this information on our hypothesis, H_i^n, can be calculated using the Bayesian rule as

$$p(H_i^n|d^r(\boldsymbol{y}_k), \boldsymbol{x}_0) = \frac{p(d^r(\boldsymbol{y}_k)|H_i^n, \boldsymbol{x}_0)p(H_i^n|\boldsymbol{x}_0)}{p(d^r(\boldsymbol{y}_k)|\boldsymbol{x}_0)}, \tag{1}$$

where the normalization term indicates how likely it is that the same output of the feature detector can be obtained (or "generated") under any hypothesis, $p(d^r(\boldsymbol{y}_k)|\boldsymbol{x}_0) = \sum_{n,i} p(d^r(\boldsymbol{y}_k)|H_i^n, \boldsymbol{x}_0)p(H_i^n|\boldsymbol{x}_0)$.

We will now assume that a feature detector representing the feature of the p^{th} class and positioned within the RF centered at $\boldsymbol{y}_q$ has a value $d^p(\boldsymbol{y}_q)$. The influence of this new evidence on the hypothesis can be written as

$$p(H_i^n|d^p(\boldsymbol{y}_q), d^r(\boldsymbol{y}_k), \boldsymbol{x}_0) = \frac{p(d^p(\boldsymbol{y}_q)|d^r(\boldsymbol{y}_k), H_i^n, \boldsymbol{x}_0)p(H_i^n|d^r(\boldsymbol{y}_k), \boldsymbol{x}_0)}{p(d^p(\boldsymbol{y}_q)|d^r(\boldsymbol{y}_k), \boldsymbol{x}_0)}. \quad (2)$$

The main question is how to calculate the likelihood $p(d^p(\boldsymbol{y}_q)|d^r(\boldsymbol{y}_k), H_i^n, \boldsymbol{x}_0)$? In principle, if the pattern does not represent any object but just a random background image the outputs of the feature detectors $d^p(\boldsymbol{y}_q)$ and $d^r(\boldsymbol{y}_k)$ are independent of each other. If, on the other hand, the pattern represents a specific object, say an object of the n^{th} class, then the local regions of the pattern within the detectors RFs, and therefore the features that capture the properties of those regions, are not independent from each other, $p(d^p(\boldsymbol{y}_q)|d^r(\boldsymbol{y}_k), H^n, \boldsymbol{x}_0) \neq p(d^p(\boldsymbol{y}_q)|H^n, \boldsymbol{x}_0)$. However, once we introduce a hypothesis of a specific view, the features become much less dependent on one another. This is because the hypothesis H_i^n is much more restrictive and at the same time more informative than the hypothesis about only the object class, H^n. Given the hypothesis H^n, each feature depends both on the locations of other features and the confidences with which they are detected (outputs of feature detectors). The hypothesis H_i^n significantly reduces the dependence on the locations of other features since it provides information about the location of each feature *within* the object up to the uncertainty given by the size of the feature's RF.

The likelihood term, under the independence assumption, can therefore be written as $p(d^p(\boldsymbol{y}_q)|d^r(\boldsymbol{y}_k), H_i^n, \boldsymbol{x}_0) = p(d^p(\boldsymbol{y}_q)|H_i^n, \boldsymbol{x}_0)$. Note that this property is very important from the computational point of view and allows for a very fast training procedure. The dependence of the hypothesis on the collection of outputs of feature detectors A_0 can be written as

$$p(H_i^n|, A_0, \boldsymbol{x}_0) = \frac{\prod_{r,k\in A} p(d^r(\boldsymbol{y}_k)|H_i^n, \boldsymbol{x}_0)p(H_i^n|\boldsymbol{x}_0)}{\sum_{n,i}\prod_{r,k\in A} p(d^r(\boldsymbol{y}_k)|H_i^n, \boldsymbol{x}_0)p(H_i^n|\boldsymbol{x}_0)}, \quad (3)$$

where r, k goes over all possible feature detector outputs contained in the set A_0 and n, i goes over all possible hypotheses.

Combining information across fixations. Let us now calculate how the evidence about the *locations* of different fixations influence the confidence about the specific hypothesis, H_j^n, associated with fixation point $\boldsymbol{x}_t$. We will assume that at time $t-1$ a hypothesis has been made that the fixation centered at $\boldsymbol{x}_{t-1}$ represented the center of the i^{th} view of the object of the n^{th} class. Similarly, we will assume that at time $t-2$ a hypothesis has been made that the fixation centered at $\boldsymbol{x}_{t-2}$ represented the center of the k^{th} view. The location $\boldsymbol{x}_{t-1}$ is measured with respect to some fixed reference point while this same location measured with respect to the position of the current fixation $\boldsymbol{x}_t$ is $\boldsymbol{z}_{t-1}^i$. Similarly, the location of the fixation, $\boldsymbol{x}_{t-2}$, when transformed into the reference frame of the current fixation is $\boldsymbol{z}_{t-2}^k = \boldsymbol{x}_{t-2} - \boldsymbol{x}_t$. We denote with the symbol A_t the outputs of all the feature detectors that are used to calculate the (new)

hypothesis H_j^n. The influence of the evidence about the locations of the previous hypotheses on the current hypothesis can be written as

$$p(H_j^n|\boldsymbol{z}_{t-1}^k, \boldsymbol{z}_{t-2}^i, A_t, \boldsymbol{x}_t) = \frac{p(\boldsymbol{z}_{t-1}^k|H_j^n, \boldsymbol{z}_{t-2}^i, A_t, \boldsymbol{x}_t)p(H_j^n|\boldsymbol{z}_{t-2}^i, A_t, \boldsymbol{x}_t)}{p(\boldsymbol{z}_{t-1}^k|\boldsymbol{z}_{t-2}^i, A_t, \boldsymbol{x}_t)}. \quad (4)$$

The question is now whether the location of any object view depends on locations of other views or just on the location of the current fixation point. Unfortunately, conditioning on the hypothesis in this case does not always reduce the dependence on the locations of other views. However, in order to make the model computationally tractable, we will assume that the view locations are independent from one another given the hypothesis.

Since the location of the k^{th} view of the object does not depend on the configuration of feature detectors that is associated with the current view, and assuming that view locations are independent from one another, the likelihood term from Equation (4) becomes $p(\boldsymbol{z}_{t-1}^k|H_j^n, \boldsymbol{z}_{t-2}^i, A_t, \boldsymbol{x}_t) = p(\boldsymbol{z}_{t-1}^k|H_j^n, \boldsymbol{x}_t)$. The probability that the input pattern represents the j^{th} view of the object of the n^{th} class, given the outputs of the feature detectors A_t and locations of other views, B_t, can be written as

$$p(H_j^n|A_t, \boldsymbol{x}_t, B_t, f(s)) = \frac{\prod_{s<t} p(\boldsymbol{z}_s^{f(s)}|H_j^n, \boldsymbol{x}_t)p(H_j^n|A_t\}, \boldsymbol{x}_t)}{\sum_i \prod_{s<t} p(\boldsymbol{z}_s^{f(s)}|H_j^n, \boldsymbol{x}_t)p(H_j^n|A_t, \boldsymbol{x}_t)}, \quad (5)$$

where i goes through views of the n^{th} object, s goes through the locations of all the fixations and the function $f(s)$ maps a location $\boldsymbol{y}_s$ to a specific hypothesis. With symbol B_t we denoted the set of the locations of all the fixations (object views) with respect to the location of the current fixation, $\boldsymbol{x}_t$. The second term in the numerator is calculated using Equation (3).

3 Implementation

Modeling Likelihoods. We model the likelihoods in Equation (3) using Gaussian distributions. The probability that the output of the feature detector representing the feature of the r^{th} class and positioned within the RF centered at $\boldsymbol{y}_k$ has a value $d^r(\boldsymbol{y}_k)$, given a specific hypothesis and the location of the fixation point, is calculated as

$$p(d^r(\boldsymbol{y}_k)|H_i^n, \boldsymbol{x}_t) = \frac{1}{\sigma_k^r\sqrt{2\pi}} exp \frac{-(\mu_k^r - d^r(\boldsymbol{y}_k))^2}{2(\sigma_k^r)^2}. \quad (6)$$

This notation for the mean and the variance assumes a particular hypothesis so we omitted some indices, $\sigma_k^r = \sigma_k^r(n, i)$. The values for the mean and variance are calculated in the batch mode but, as we will see in the next section, only a small number of instances are used for training so the memory requirement is minimal. For modeling the location likelihoods in Equation (5) we use the multivariate Gaussian distributions since in this case the mean location is a

vector and similarly the variance is a covariance matrix. Note also the difference in measuring the location of the center of a specific RF, $\boldsymbol{y}_k$, and in measuring the location of the fixation point $\boldsymbol{z}^k$. Although both distances are calculated with respect to the same reference point (the fixation point) the centers of the RFs form a fixed grid while the locations of fixation points can vary continuously.

Feature Detectors and Receptive Fields. In this work we extract features using a collection of Gabor filters of four orientations. The orientations and bandwidths of the filters are set to: $\theta = \{0, \pi/4, \pi/2, 3\pi/4\}$ and $\sigma = \{2, 4, 6, 8\}$. Each RF has a square form and the size of the smallest RF is 31x31 pixels. The RFs are arranged along 8 directions and the sizes of the RFs are increased at the ratio of 1.4 (controlled by the enlarge parameter). For example, the sizes of the RFs that are nearest neighbors to the central RF are (31x1.4)x(31x1.4). The overlap between two neighboring receptive fields is 50% meaning that for two neighboring RFs, the larger RF covers 50% of the area of the smaller receptive field. The recognition results are not very sensitive to the small changes in the overlap, enlarge parameter, and the sizes of the receptive fields.

With each RF we associate 16 feature detectors where each feature detector signals the presence of a feature (i.e. a Gabor filter of specific orientation and size) to which it is selective no matter where the feature is within its receptive field. One way to implement this functionality is to use a max operator. The processing is done in the following way. On each region of the image, covered by a specific RF, we apply a collection of 16 Gabor filters (4 orientations and 4 sizes) and obtain 16 maps. Each map is then supplied to a corresponding feature detector and the feature detector then finds a maximum over all possible locations. As a result, each feature detector finds the strongest feature (to which it is selective) within its RF but does not provide any information about the location of that feature.

The Training Procedure. The training is done in a supervised way. We constructed an interactive environment that allows the user to mark a section of an object and label it as a fixation region associated with a specific view. Therefore, every point within this region can serve as the view center. Once the user marks a specific region, the system fixates on the points within it and calculates the mean and variance for each Gaussian. Since the number of training examples is small the training is very fast.

Note that during the training procedure the input to the system is the whole image and the system learns to discriminate between an object and the background. It is important to stress that the system does not learn parts of the object, but the whole object from the perspective of the specific fixation point.

4 Results

We tested the performance of our system on four object categories (faces, cars, airplanes and motorcycles) using the Caltech database as in [3,12]. For illustrative purposes, we choose a face category to present some of the properties of our system in more detail.

The system was first trained on background images in order to learn the "background" hypothesis. We used 20 random images and within each image the system made fixations at 100 random locations. The system was then trained on specific views of specific objects. For example, in training the system to learn the face from the perspective of the right eye, the user marks with the cursor the region around the right eye and the system then makes random fixations within this region and learns the parameters of the Gaussians. During the testing phase, the system makes random fixations (but this time over the whole image) and for each fixation point calculates the probability that the configuration of the outputs of feature detectors represents a face from the perspective of the right eye. To make sure that among the random fixations are also positive examples, each testing image is divided into the view center region(s) (in this case the right eye region) and the rest of the image represents the "background" class. Therefore, positive examples consisted of random fixation within the region of the right eye and negative examples consisted of random fixations outside the region of the right eye. The system was tested on people that were not used for training. We used 200 positive examples and 1000 negative examples for testing.

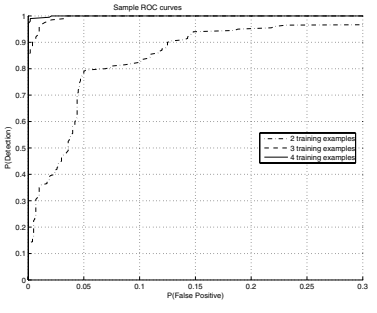


Fig. 1. ROC curves (for the right eye view) for different numbers of training examples and using 10 sampling points per view

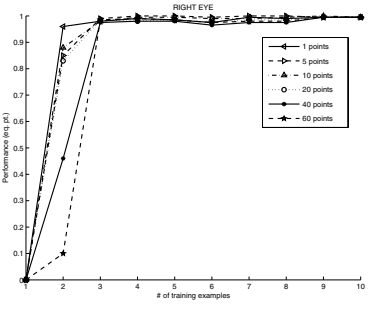


Fig. 2. Performance (for the right eye view) as a function of the number of training examples and sampling points

As a measure of performance we use both the Receiver Operator Characteristic (ROC), Figure 1, and the error rate at equilibrium point (EP), Figure 2, which means that the threshold is set so that the miss rate is equal to the false positive rate. However, since much more information can be represented in one graph using the EP measure compared to the ROC measure, we choose the latter to present most of our results.

As illustrated in Figure 2, learning depends both on the number of training examples and on the number of sampling points that is used in order to learn the view. Since the system was not able to learn much from one example, we set the performance to zero for one training example. In order for the system to learn a face (and "discard" information from the background) it has to be presented with more than one face. As it turns out, two examples are not quite

enough, as can be clearly seen in Figure 1, but with three examples the system can learn the face category (the specific view of the face) with high confidence.

One can see that using more sampling points is not necessarily better especially if the number of training examples is small. This is to be expected since the system becomes biased to the training examples(s). On the other hand, using just a single fixation per view is not sufficient if the number of examples is small. For learning a view using one fixation and only one training face we set the variance by hand and in Figure 2 this number just happened to be a good guess. In all of the experiments that follow, we set the number of fixations per view (the number of sampling points) to 10. The performance of the system using different views of a face is illustrated in Figure 3. It is clear that the easiest views are those centered at the right and the left eye while the tip of the nose

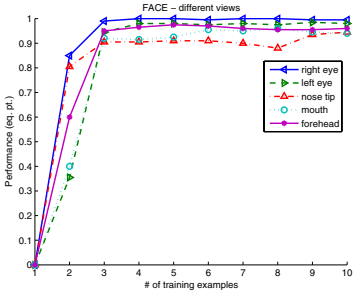


Fig. 3. Performance comparison for different views

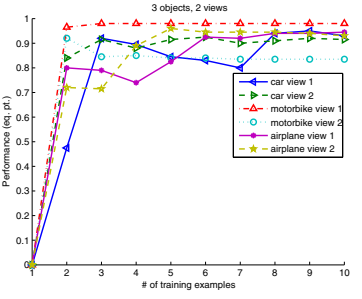


Fig. 4. Performance for three different object classes using two views per object

Fig. 5. First two columns: Examples of view regions as selected by the teacher. Last column: Large view regions (top) and corresponding performance (bottom).

required more training examples. Most errors occurred for the fixation points around the boundaries of the view center regions which is to be expected.

The dependence of the performance on the size of the region that is selected to represent the view center is illustrated in Figure 5 (last column, bottom). The examples of the selected regions are shown in Figure 5 (last column, top). Clearly, learning a view by fixating on a large uniform region, such as the right cheek is more difficult than learning a view using the smaller fixation region such as the nose tip. In Figure 4 we show that the system can easily learn classes other than faces. For each class we used two views as illustrated in Figure 5 (first two columns).

Although the performance of the system is very good using only a single view, we tested whether and how much information from other fixations improve the performance. The tests were done on faces and cars and we used 4 views per class. During the training phase, the user marks the fixation (view) regions and the system then calculates the location likelihoods for each pair of regions separately by randomly selecting points from each region. During the testing phase, in order to estimate the location of the view center, the system selects 10 points with the highest probabilities (as representing the view center) and takes the average over their locations. As expected, the recognition rates consistently increased with the number of views and for some views the error rate was decreased by more than 15%. Due to the lack of space, detailed results will be presented elsewhere.

5 Summary and Future Work

In this work we introduced a Bayesian Integrate And Shift (BIAS) model for learning object categories. The model is biologically inspired and uses Bayesian inference to integrate information within and across fixations. One of the main contributions of the paper is that we introduce a new representation of an object in terms of a finite number of object views. The strength of this representation comes from the fact that once the outputs of the feature detectors are conditioned on a specific object view, their dependence on one another is significantly reduced. This makes the independence assumption more realistic and allows that the parameters of each feature detector can be learned independently. The price that has to be paid for using this representation is that the labeling of an object is now more detailed and the training procedure appears to be more involved. However, the training is very simple and the involvement of a teacher is minimal. The fact that the number of parts per object is very small combined with the fact that the number of examples necessary to learn a new class is also very small makes the training procedure very fast.

We tested the algorithm on various objects and demonstrated that it can learn new categories from only a few examples. Moreover, it can do so using only a single view. Although the focus of this paper is on learning and not on recognition aspects of our system, we also demonstrated that the system achieves very high recognition rates, comparable to those presented in [3] and [12]. However, in contrast to these two approaches, our model uses much simpler features and

does not require a feature learning stage. Some of the additional advantages of our system are: a) it utilizes information from the whole image as opposed to several local regions (as in [3]) which makes it more robust to missing features and occlusions, and b) it is hierarchical in the sense that it can progressively improve recognition by adding information from new fixations.

Among the properties that would improve our system are scale-invariant recognition, and an efficient search algorithm for automatically localizing view center regions within the image. We are currently extending our system to include above properties as well as conducting more comprehensive test to evaluate the robustness to varying lighting conditions and occlusions.

Acknowledgments. This work is supported in part by the ARO under contract W911NF-04-1-0357.

References

1. S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *PAMI*, 26(11):1475–1490, 2004.
2. I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.
3. L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised learning of object categories. In *Proc. ICCV*, 2003.
4. J. Keller, S. Rogers, M. Kabrisky, and M. Oxley. Object recognition based on human saccadic behaviour. *Pattern Analysis and Applications*, 2:251–263, 1999.
5. D. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999.
6. B. Mel. Seemore: Combining color, shape and texture histogramming in a neurally-inspired approach to visual object recognition. *Neural Comp.*, 9(4):777–804, 1997.
7. P. Neskovic, P. Davis, and L. Cooper. Interactive parts model: an application to recognition of on-line cursive script. In *Proc. NIPS*, 2000.
8. D. Noton and L. Stark. Scanpaths in eye movements during pattern perception. *Science*, 171:308–311, 1971.
9. M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nat. Neurosci.*, 2(11):1019–1025, 1999.
10. I. A. Rybak, V. I. Gusakova, A. Golovan, L. N. Podladchikova, and N. A. Shevtsova. A model of attention-guided visual perception and recognition. *Vision Research*, 38:2387–2400, 1998.
11. H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *Proc. CVPR*, 2000.
12. T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *Proc. CVPR*, 2005.
13. A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *Proc. CVPR,*, 2004.
14. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, 2001.

Feature Conjunctions in Visual Search

Antonio J. Rodríguez-Sánchez, Evgueni Simine, and John K. Tsotsos

Dept. of Computer Science & Engineering, and Centre for Vision Research,
York University, Toronto, Canada
ajrs@cs.yorku.ca, eugene@cs.yorku.ca, tsotsos@yorku.ca

Abstract. Selective Tuning (ST) [1] presents a framework for modeling attention and in this paper we show how it performs in visual search tasks. Two types of tasks are presented, a motion search task and an object search task. Both tasks are successfully tested with different feature and conjunction visual searches.

1 Introduction

Visual attention involves much more than simply the selection of next location to fixate the eyes or camera system, regardless of the fact that the vast majority of all computational approaches to attention focus on this issue exclusively. The breadth of functionality associated with attentional processing can easily be seen in several overviews (e.g., [2][3]). One of the most studied topics and with a very significant literature is that of visual search. Visual search experiments formed the basis and motivation for the earliest of the influential models [4][5]. Yet, no satisfactory explanation of how the network of neurons that comprise the visual cortex performs this task exists. Certainly, no computational explanation or model exists either. The recent models derived from these two classic works are compared to human eye movement tracks - overt attention - as validation; but this is not the same as visual search data which is almost exclusively covert, with no eye movement. That humans are able to attend to different locations in their visual field without eye movements has been known since [6]. Further, eye movements require a shift of visual attention to precede them to their goal ([2] surveys relevant experimental work). Attentional models have matured sufficiently so that this problem can now be confronted. This paper makes several steps towards the development of such an explanation expanding the Selective Tuning model [1][7] and comparing performance with existing visual search psychophysical performance. This is done with simple motion stimuli as well as with simple coloured shape stimuli.

2 Motion Visual Search

Here we present a short description of the Motion Model and explain the main concepts and output conventions in order to be able to explain the experimental results. Mathematical details are omitted since they have been published elsewhere [7].

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 498–507, 2006.

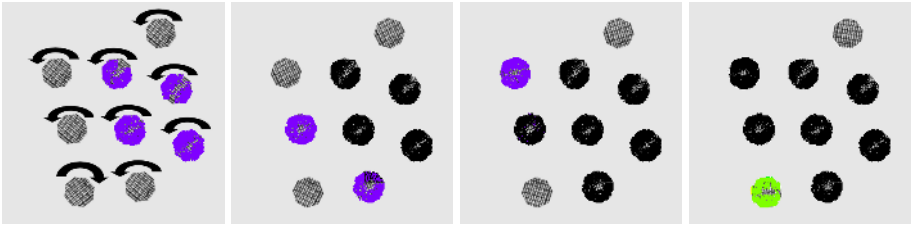
2.1 Description

The Motion Model (MM) is a computational model of attention that works in the motion domain. As input it accepts a video stream in the form of sequences of images and is able to detect, localize and classify moving objects in the scene. The processing of information is inspired by biological research and therefore the computational structure of the model mimics some known properties of the monkey visual pathway. There are four distinct areas of the cortex that are simulated in the model: V1, MT, MST and 7a. All these areas are known to participate in processing of visual information and specifically that which is perceived as motion. The model consists of 690 feature maps each of which encodes the whole visual field in a unique way. Those feature maps are organized into the areas based on their properties and areas are positioned in the form of a pyramid with information flowing from the input to the top of the pyramid and from the top back to the bottom providing feedback.

The internal architecture of the model is rather complicated and full description of it is beyond the scope of this paper (see [7]). However, the aspect that is important to the current discussion is how the model processes complex motion patterns such as rotation, expansion and contraction. Every point in the complex motion pattern moves with a unique velocity (i.e. the direction or the magnitude or both are different for every point). So as complex motion is processed by the model many different feature maps are activated by that motion. For example, the neurons of the area V1 encode only simple linear motion in 12 different directions. Therefore all of V1 will have some activation since there are points moving in each of 12 directions encoded by V1. Further, in MT the moving object is decomposed into regions of common spatial derivatives of local velocity. The full representation of the complex motion is thus the conjunction of different features. Therefore the search for the target that exhibits complex motion among the complex motion distractors can be viewed as a conjunction search and can be expected to produce serial-like performance.In the following example we present the results of the performance of the model in an odd-man-out search of rotating octagons.

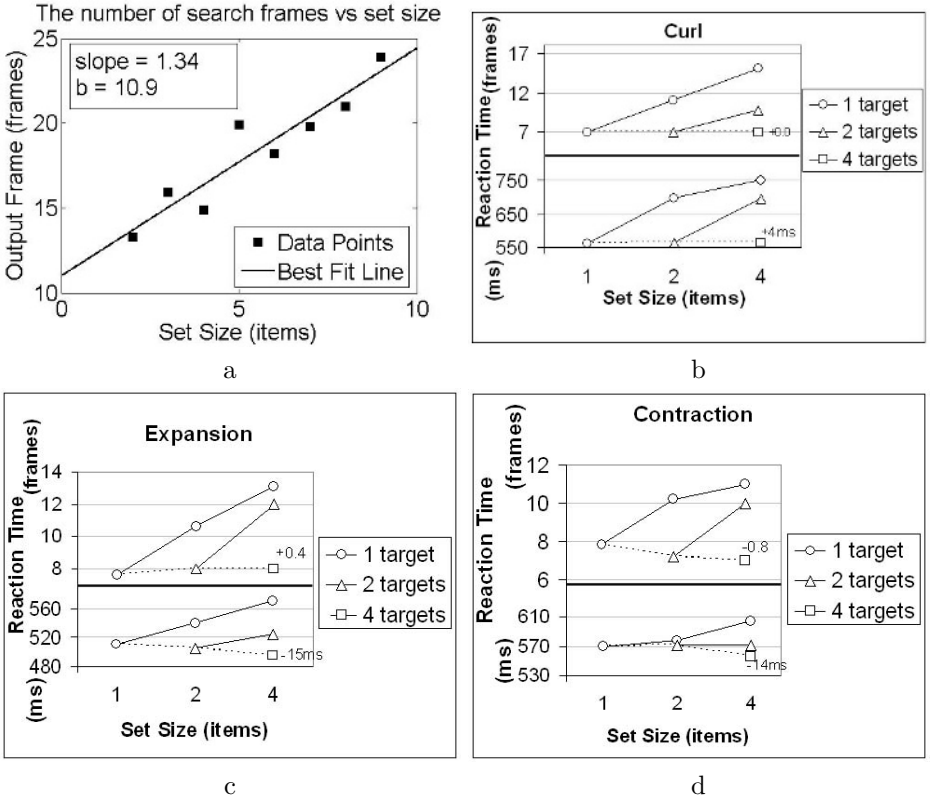
2.2 Motion

Method: To test the performance of the MM we carried out two experiments. First we examined how the model performs a standard visual search task. We used a singleton design where each trial contained only one target and number of distractors was varied from trial to trial. Images of size 445x445 pixels contained one target and from 1 to 8 distractors. A typical input is shown on Fig.1a (the arrows depict the direction of rotation). The target and distractor images were identical textured octagons of 65 pixels in diameter. The target image was rotating counterclockwise and distractors were rotating clockwise both with the angular speed of 3 deg/frame. The target and the distractors were randomly positioned on the white background without overlapping. Fig. 1 b, c and d show the progress of the search. In the second example we compared the performance of the model with the human data by reproducing the experiment described



a b c d

Fig. 1. Typical output of the search task. The most conspicuous locations are attended first a,b,c)the target is not found and the distractors are inhibited to allow for the new location to be examined. d) the search is terminated when the target is found.



a b c d

Fig. 2. Search Results a) standard visual search for the stimulus in 1 b,c,d) the model's performance on the stimuli used in Thornton and Gilden paper. The top half of each graph shows the output of the model and the bottom half of the graph is the data reported by Thornton and Gilden[8].

by Thornton and Gilden [8]. We used similar random noise patches exhibiting complex motion. Instead of measuring the reaction time for finding the target we counted the number of frames processed by the model until the target was localized. The least mean squares method was used to fit the straight line into the set of points.

Results: Fig.2a the results of the first experiment, standard visual search, on the stimulus on Fig.1. Positions of the points on the graph suggest linear dependence. The best fit line has a slope of 1.34 frames/item and intersects with y-axis at 12.3 frames. Fig.2b, c, and d show how the model performs on the stimulus similar to one in the Thornton and Gilden experiment.

Discussion: The above results show the ability of the Motion Model to perform a standard visual search task. The equivalent of reaction time (RT) is expressed in the number of frames needed to find the target. The values appear to be linearly increasing as we increase set size, which seems to be in agreement with psychophysical data [8], [9]. The typical output of the model is shown on the Figure 1. We can see that objects are selected in groups rather than one at the time. This behavior is caused by the fact that the model is attending to the specific motion type at the specific spatial location. The location is defined by the receptive field of the winner neuron at the top of the pyramid. Therefore, every object or part of an object that lies within the attended receptive field and exhibits the attended motion will be selected and processed in parallel. Several other researchers proposed that multiple items can be processed in a single attentional fixation, see review [10].

The model shows similar results in processing complex motion to those reported by Thornton and Gilden. Fig.2b-d shows that the qualitatively performance of the model is similar to human data. The top half of each graph shows the output of the model and the bottom half of the graph is the data reported by Thornton and Gilden[8]. The largest increase in RT is observed for the rotating stimulus. Expansion and contraction graphs are also very similar to psychophysical data. Overall the comparison is qualitatively correct, an encouraging sign for the biological plausibility of the model.

3 Object Visual Search

3.1 Description

The input for the model is a scene with several objects, and the task of the model is to find a particular object whose representation has been learned previously.

The model we propose tries to mimic the human visual pathway for object recognition. It is composed of two hierarchical pyramids with information flowing from the input to the top of the pyramid and from the top back to the bottom providing feedback. One path is for shape processing and the other for color processing. There are four visual areas simulated in the model: LGN, V1, V4 and IT. The model consists of 22 feature maps, each feature map encodes the visual field in a unique and hierarchical way.

Information first flows from the input to area LGN and V1. LGN extracts three color feature maps (red, green and blue). V1 is composed of edge detectors organized in 8 feature planes (each containing neurons tuned to one of 8 directions). Two additional feature maps in V1 compute center-surround color differences from the LGN color feature maps. Information from V1 flows to V4, which comprises 8 feature maps for curvature. Finally, IT neurons encode a representation of the whole object based on curvature and color differences. The shape and color analysis are explained in more detail in the following sections.

Shape analysis. The shape processing pathway (Fig. 3a) is more complex and is inspired by [11]. Visual Area V1 contains neurons that perform edge analysis. Gabor filters [12] are used with 8 different orientations. Size of the neurons is 16×16 pixels. The output of V1 neurons is 8 feature planes, representing edges at 8 orientations. After non-maximal suppression [13], the output from V1 neurons feed into V4 neurons that compute curvature values based on orientation changes in groups of adjacent V1 neurons from the 8 V1 planes. For example, if a V1 neuron in a V4 receptive field had its highest response for $\theta = 0$ and another adjacent one had a high response for $\theta = \frac{\pi}{4}$, we would have a corner. If both orientations were equal, it would correspond to a straight line. Curvature for V4 is then defined as:

$$curv = min(|\theta_1 - \theta_2|, 2\pi - |\theta_1 - \theta_2|); curv \in [0, \pi) \tag{1}$$

Where θ_1 and θ_2 are the orientations of two V1 cells. A value of π can be added to θ_1 and/or θ_2 depending on the neurons' relative positions inside the V4 receptive field due to the fact that the same gabor filter orientation can account for two different angles. The activation value of the V4 neuron is the summed activations from the V1 neurons used to obtain the curvature. V4 neurons receptive field comprise groups of 4×4 V1 neurons.

V4 neurons' output is 8 2D feature maps that encode for the difference of curvature among groups of V1 neurons. This output feeds into IT at the very top of the hierarchy (Fig. 3a). The receptive fields of IT neurons comprise an area of 32×32 V4 neurons (that is, 128×128 pixels). The center of mass is calculated for every group of V4 neurons as the mean of the V4 neuron coordinates where responses are different from zero. Then, at each angular position (in 10 deg bins), its curvature is computed [11], obtaining a histogram-like representation for IT neurons where one axis correspond to the angular position (λ) and the other coordinate is the curvature ($curv$) for that position (Fig. 3a):

$$\lambda = round[\frac{tan^{-1}(\frac{y-centroid_y}{x-centroid_x}) * 18}{\pi}]; IT(\lambda) = curv \tag{2}$$

The term $\frac{18}{\pi}$ is for the angular position to be in 10 deg bins.

All neuron relative sizes were chosen to correspond closely to the neurophysiological measured sizes of [14] considering a distance of 30 cm (usual psychophysical distance) to a 1280×1024 display. Neurons' receptive fields are overlapped.

Bias. An important part of the Selective Tuning Model [1] is top-down bias (Fig. 3a). Given a target stimulus, the features not relevant for target recognition are inhibited by multiplying an inhibitory bias (greater than 0.0 and less than 1.0) and the actual feed-forward response of the neurons. The target representation is obtained from the responses of the IT level neurons on seeing the target stimulus alone in the visual field.

Color analysis. The processing of color follows a centre-surround analysis [15]. A first layer (LGN) extract 3 feature maps for red (R), green (G) and blue (B) responses. In the upper layer (V1), surround values for red-green (RG), green-red (RG), blue-yellow (BY) and yellow-blue (YB) are extracted following most models (e.g [16]). RG feature plane also accounts for GR differences, the same applies to the BY feature plane. Color layers are also biased in a similar way as shape for particular targets.

3.2 Recognition

Finding an object works in two phases: On a first stage, a sample of the object is supplied to the network. The object's representation for color and shape is extracted at every layer. After the object has been learned by the network, the network is able to search for it in test scenes and on presentation of the test stimulus, the processing is biased by the learned object or target. The search begins after an initial feed-forward activation by considering the best matching IT neuron.

To determine how close is the shape to the desired shape, distance to the target IT histogram is computed, for this distance, a measure similar to cummulative distance is used. The number of peaks ($\sharp peaks$) and the difference in peak values will be considered as follows:

$$d = (\sharp peaks_{sample} - \sharp peaks_{candidate})^2 + \sum |peak_{sample}(x) - peak_{candidate}(x)| \tag{3}$$

The activation of the neuron is inversely proportional to d. Both activation values for color and shape $\in [0, 1]$ and the activation of the candidate IT neuron is the addition of both values. Even though the object can be in the receptive field of the highest activated IT neuron, due to its large receptive field and even after the bias, it can accommodate other objects (that may even disturb the firing values of the IT neuron). Information is further filtered in the lower layers (V4, V1) by computing winner-take-all in a hierarchical fashion [1]. The WTA processes in V4 are grouped by curvature angle. There is a separate WTA process for each 10 deg bin (as determined by Eq. 2), i.e., a V4 neuron will only compete with neurons in the same bin. In V1 only those neurons connected with the V4 winners are considered, and the same process is applied when going from V1 to the image, finding the contour of the candidate object. Figure 3b shows an example of this process. Inhibition of return was implemented in by blanking the part of the input image corresponding to the analyzed object.

3.3 Visual Search: Efficient and Inefficient Searches

Recently, it has been shown that conjunction searches (See [10] for a review) may exhibit shallower slopes than those found by [4], and there seems to exist a continuum from efficient to inefficient visual search. An interesting theory is the one proposed by Duncan and Humphreys ([17]), they argue that visual search is influenced by the similarity between target and distractors. Here we will test the model with several experiments concerning this continuum. The sample was given as input in a 128×128 pixel image, and the scenes were 640×640 pixels.

Experiment 1: Color differences

Method: In this experiment we study how the model performs in a color similarity search. We try here to simulate Nagy and Sanchez's experiment [18], who showed that feature search can be inefficient if the differences in color are small. We used the CIE values [18] from their experiments converted to RGB with a fixed luminance (Y) of 0.25. The task is to find the redder circle among 5, 10, 15, 20 and 25 distractors for two conditions: small and large color differences. The target and distractors were randomly positioned on a black background. The least mean squares method was used to fit the straight line into the set of points.

Results: An example is shown in Figure 4a, where, when there are small differences between the target and the distractors, a much larger number of fixations are needed to find the target. Figure 4b shows how the number of fixations increases as the set size increases. This experiment reports similar results to Nagy and Sanchez's where color search is inefficient if color difference is small between target and distractors (slope=0.39) and efficient if the difference is large (slope=0.01).

Experiment 2: Feature, conjuntion and inefficient search

Bichot and Schall showed that monkey visual search reaction times are comparable to human [19] [20], namely they show that the conjunction of two different features (shape and color) is steeper than feature search, but shallower than what was obtained by [4]. They report slopes of 3.9 ms/item. Searching for a rotated T among rotated Ls, [21] reported that this search was quite inefficient (20 msec/item), and less efficient than conjunction searches. To find a T among Ls is more inefficient than a conjunction search, which is less efficient than a simple feature search.

Method: In this experiment we study how the model performs in a simple feature search, a conjunction search and an inefficient search. Conjunction search was similar to that of [19]. The stimuli were crosses and circles, red or green colored. The task was to find a red circle among green circles and red crosses, here we used 8, 12, 16, 18, 22 and 24 distractors. Feature search was a simplification of the previous conjunction search, that is, to look for a circle among crosses. For inefficient search, a rotated T was to be found among Ls rotated at 0, 90 and 180 degrees, in this case we used 6, 9, 12, 15, 18 and 21 distractors. Analysis was the same as for previous experiments.

Results: An example searching for a T among Ls is shown in Figure 4c, many fixations are needed to find the target. Figure 4d shows the number of fixations as the set size increases for the feature search(find a circle among arrows),

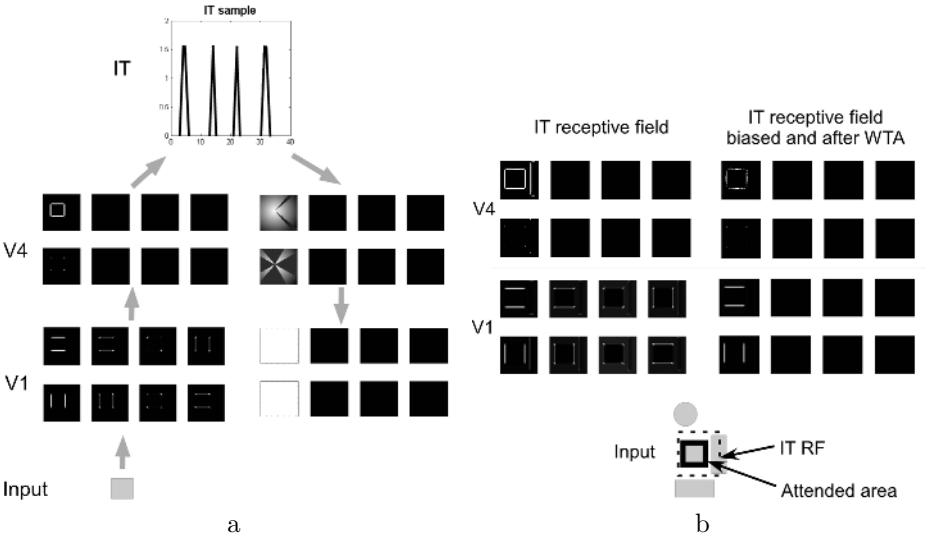


Fig. 3. Shape analysis on target stimulus a) Sample: edges are extracted in V1 at each different orientation, then in V4 curvatures are calculated (bottom-up), finally IT computes the *curvature* × *position* representation [11] and a bias is constructed for V4 and V1 layers (top-down). b) Analysis of a scene: find the square. Lower layers in the hierarchy are first biased and information is later filtered through a winner take all process (See [1] for a full explanation) to attend to the position of the square inside the IT receptive field (RF). Bottom: Scene with a square. Left: IT neuron RF containing the square. Right: IT neuron attending for the square.

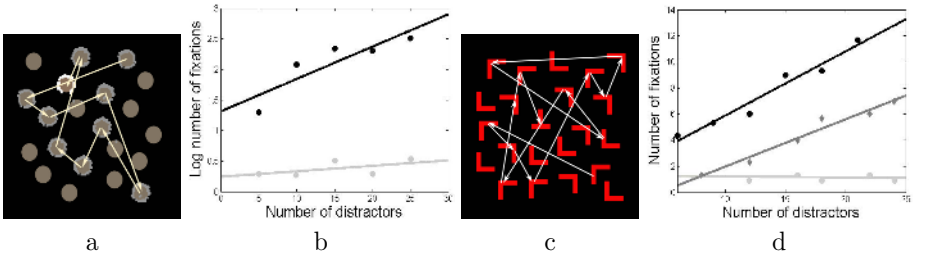


Fig. 4. Visual Search Results a) Example where the target and distractors have small color differences, 10 fixations were needed to find the redder item (white outline). b) The number of fixations as a function of set size. *Gray line: large color difference, black line: small color difference* c) Inefficient search: Find the rotated T among 21 Ls, 14 fixations were needed to find the T. d) The number of fixations as a function of set size for feature search (light gray), conjunction search (gray) and inefficient search (black).

conjunction search (find a red circle among red arrows and green circles) and inefficient search (find a rotated T among Ls). The figure shows how the steepest fitted line is the one corresponding to looking for a T among Ls (inefficient search, slope of 0.49) experiment, followed by conjunction search (slope of 0.36) and feature search is practically flat (slope of 0.00). These results are in accordance with the continuum from efficient to inefficient search psychophysical experiments have shown (see [10] for a review).

Discussion. The above results show the ability of the Object Recognition Model to perform visual search. The reaction time is shown based on the number of fixations. We performed easy feature search, difficult feature search, conjunction search and inefficient search. The obtained results seem to agree with the increasing degrees of difficulty reported by psychophysical data from [18], [19] and [21], whose experiments were simulated above. Our experiments seem to agree also with the proposal that search is more efficient when objects are more dissimilar [17] and the continuum efficient-inefficient search found in the literature [10].

4 Conclusions

Two examples of how the Selective Tuning model can account for the visual search observations of a significant set of psychophysical experiments have been presented. One of the examples dealt with motion patterns while the other with coloured objects. In each case, both feature singleton and feature conjunction image items can be correctly handled. The work is in stark difference to other seemingly related research (such as [16], [22]). Here the performance comparison is not eye movement based as in Itti et al. They model bottom-up saliency and cannot include top-down effects of general knowledge while at the same time use tracking data that is confounded by such knowledge. Riesenhuber and Poggio also model bottom-up recognition with no need for attention and thus have no natural mechanism for serial search through a collection of stimulus items in a display. The contribution in this paper of mechanisms that can provide an explanation for visual search performance has the promise of enhancing performance of recognition algorithms in complex scenes.

References

1. Tsotsos, J., Culhane, S., Wai, W., Lai, Y., Davis, N., Nuflo, F.: Modeling visual attention via selective tuning. Artificial Intelligence **78** (1995) 507–545
2. Pashler, H., ed.: Attention. London UK: University College London Press (1998)
3. Itti, L., Rees, G., Tsotsos, J.: Neurobiology of Attention. Elsevier Science (2005)
4. Treisman, A., Gelade, G.: A feature integration theory of attention. Cognitive Psychology **12** (1980) 97–136
5. Koch, C., Ullman, S.: Shifts in selective visual attention towards the underlying neural circuitry. Human Neurobiology **4** (1985) 219–227
6. Helmholtz, H.: Teatrise on physiological optics (southall, trans. from 3d german edition, 1909). The Optical Society of America (1924)

7. Tsotsos, J., Liu, Y., Martinez-Trujillo, J., Pomplun, M., Simine, E., Zhou, K.: Attending to motion. Computer Vision and Image Understanding **100**(1-2) (2005) 3–40
8. Thornton, T., Gilden, D.: Attentional limitations in the sensing of motion direction. Cognitive Psychology **43** (2001) 23–52
9. Hillstrom, A., Mantis, S.: Visual motion and attentional capture. Perception and Psychophysics **55**(4) (1994) 344–411
10. Wolfe, J.: Visual Search. In: Attention. London UK: University College London Press (1998)
11. Pasupathy, A., Connor, C.: Shape represantion in area v4: Position-specific tuning for boundary conformation. Journal of Neurophysiology **86** (2001) 2505–2519
12. Marcelja, S.: Mathematical description of the responses of simple cortical cells. Journal of Optical Society of America **70** (1980) 1297–1300
13. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **8** (1986) 679–698
14. Felleman, D., Essen, D.V.: Distributed hierarchical processing in the primate cerebral cortex. Cerebral Cortex **1** (1991)
15. Rolls, E., Deco, G.: Computational Neuroscience of Vision. Oxford, New York (2002)
16. Itti, L., Koch, C., Niebur, E.: A model for saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence (1998) 1254–1259
17. Duncan, J., Humphreys, G.: Visual search and stimulus similarity. Psychological Review **96** (1989) 433–458
18. Nagy, A., Sanchez, R.: Critical color differences determined with a visual search task. Journal of the Optical Society of America A **7** (1990) 1209–1217
19. Bichot, N., Schall, J.: Saccade target selection in macaque during feature and conjunction visual search. Visual Neuroscience **16** (1999) 91–89
20. Wolfe, J., Cave, K., Franzel, S.: Guided search: An alternative to the feature integration model for visual search. Journal of Experimental Psychology: Human Perception and Performance **15** (1989) 419–433
21. Egeth, H., Dagenbach, D.: Parallel versus serial processing in visual search: Further evidence from subadditive effects of visual quality. Journal of Experimental Psychology: Human Perception and Performance **17** (1991) 551–560
22. Riesenhuber, M., Poggio, T.: Hierarchical models of object recognition in cortex. Nature Neuroscience **2** (1999) 1019–1025

A Biologically Motivated System for Unconstrained Online Learning of Visual Objects

Heiko Wersing[1], Stephan Kirstein[1], Michael Götting[2], Holger Brandl[1], Mark Dunn[1], Inna Mikhailova[1], Christian Goerick[1], Jochen Steil[2], Helge Ritter[2], and Edgar Körner[1]

[1] Honda Research Institute Europe GmbH,
Carl-Legien-Str. 30, 63073 Offenbach/Main, Germany
[2] Bielefeld University - Neuroinformatics Group, Faculty of Technology
PO Box 100131, D-33501 Bielefeld, Germany

Abstract. We present a biologically motivated system for object recognition that is capable of online learning of several objects based on interaction with a human teacher. The training is unconstrained in the sense that arbitrary objects can be freely presented in front of a stereo camera system and labeled by speech input. The architecture unites biological principles such as appearance-based representation in topographical feature detection hierarchies and context-driven transfer between different levels of object memory. The learning is fully online and thus avoids an artificial separation of the interaction into training and test phases.

1 Introduction

The capacity for learning and robust recognition of numerous objects makes the human visual system superior to all currently existing technical object recognition approaches. One aspect of this is the capability of quickly analyzing and remembering completely unknown new objects. In this contribution we refer to this ability as *online learning*, which is of high relevance for cognitive robotics and computer vision. A typical application domain we are heading for is to increase the knowledge of an assistive robot in a changing and unpredictable environment [1]. The capability of learning online constitutes a fundamental difference to offline learning, since it enables an interactive process between teacher and learner. The immediate feedback about the current learning state can induce an instanteneous and active learning process that reduces the amount of necessary training data and allows an iterative error correction based on user feedback.

To realize such learning, we present a system that combines a flexible neural object recognition architecture with a biologically motivated attention system for gaze control, and a speech understanding and synthesis system for intuitive interaction. The target is to obtain a flexible object representation system that is capable of high-performance appearance-based object recognition of complex objects together with a particularly rapid online learning scheme that can be

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 508–517, 2006.

carried out by cooperative training with a human teacher. A high level of interactivity is achieved by avoiding an artificial separation into training and testing phase, which is still the state-of-the-art for most current trainable object recognition architectures. We do this by using an incremental learning approach that consists of a two-stage memory architecture of a context-dependent working or sensory memory and a persistent object memory that can also be trained online.

The learning is unconstrained in the sense that we do not impose any preconditions on the environment, except that objects are presented to the system by showing them by hand. To allow online learning in this difficult scenario, we use a dynamic segmentation approach that performs a fast figure-ground separation based on an initial stereo-based coarse object hypothesis. The object recognition architecture is motivated from the ventral pathway of the human visual cortex and can be applied to arbitrary complex-shaped objects. Fast online learning can be achieved with this architecture, because object-specific learning occurs only on the highest levels of the hierarchical feature detection stages. The lower stages of the model correspond to earlier and intermediate feature detection stages in the visual cortex and are trained by sparse coding learning rules [2]. This results in a particularly robust appearance-based representation of objects using a consistent library of typical local shape elements.

In the following we review related work in Section 2 and give an overview over our system in Section 3. In Section 4 we describe the components of the visual memory in more detail, show results on the performance and learning behaviour in Section 5 and give a short final discussion in Section 6.

2 Related Work

Compared to the large body of work on offline training of model-free object recognition architectures, only few work has been done on online learning for complex-shaped objects. The main problems are poor generalization due to the inherent high dimensionality of visual stimuli, and the difficulty to achieve incremental online learning with standard classifier architectures like multi layer perceptrons or support vector machines.

To make online learning feasible, the complexity of the sensorial input has been reduced to simple blob-like stimuli [3], for which only positions are tracked. Based on the positions, interactive and online learning of behavior patterns can be performed. A slightly more complex representation was used by Garcia et al. [4], who have applied the coupling of an attention system using features like color, motion, and disparity with a fast learning of visual structure for simple colored geometrical shapes like balls, pyramids, and cubes.

Histogram-based methods are another common approach to tackle the problem of high dimensionality of visual object representations. Steels & Kaplan [5] have studied the dynamics of learning shared object concepts based on color histograms in an interaction scenario with a dog robot. Another model of word acquisition that is based on multidimensional receptive field histograms for shape representation and color histograms was proposed by Roy & Pentland [6]. The

learning proceeds online by using a short-term memory for identifying reoccurring pairs of acoustic and visual sensory data, that are then passed to a long-term representation of extracted audiovisual objects.

Arsenio [7] has investigated a developmental learning approach for humanoid robots based on an interactive object segmentation model that can use both external movements of objects by a human and internally generated movements of objects by a robot manipulator. Using a combination of tracking and segmentation algorithms the system is capable of online learning of a few objects by storing them using a geometric hashing representation.

An interesting approach to supervised online learning for object recognition was proposed by Bekel et al. [8]. Their VPL classifier consists of three major stages. The two feature extraction stages are based on vector quantization and a local PCA measurement. The final stage is a supervised classifier using a local linear map architecture. The image acquisition of new object views is triggered by pointing gestures on a table, and is followed by a short training phase, which take some minutes. The main drawback is the lack of an incremental learning mechanism to avoid the complete retraining of the architecture.

Kirstein et al. [9] have presented an online learning architecture that is operated in a more constrained scenario with defined black background to ease the figure-ground segmentation. Their focus was the transfer from a short-term to more condensed long-term memory representation using incremental vector quantization methods.

3 System Overview

The visual input is a left and right image pair, obtained from a stereo camera head mounted on a pan-tilt unit. The gaze control of the head is driven by an independent circuit that combines the cues of motion, color, and depth for attention-driven selection of the gaze direction. We use the concept of peripersonal space [10] to establish shared attention on a presented object during learning. This means that the system will focus its attention on an object that is presented within a particular short-distance range interval that roughly corresponds to the biological concept of the manipulation space around the body. If nothing is present within this space, the cues of motion and color/intensity determine the gaze selection of the system (see [10] for more details).

The online learning system is working with the camera output that is generated according to the gaze selection of the independent attention system. Based on the current stereo view pair, a depth map is computed that is aligned with the left camera image. The left camera image and the depth map are passed to the peripersonal blob detection stage that generates a square region of interest (ROI), based on the estimated distance of the current object hypothesis. By estimating the distance, the apparent size of objects within the ROI can be normalized with remaining uncertainties due to the limited precision of the depth computation. The square ROI with distance dependent size in the original image is scaled to a normalized size of 144x144 pixels.

The normalized ROI around the object hypothesis together with the corresponding part of the depth map is passed to the figure-ground segmentation stage of processing, the adaptive scene-dependent filters (ASDF) [11]. The ASDF method makes no strong assumptions on the objects like e.g. being single-colored. Based on the depth map, a relevance map is obtained that covers the object only coarsely with considerable overlap to the background. For each pixel location in the ROI, a local feature vector is computed based on RGB color channels, depth, and pixel position. Using a dynamic vector quantization model first an unsupervised segmentation is computed using the local feature vectors in the ROI as input ensemble and then the input image is segmented according to the mapping to the Voronoi cells of the found vector quantization centers. Due to a sufficient number of centers, we obtain an oversegmentation and can then select object segments as those that are sufficiently contained within the relevance map (see [11] for more details). The method obtains an intrinsic stability by continuously iterating the vector quantization based on state on the previous frame. We additionally use skin color detection to remove parts of the hand that hold the object. The output of the ASDF stage is a mask describing the current figure-ground hypothesis on the ROI.

The selected ROI and the segmentation mask from the ASDF stage are fed into the model of the ventral visual visual pathway of Wersing & Körner [2] to obtain a complex feature map representation that is based on 50 shape and 3 color feature maps. The color channels are just downsampled images in the three RGB channels. The output is a high-dimensional view-based representation of the input object, that is then passed to the further object memory representation stages for learning and recognition.

To allow a particularly interactive online learning we use a memory concept that is separated into a sensory memory carrying the currently attended object and a persistent memory that carries consolidated and consistently labeled object view representations. As long as an object is presented within the peripersonal space and has not been labeled or confirmed, the obtained feature map representations of views are stored incrementally within the sensory memory. At the same time, all newly appearing views are being classified using the persistent object memory. If the human teacher remains silent, then the system will either generate a class hypothesis, or reject the presented object as unknown and verbalize this using the speech output module. The human teacher can confirm the hypothesis or make a new suggestion on the correct object label. As soon as feedback by the teacher is available, the learning architecture starts the concurrent transfer from the sensory memory buffer into the consolidated object memory. This extends over the whole history of collected views during the presentation phase and also proceeds with all future views, as long as the object is still present in the peripersonal space. The labeling of the current object can be done by the teacher at any time during the dialogue and is not restricted to being a reaction on a class hypothesis of the recognition system. The concept of a context-dependent memory buffer makes a separation into training and testing phases unnecessary. The transfer from the sensory to the object memory is

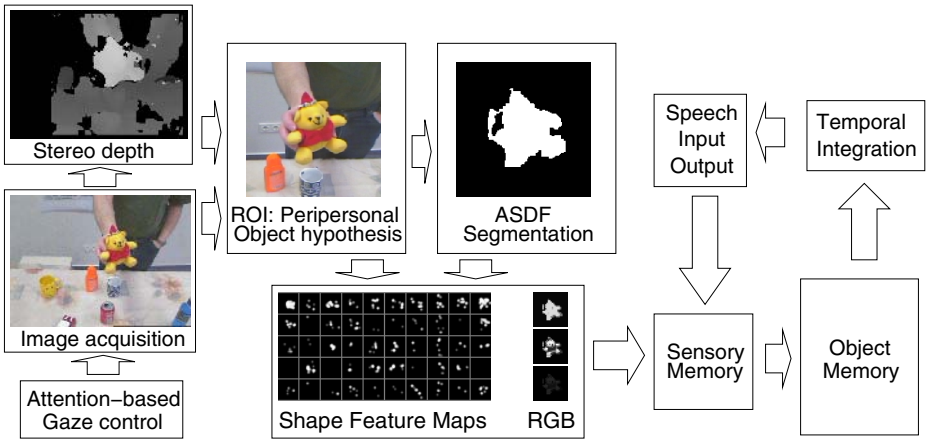


Fig. 1. Overview over the visual online learning architecture. See text for explanation.

sufficiently fast to remain unnoticed to the human trainer and the learning success can be immediately tested, allowing for a real online learning interaction.

The speech input and output is very important for the intuitive training interaction with the system. We use a system with a headset, which is the current state-of-the-art for speaker-independent recognition. The vocabulary of object classes is specified beforehand, to be able to label arbitrary objects we also use wildcard labels such as "object one", "object two" etc.

4 Object Memory Representation

In the following we describe in more detail the main components of the object memory and recognition system. For a more detailed description of the attention, gaze selection and stereo processing system we refer the reader to [10].

4.1 Hierarchical Feature Processing

The output of the ASDF figure-ground segmentation stage is a mask signal that is combined with the candidate ROI (of size 144x144 pixels) and fed into the hierarchical model of the ventral visual pathway developed by Wersing & Körner [2]. To obtain invariance against rotations in the image plane, which is normally quite a challenge for appearance-based recognition, we determine the principal axes of the figure-ground mask and rotate the ROI and mask aligned with the horizontal direction. This normalization introduces much better robustness for the recognition of elongated objects like e.g. bottles.

The rotation-normalized ROI is processed using a hierarchy of feature detection and pooling stages that achieves a robust appearance-based representation of an object view as a collection of several sparsely activated feature map representations (see Fig. 1). In the system that we consider here, we use 50 shape

features, that are sensitive to particular local structural elements in the image, and the three RGB channels. The 50 shape feature maps are represented at a resolution of 18x18, due to the spatial convergence in the hierarchy. As was shown before, the output of the feature representation of the complex feature layer can be used for robust object recognition that is competitive with other state-of-the-art models, when offline training is being used [2].

The efficiency of the representation is achieved by sparse coding that ensures that object views are represented using only sparse activation in the high-dimensional feature space. To represent also coarse color information, the 3 RGB channels are used as a downsampled ROI at the same resolution of 18x18 as the shape features. Although the complete dimensionality of a single view representation is thus (50+3)x18x18=17172, the effective dimensionality is much smaller, due to the sparsity of the representation vector and the restriction of activity around the figure-ground mask. Nevertheless it is a key feature of our biologically motivated visual processing model that robustness, generalization and speed of learning is not achieved by a dimension reduction as in most other current online learning models [3,4,5,6,7,8]. The key element is a transformation of the input into a sparse robust feature map representation that captures locally invariant relevant structures of the objects.

4.2 Sensory and Object Memory

The object representation system for online learning and recognition is separated into two subsystems: A sensory memory for temporarily remembering the currently attend object within focus and a persistent object memory that integrates all object knowledge incrementally over time.

The high-dimensional output vectors of the feature hierarchy are continuously stored within the sensory memory. The task of this memory is to capture all current views of an object to be able to use them for transfer to the object memory when a speech label has been given. This means that also those views can be used for training that were recorded before a labeling of the object was obtained from the human trainer, relaxing the constraints on the training dialogue. The sensory memory is realized as an incremental vector quantization model, where new representatives are added, when they are sufficiently dissimilar to all current entries in the sensory memory. The similarity is measured based on Euclidean distance in the feature map vector space. Due to the sparsity of the feature map vectors this similarity computation can be very efficiently implemented [9].

When a labeling signal arrives, because the human teacher has labeled an object or has confirmed a hypothesis generated from the object memory, the information accumulated in the sensory memory is transferred to the object memory in real time. Here we use the same incremental vector quantization model. If there are already some views available in the object memory, the comparison is performed against the already existing representation. The main advantage of the template-based representation is that training is fully incremental and

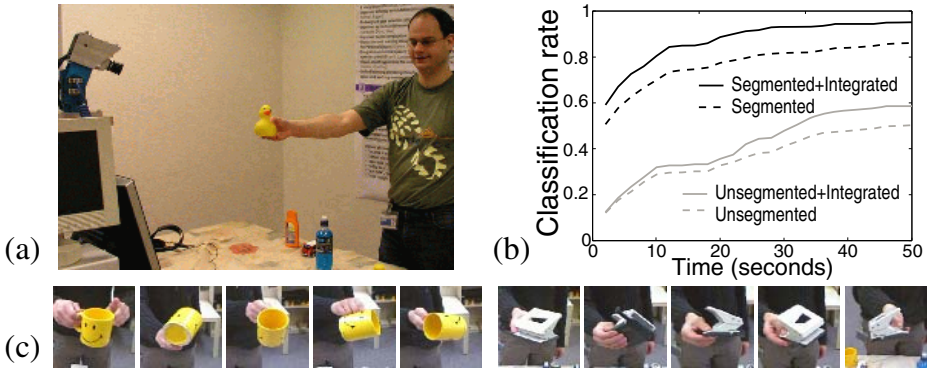


Fig. 2. Presentation scenario for our online learning architecture (a), and average recognition performance versus training time (b) for training the 10th object after 9 were already trained, with and without segmentation and temporal integration.(c) demonstrates the typical rotation variation that is applied during all experiments.

non-destructive with regard to previous information. This representation can be later condensed and consolidated using additional learning mechanisms that operate on a slower time scale [9].

Every arriving view is being classified based on the information in the object memory using a nearest-neighbour classifier for the labeled representatives. Since the system is running at a sufficient frame rate, we can use a temporal integration over different views to improve the classification results considerably. Our results have shown that a majority voting scheme is particularly efficient in combination with the nearest-neighbour classification approach in the object memory, since it allows to use more ensemble information of the exemplar-based representation stored in memory. In our experiments we use a history of 10 classifications, and assign the output class that achieves most single classification votes. An object is rejected as unknown if this majority vote is less than 50% or if the mean similarity to the majority representatives, measured in the Euclidean feature space, is below a fixed threshold.

5 Results

The complete system has been realized on a cluster of one dual processor PC for gaze control and image capture, one desktop PC running the speech recognition and synthesis system, and one dual processor PC performing all visual processing and online learning after the gaze selection. The recognition system is running at a frame rate of roughly 6Hz, which enables interaction and online learning with direct feedback on the learning result. A generic training scenario is shown in Fig. 2a, with typical ROI views of objects that are being processed. During all experiments the objects were freely rotated by hand to obtain a strong appearance variation.

In Fig.2b we show plots of the recognition performance versus training time during online learning. For this evaluation we train nine objects from a training set of 10 objects (upper row in Fig. 3) that was generated by storing 300 views per object from a typical training session. Then the tenth object is trained in steps of 10 images (1.67 sec in Fig. 2c) and a testing step is performed. The test is done by classifying a completely disjoint test set of 300 views per object that was collected using a different training person. Test performance is measured over all 300 test images of the currently trained object giving the classification rate as percentage of correctly recognized objects at this point of online learning. Then training proceeds until all 300 training images are used. The plots shown in Fig. 2b show the resulting classification rate, averaged over an ensemble of experiments, where each of the 10 objects was one time the final object.

We compare in Fig. 2b the conditions of either using ASDF segmentation or omitting it (and thus also rotation normalization), and with or without temporal integration with voting over a past history of 10 classifications. The results demonstrate that due to the cluttered background, training with the ASDF speeds up learning considerably and gives a significantly higher recognition rate. Using the temporal integration can additionally reduce the error from 15% after 50 seconds of training to 4% error. If we remove the color features and use only the shape representation in combination with ASDF and temporal integration we obtain a residual error of 10%, underlining the independent quality of the shape representation.

We visualize the actual time course of the different memory types during a training session of 18 objects in Figure 3. The plot displays the number of used representatives in the sensory and object memories together with the training dialogue (abbreviated, the actual dialogue is a little more elaborate). Starting from a completely empty object memory, we first perform a training of 10 objects. In this first phase the system first consistently matches the cola can to the previously trained "sun cream" object, and thus classifies the cola can initially as "sun cream", which is then corrected by the teacher. Due to the similar red-white color and shape composition the "mini car" is also first confused with the cola can, and is corrected. Due to the shape similarity the green bottle is first labeled as blue bottle, which is a reasonable error, as long as no correction signal is given. After the feedback by the teacher, the system has learned to discriminate the first 10 objects after 5 minutes of training from many different viewing angles, which is evaluated directly afterwards. In the second training phase 8 objects are added. The initial confusion occurs quite reasonably between cola can and a yellow can, another red car and the mini car, a new blue mug and the first blueishly patterned mug, and a new blue rubber duck and the initial yellow one. After the initial training in the second phase, the garlic press and police car object have to be additionally refined. After that second retraining phase, all 18 objects are classified from any reasonable viewing angle without further errors.

An important property of the system is that learning occurs most of the time and is not separated into artificial training and testing phases. This can be seen

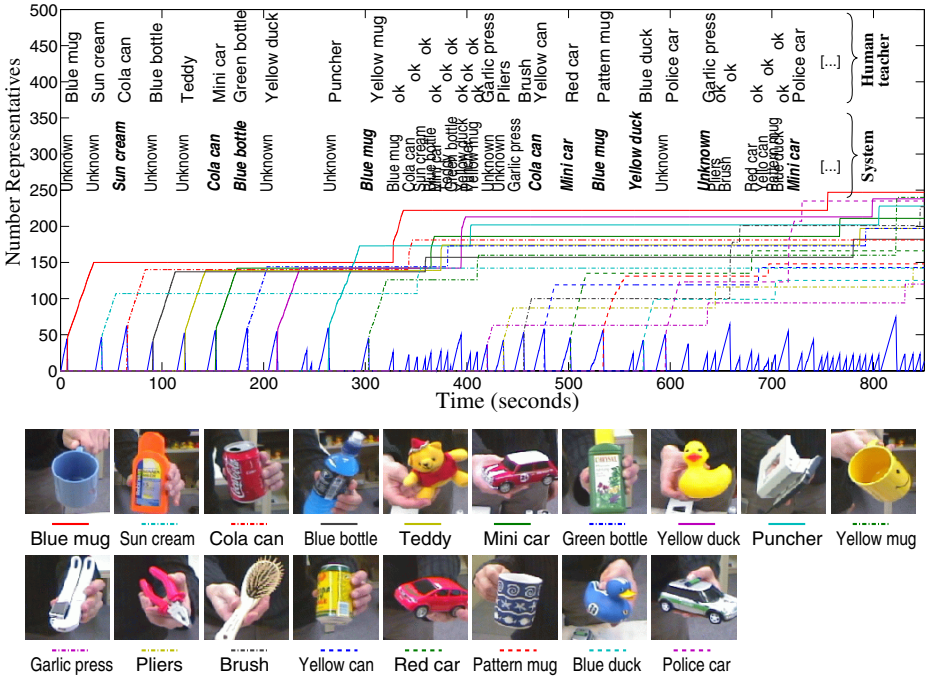


Fig. 3. Temporal learning dynamics during a training session for 18 objects. The plot shows the number of representatives for the sensory memory ("sawtooth" at bottom of plot) and representatives for each object in the object memory over time. The corresponding training dialogue is stated synchronously at the top. The top row states the given labels by the human trainer, while the bottom row gives the classification results of the system, before a human labeling is given. Errors of the system are printed in bold italics. From 0 to 310s the first 10 objects are trained, the recognition of these 10 objects is evaluated from 320s to 420s without any errors. From 420s to 730s another 8 objects are added, and all 18 objects are checked after 730s without errors.

from the time course in Fig. 3, where during the first evaluation of the first 10 objects between 320s and 420s the object memory is still expanding, due to the confirmation signals of the human teacher on the system classifications. The same applies to the second evaluation and error correction phase between 640s and 850s. The complete duration of the session until no further recognition errors are encountered is about 12 minutes. This highlights the gain in learning speed that can be achieved due to the active error correction process during learning. When the object memory is enlarged over time, we encounter a slight slowing down of the system frame rate from 6Hz to approximately 4Hz, since the comparison to the memory takes longer.

6 Discussion

We have presented an architecture for online learning of arbitrary objects that uses aspects of biologically motivated visual processing in a very efficient and robust way. To our knowledge it is the first system that focuses on real online learning of several objects of arbitrary color and shape and their later robust recognition in an unconstrained scenario. The representation is based on a neural model of the ventral pathway and combines a large storage capacity with robustness in difficult real-world scenarios. Due to the integration of speech dialogue with a context-dependent memory architecture we achieve a high level of interactivity that makes the training procedure simple and intuitive. We consider this as an important step towards cognitive vision systems for robotics and man-machine interfaces that gain considerable flexibility by learning.

Acknowledgments. We thank J. Eggert, A. Ceravola, and M. Stein for providing the processing system infrastructure. We thank F. Joublin and H. Janssen for their contributions to the setup of the speech recognition and synthesis system.

References

1. Steil, J.J., Wersing, H.: Recent trends in online learning for cognitive robotics. In: Proc. ESANN, Springer (2006) 77–87
2. Wersing, H., Körner, E.: Learning optimized features for hierarchical models of invariant recognition. Neural Computation **15**(7) (2003) 1559–1588
3. Jebara, T., Pentland, A.: Action reaction learning: Automatic visual analysis and synthesis of interactive behaviour. In: Int. Conf. Computer Vision Systems. (1999)
4. Garcia, L.M., Oliveira, A.A.F., Grupen, R.A., Wheeler, D.S., Fagg, A.H.: Tracing patterns and attention: Humanoid robot cognition. IEEE Intelligent Systems **15**(4) (2000) 70–77
5. Steels, L., Kaplan, F.: AIBO's first words. The social learning of language and meaning. Evolution of Communication **4**(1) (2001) 3–32
6. Roy, D., Pentland, A.: Learning words from sights and sounds: a computational model. Cognitive Science **26**(1) (2002) 113–146
7. Arsenio, A.: Developmental learning on a humanoid robot. In: Proc. Int. Joint Conf. Neur. Netw. 2004, Budapest. (2004) 3167–3172
8. Bekel, H., Bax, I., Heidemann, G., Ritter, H.: Adaptive computer vision: Online learning for object recognition. In: Proc. DAGM, Tuebingen. (2004) 447–454
9. Kirstein, S., Wersing, H., Körner, E.: Rapid online learning of objects in a biologically motivated recognition architecture. In: 27th Pattern Recognition Symposium DAGM, Springer (2005) 301–308
10. Goerick, C., Wersing, H., Mikhailova, I., Dunn, M.: Peripersonal space and object recognition for humanoids. In: Proc. Humanoids Conf., Tsukuba. (2005)
11. Götting, M., Steil, J., Wersing, H., Körner, E., Ritter, H.: Adaptive scene-dependent filters in online learning environments. In: Proceedings Eur. Symp. Neur. Netw. ESANN, Bruges. (2006)

Second-Order (Non-Fourier) Attention-Based Face Detection

Albert L. Rothenstein[1], Andrei Zaharescu[2], and John K. Tsotsos[1]

[1] Dept. of Computer Science & Engineering and Centre for Vision Research
York University, Toronto, Canada
{albertlr, tsotsos}@cs.yorku.ca
[2] INRIA Rhone-Alpes, Montbonnot, France
andrei.zaharescu@inrialpes.fr

Abstract. We present an attention-based face detection and localization system. The system is biologically motivated, combining face detection based on second-order circular patterns with the localization capabilities of the Selective Tuning (ST) model of visual attention [1]. One of the characteristics of this system is that the face detectors are relatively insensitive to the scale and location of the face, and thus additional processing needs to be performed to localize the face for recognition. We extend ST's ability to recover spatial information to this object recognition system, and show how this can be used to precisely localize faces in images. The system presented in this paper exhibits temporal characteristics that are qualitatively similar to those of the primate visual system in that detection and categorization is performed early in the processing cycle, while detailed information needed for recognition is only available after additional processing, consistent with experimental data and with certain theories of visual object recognition[2].

1 Introduction

One of the major limitations in current object recognition schemes is the inherent difficulty of extracting reliable and repeatable features from highly textured real-world images. One of the sources of this limitation is the fact that methods rely mainly on linear filtering through various kernels (e.g. for edge detection). The major area in which non-linear feature extraction techniques have been used is perceptual grouping (e.g. [3]), inspired by Gestalt psychology [4,5]. A continuous source of inspiration for researchers has been the study of the primate visual system, with results used mainly to augment edge detection algorithms [6,7]. While these results are very promising, they generally limit themselves to simple edge-based perceptual grouping and center-surround competition.

In the current paper we propose a novel approach: non-linear processing targeted at object detection/recognition within a biologically plausible framework. In particular, we address the task of frontal face detection. This paper is organized as follows: In Sect. 2 we briefly describe previous work done in face detection and provide a brief overview of the Selective Tuning model of visual

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 518–527, 2006.

attention. Section 3 describes our contribution – the algorithm proposed in order to perform face detection and the coupling with visual attention. Section 4 describes the implementation of the system and presents some of the results obtained. The results are discussed in Sect. 5.

2 Background

Detection is generally the first step in a face recognition system, an area that has received significant attention recently, especially for biometrics and security applications (see [8,9] for recent reviews). The best results seem to come from appearance-based and learning approaches. The work of Turk and Pentland [10] on PCA-based eigenfaces has been very influential not only in face detection and recognition, but also in the more general context of object recognition. Subsequent work [11,12] improves on the eigenfaces approach, mainly by using learning classifiers and clustering. The most successful recent face detection system, that of Viola and Jones [13,14], uses AdaBoost learning to build a very rapid "cascade" classifier based on weak classifiers (Harr-like basis functions). The original work on frontal faces has been extended to detect tilted and non-frontal faces by extending the set of basic features and by the introduction of a pose estimator [15]. Variations of the framework that use different basis sets have been presented, e.g. Gabor wavelets [16], and local orientations of gradient and Laplacian based filters [17].

The primate visual system consists of a multi-layer hierarchy with pyramidal abstraction [18], a structure that makes computations tractable, but characterized by a loss of spatial information. As information progresses up the hierarchical structure, neurons represent more and more abstract information, but with less and less spatial accuracy. Due to the nature of the pyramidal structure, a neuron activated at the highest level of the pyramid corresponds to a large subregion in the first layer of the pyramid. In an extreme situation, the top layer can consist of a single neuron that only fires if a face exists in the input image. A mechanism is needed to be able to go down the pyramidal structure and locate the detected item at high spatial resolution, a mechanism provided by the Selective Tuning (ST) model of visual attention [1,19] – see [20] for a comprehensive review of computational models of visual attention. This is performed in practice via a Winner-Take-All mechanism that will select the most activated region at the highest level. Results are propagated down the pyramidal structure through winner-take-all competitions within the winning receptive fields, until the first layer is reached. Regions that do not contribute to the high level decision are inhibited, thus eliminating distractors and improving the signal-to-noise ratio. This process effectively segments out the detected structure in the input layer, as demonstrated on video sequences (simulated and real) in [19]. A second feed-forward pass through the pyramidal structure will allow only the signals that participate in the detection task to propagate upwards. A second feed-forward pass through the network will provide a much cleaner detection result, since

features from the input layer that do not participate to the detection task and that would normally propagate up the pyramid are now blocked from the top layer.

3 Face Detection

The face detection system relies on circular pattern detectors based on second-order processing, corresponding to the behaviour of complex, end-stopped cortical neurons [21,22]. Dobbins demonstrated that end-stopped cells can be used to encode boundary curvature [23], while Koendrink provided a theoretical basis [24]. Second-order filtering has been previously used in computer vision in motion analysis [25], texture boundary extraction [26] and non-Cartesian feature detection [27,28]. The circular pattern detection is accomplished with the neural network presented in Figure 1. Each pathway detects end-stopped segments, and these are combined spatially to detect circular patterns [29]. The rectification step makes the system insensitive to the direction of contrast in the input image. The equations describing the filters are presented in the Appendix.

The idea behind the detection system presented here is that a face (in frontal view) is characterized by constellations of quasi-circular features at different scales (an idea originally proposed by Wilson [29]. Note that while the solution is biologically-inspired, we are not proposing that the primate visual system detects faces in this manner).

As it can be observed in Figure 2, we model a face by grouping circles at 3 spatial resolutions: small for eyes and nose-tip; medium for the eye-sockets and

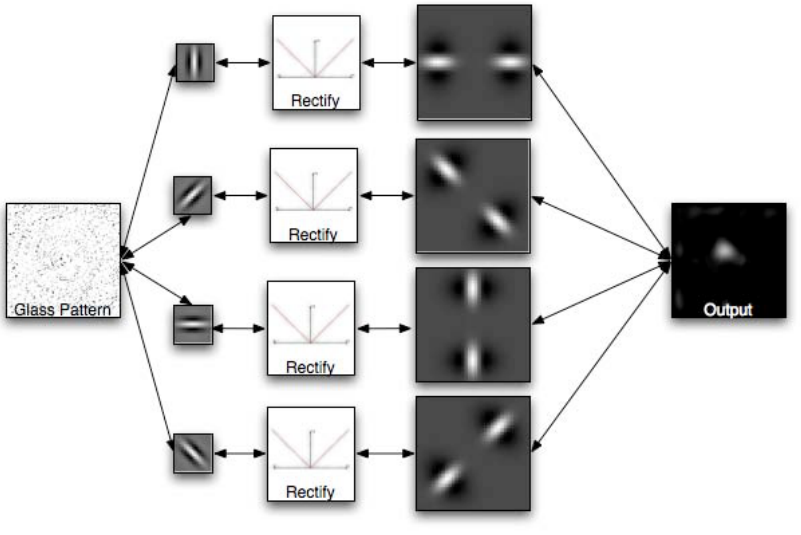


Fig. 1. Diagram of the circular pattern detectors. Only four pathways are represented for clarity. See text for details. Adapted from [29].

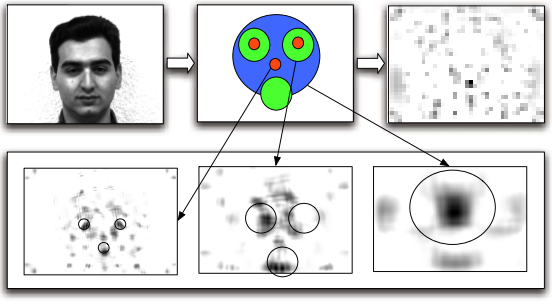


Fig. 2. Diagram illustrating the layout of the face detector based on the circular pattern detector at different three different scales

the mouth region and large for the overall face contour. A second-order circle detector is broadly tuned for a particular circle radius. By combining these circle detectors, we obtain a reliable face detector able to easily overcome changes in illumination, color and facial expressions.

So far have presented the face detection system focusing on the overall layout of the system. We have assumed that all the feature planes are of the same dimensions, and we have not worried about the computational cost of each feature and layer. To overcome the performance limitations of this approach, we implemented the system in a pyramidal fashion, coupled with The Selective Tuning (ST) model, which is able to recover the correct location of the detected stimuli. In Figure 3 we show the final layout of current the system. The sizes of the feature planes are also depicted, in order to illustrate overall pyramidal structure. All the connecting arrows between the feature planes are bi-directional, denoting the presence of top-down connections.

4 Implementation and Results

All simulations were implemented in the TarzaNN neural network simulator [30]. The simulator, instructions and all additional files needed to reproduce the results presented here are available online at http://www.TarzaNN.org. The simulations were performed on a Macintosh PowerMac G5. Note that the simulator will also work on other platforms, including Windows, Linux and Solaris. Testing has been performed on images from the Yale face database [31,32], on composite images derived from the database, and on a group photo. The size and spatial distribution of the circular detectors was tuned manually based on three examples from the Yale database, and we expect results to improve with the inclusion of a learning algorithm. All the results presented are "out of the box" using the system as designed, the only exception being the thresholding of the outputs in some figures, which was tuned manually where indicated. The manual thresholding was performed to enhance the graphical representation of the results, and has no functional role in the system.

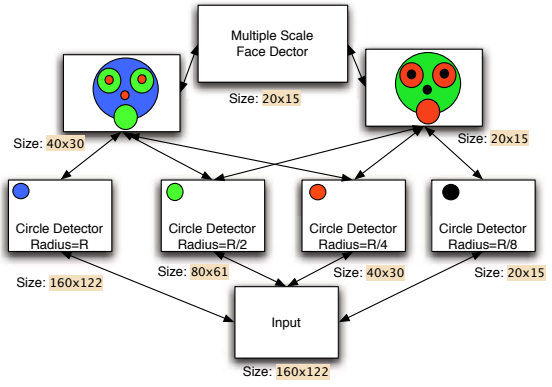


Fig. 3. Diagram illustrating the layout of the face detector based on the circular pattern detector at different scales

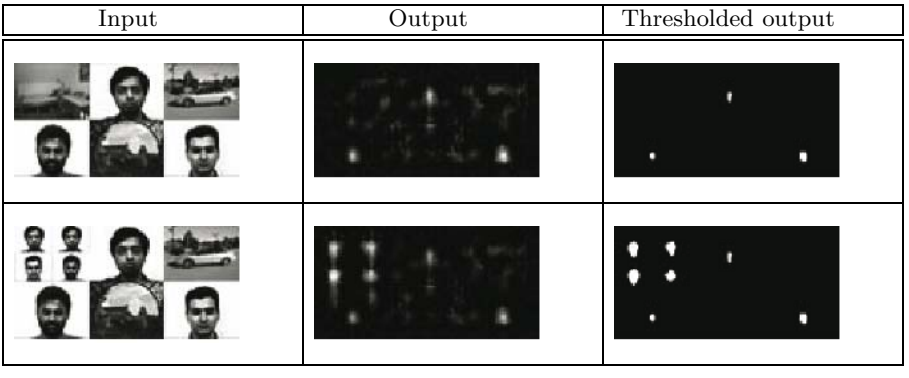


Fig. 4. Responses of the system to two composite images, including faces at two scales and other objects

Figure 4 illustrates the responses of the system to two composite images, including faces at two scales and other objects. In both cases, responses to faces are significantly stronger than those to other objects, including other circular features such as the wheels and front of the car. The thresholded results show that the system is able to detect and localize the faces.

Figure 5 is a group photo, with superimposed thresholded system output. Thresholding parameter was adjusted in favour of false-positives, so that all faces are shown as detected. Most false-positives are in the neck and chest areas. In the same Figure we present a couple of representative false-positives. Figure 5(b) illustrates a shirt, where the collars, shirt patterns, and occlusions form patterns that the system classifies as face-like. In Figure 5(a), symmetrical chin shadows and shirt neck line form a pattern that the system responds to. This pattern is caused mainly by the strong vertical lighting from above (the picture was taken in an atrium with glass ceiling).

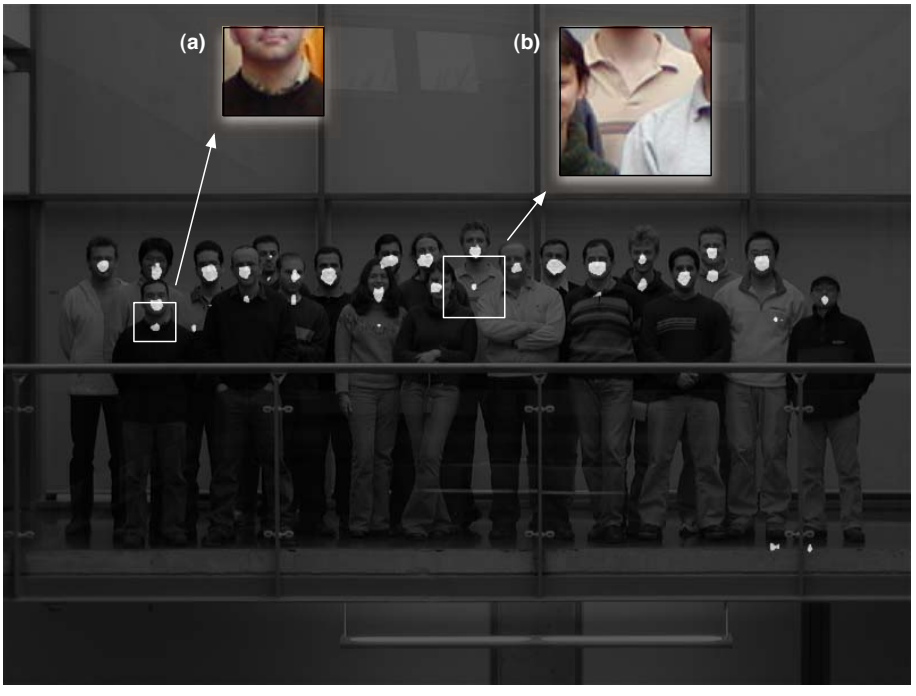
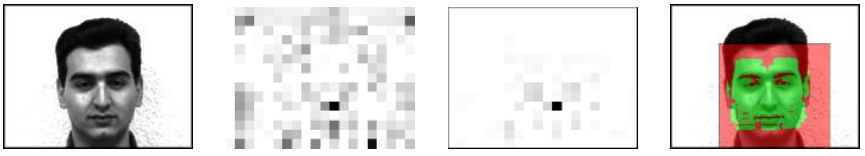


Fig. 5. Group photo with superimposed thresholded system output. Thresholding parameter was adjusted in favour of false-positives, so that all faces are shown as detected. Most false-positives are in the neck area and fall into one of the two categories illustrated above. These errors demonstrate both the flexibilty of the feature extraction process and the frailty of the template matching process. (a) "Face" created by chin shadows and shirt neck line (dominant lighting from above) and (b) "Face"-like shirt.

The inclusion of the attentional mechanism is demonstrated in Fig. 6 and 7. Fig. 6(b) shows the output without attention, note the very noisy output. Fig. 6(c) illustrates the effect of the ST attentional filtering on the face detector output, with a much clearer peak of activation corresponding to the detected face. Fig. 6(d) presents the localization of the face in the original input image, together with the inhibited region.

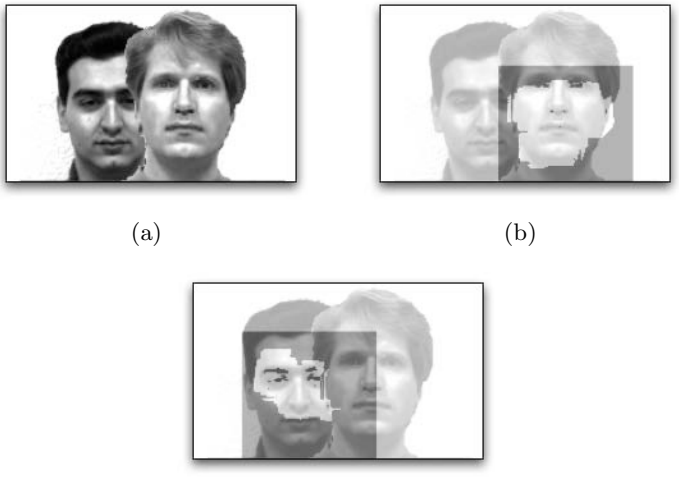
Figure 7 shows the behavior of the system with overlapping faces. In the first fixation, (Fig. 7(b)) the first face is detected and localized. Following this, ST inhibits the connections corresponding to the winning units, and a second pass through the network detects the and localizes the second face (Fig. 7(c)). Note that only the visible part of the second face is selected, since only those pixels contributed to the second detection.

The focus of the current implementation was on demonstrating the principles and feasibility of the method, and little effort has been invested in the performance aspects. In general the computational load of the method is significantly



(a) (b) (c) (d)

Fig. 6. Effects of attention on the face detection task. (a) Input image. (b) Output of the face detector layer without attention.(c) Output of face detector layer with attention. (d) Location of the face in the input layer using the attentional beam (the outer square represents the inhibited region).



(a) (b)

(c)

Fig. 7. Input with overlapping faces. (a) Input image. (b) First attentional fixation selects the first face (the outer square represents the inhibited region). (c) Second attentional fixation. Note that the second fixation only selects the visible part of the occluded face.

higher than that of other current face detection methods due to the multi-scale convolutions with fairly large kernels, and the ST adds a significant memory load due to the need to have all the intermediate results of the convolution available for the feedback pass.

5 Conclusions

One of the major limitations in current object recognition schemes is the inherent difficulty of extracting reliable and repeatable features from highly textured real-world images. Here we propose the use of second order processing

as a solution to this problem, and demonstrate the validity of the approach by applying it to the problem of face detection, with encouraging results. The currently presented detection system is able to correctly detect faces from the Yale Face database [31,32], under numerous variations (changes in illumination, colour, etc). The main limitation specific to this system is imposed by the ad-hoc nature of the template, generated manually and based on visual inspection. Stronger templates, based on learning and statistical analysis of face images would most likely improve the performance of the system. Implementing the system in a pyramidal, hierarchical fashion has posed the additional problem of recovering the exact location of the face in the input images, task accomplished by using the attentional feedback mechanism of Selective Tuning [1]. This is the first demonstration of Selective Tuning in a complex object recognition network.

While the current system does not solve the generic object recognition problem, it provides certain intuitions for how higher level cognition tasks can be performed within a biologically plausible framework. It is important to observe that the temporal structure of the proposed solution (i.e. detection followed by selection and recognition) is consistent with recent psychophysical results [2] that show a temporal lag between the detection and identification of faces. Also, see [33] for a review and discussion of results that highlight the importance of feedback connections and of early visual areas in conscious perception.

Acknowledgements

The authors would like to thank Kosta Derpanis, Sven Dickinson, and Radu Horaud for helpful comments and suggestions.

References

1. Tsotsos, J.K., Culhane, S.M., Wai, W.Y.K., Lai, Y.H., Davis, N., Nuflo, F.: Modeling visual-attention via selective tuning. Artif. Intell. **78**(1-2) (1995) 507–545
2. Grill-Spector, K., Kanwisher, N.: Visual recognition: as soon as you see it, you know what it is. Psychological Science **16**(2) (2005) 152–160
3. Lowe, D.G.: Perceptual organization and Visual Recognition. Kluwer (1985)
4. Kanizsa, G.: Organization in Vision: Essays on Gestalt Perception. Praeger (1979)
5. Koffka, K.: Principles of Gestalt Psychology. Kegan Paul, London (1936)
6. Zucker, S.W.: Computational and psychophysical experiments in grouping: Early orientation selection. In Beck, J., Hope, B., Rosenfeld, A., eds.: Human and Machine Vision. Academic Press (1983) 545–567
7. Grigorescu, C., Petkov, N., Westenberg, M.A.: Contour and boundary detection improved by surround suppression of texture edges. Image and Vision Computing **22**(8) (2004) 609–622
8. Hjelmåsa, E., Low, B.K.: Face detection: A survey. Computer Vision and Image Understanding **83**(3) (2001) 236–274
9. Yang, M.H., Kriegman, D.J., Ahuja, N.: Detecting faces in images: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(1) (2002) 34–58
10. Turk, M., Pentland, A.: Eigenfaces for recognition. Cognitive Neuroscience **13**(1) (1991) 71–96

11. Moghaddam, B., Pentland, A.: Probabilistic visual learning for object representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(7) (1997) 696–710
12. Sung, K.K., Poggio, T.: Example-based learning for view-based human face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(1) (1998) 39–51
13. Viola, P., Jones, M.: Robust real-time object detection. In: ICCV 2001 Workshop on Statistical and Computation Theories of Vision. (2001)
14. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE Conf. Computer Vision and Pattern Recognition. Volume 1. (2001) 511–518
15. Jones, M., Viola, P.: Fast multi-view face detection. In: IEEE Conference on Computer Vision and Pattern Recognition. (2003)
16. Zhang, L., Li, S.Z., Qu, Z.Y., Huang, X.: Boosting local feature based classifiers for face recognition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). Volume 5., Washington, D.C., USA (2004) 87
17. Mikolajczyk, K., Schmid, C., Zisserman, A.: Human detection based on a probabilistic assembly of robust part detectors. In Pajdla, T., Matas, J., eds.: European Conference on Computer Vision. Volume 1., Springer Verlag (2004) 69–82
18. Tsotsos, J.K.: A complexity level analysis of immediate vision. International Journal of Computer Vision **1**(4) (1987) 303–320
19. Tsotsos, J.K., Liu, Y., Martinez-Trujillo, J.C., Pomplun, M., Simine, E., Zhou, K.: Attending to visual motion. Comput. Vis. Image Und. **100**(1-2) (2005) 3–40
20. Rothenstein, A.L., Tsotsos, J.K.: Attention links sensing to recognition. Image and Vision Computing (in press doi:10.1016/j.imavis.2005.08.011) (2006)
21. Hubel, D., Wiesel, T.N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. Journal of Physiology (160) (1962) 106–154
22. Dreher, B.: Hypercomplex cells in the cat's striate cortex. Invest Ophthalmol. **5**(11) (1972) 355–356
23. Dobbins, A., Zucker, S.W., Cynader, M.S.: Endstopped neurons in the visual cortex as a substrate for calculating curvature. Nature **329** (1987) 438–441
24. Koenderink, J.J., Richards, W.A.: Two-dimensional curvature operators. J. Opt. Soc. Am. A **52** (1988) 1136–1141
25. Fleet, D., Black, M., Jepson, A.: Motion feature detection using steerable flow fields. In: Proceedings of the IEEE Computer Vision and Pattern Recognition Conference (CVPR). (1998) 274–281
26. von der Heydt, R., Peterhans, E., Baumgartner, G.: Illusory contours and cortical neuron responses. Science **224** (1984) 1260–1262
27. Gallant, J., Braun, J., Van Essen, D.C.: Selectivity for polar, hyperbolic, and cartesian gratings in macaque visual cortex. Science **259** (1993) 100–103
28. Gallant, J.L., Connor, C.E., Rakshit, S., Lewis, J., Van Essen, D.C.: Neural responses to polar, hyperbolic, and cartesian gratings in area V4 of the macaque monkey. Journal of Neurophysiology **76** (1996) 2718–2737
29. Wilson, H.R.: Non-Fourier cortical processes in texture, form, and motion perception. In Ulinski, P.S., Jones, E.G., eds.: Cerebral Cortex. Volume 13. Kluwer Academic/ Plenum Publishers, New York (1999)
30. Rothenstein, A.L., Zaharescu, A., Tsotsos, J.K.: Tarzann : A general purpose neural network simulator for visual attention modeling. In Paletta, L., Tsotsos, J.K., Rome, E., Humphreys, G., eds.: Lecture Notes in Computer Science. Volume 3368. Springer Verlag (2005) 159–167

31. Bellhumer, P.N., Hespanha, J., Kriegman, D.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence **17**(7) (1997) 711–720
32. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(6) (2001) 643–660
33. Lamme, V.A.F., Roelfsema, P.R.: The distinct modes of vision offered by feedforward and recurrent processing. Trends in Neurosciences **23**(11) (2000) 571–579

Appendix

Eq. 1 represents an edge detector composed of a central elongated excitatory lobe flanked by two inhibitory areas. Eq. 2 represents the second stage filters, composed of a similar arrangement of activation lobes, but each filter has two detectors symmetrically shifted by the radius of the circle for which the detector is tuned. Filters for each pathway are rotated by an appropriate amount, and the second stage filters are orthogonal to those in the first stage. See [29] for details on the choice of parameters and on the weighting of the pathways.

$$F(x,y) = \left[Ae^{-\frac{x^2}{\sigma_1^2}} - Be^{-\frac{x^2}{\sigma_2^2}} - Ce^{-\frac{x^2}{\sigma_3^2}} \right] e^{-\frac{y^2}{\sigma_y^2}} \tag{1}$$

$$W(x,y) = \left(Ae^{-\frac{x^2}{\sigma_E^2}} - Be^{-\frac{x^2}{\sigma_I^2}} \right) \left(e^{-\frac{(y-\Delta)^2}{\sigma_y^2}} + e^{-\frac{(y+\Delta)^2}{\sigma_y^2}} \right) \tag{2}$$

Requirements for the Transmission of Streaming Video in Mobile Wireless Networks

Vasos Vassiliou, Pavlos Antoniou, Iraklis Giannakou, and Andreas Pitsillides*

Networks Research Group
Computer Science Department
University of Cyprus
vasosv@cs.ucy.ac.cy

Abstract. The ability to transmit video and support related real-time multimedia applications is considered important in mobile networks. Video streaming, video conferencing, online interactive gaming, and mobile TV are only a few of the applications expected to support the viability, and survival, of next generation mobile wireless networks. It is, therefore, significant to analyze the interaction of the particular media and applications. This paper presents the characteristics of mobile wireless networks and relates these characteristics to the requirements of video transmission. The relationship derived is based not only on the objective QoS metrics measured in the network, but also on the subjective quality measures obtained by video viewers at end hosts. Through this work we establish guidelines for the transmission of video based on the limits of mobile and wireless networks. We believe that the results help researchers and professionals in the fields of video production and encoding to create videos of high efficiency, in terms of resource utilization, and of high performance, in terms of end-user satisfaction.

1 Introduction

The basic factor behind the success of Third Generation mobile networks, like the Universal Mobile Telecommunications System(UMTS), is the availability of attractive, useful, and low cost services for the final user. Today, a very limited number of multimedia services for digital mobile communication networks exist, because of the limited abilities of user terminals, the low data transmission rates, and the relative cost. Recently, an increasing demand for digital services for the distribution stored video over the Internet is observed. With the spread of Third Generation mobile networks and the increased capabilities of mobile equipment with the ability of capture and playback video, an increase on the demand of these services is expected. Video has been an important media for communications and entertainment for many decades. The growth and popularity of the Internet in the mid-1990s motivated video communication over best-effort packet networks. Video over best-effort packet networks is complicated by a number of

* This work has been performed under the RPF project BINTEO and the UCY project ADAVIDEO.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 528–537, 2006.

factors including unknown and time-varying bandwidth, delay, and losses, as well as many additional issues such as how to fairly share the network resources amongst many flows and how to efficiently perform one-to-many communication for popular content. Video communication over a dynamic environment, such as a mobile and wireless network is much more difficult than over a static channel, since the bandwidth, delay, and loss are not known in advance and are unbounded.

When the streaming path involves both wired and wireless links, some additional challenges evolve. The first challenge involves the much longer packet delivery time with the addition of a wireless link. The long round-trip delay reduces the efficiency of a number of end-to-end error control mechanisms. The second challenge is the difficulty in inferring network conditions from end-to-end measurements. In high-speed wired networks, packet corruption is so rare that packet loss is a good indication of network congestion, the proper reaction of which is congestion control. In wireless networks, however, packet losses may occur due to corruption in the packet. In the future, we will have access to a variety of mobile terminals with a wide range of display sizes and capabilities. In addition, different radio-access networks will make multiple maximum-access link speeds available. Because of the physical characteristics of cellular radio networks, the quality and, thus, the data rate of an ongoing connection will also vary, contributing to the heterogeneity problem. A related problem is how to efficiently deliver streamed multimedia content over various radio-access networks with different transmission conditions. This is achievable only if the media transport protocols incorporate the specific characteristics of wireless links.

This paper intends to give an understanding of the transmission of video over mobile wireless networks. Adopting the transmission of MPEG4-encoded video streams over wireless network environments, we investigate the types of errors that can be observed, using objective video quality metrics such as PSNR. Furthermore, we provide subjective video quality estimation based on the evaluation of decoded video streams by informed viewers.

The paper is organized as follows. Section 2 provides an overview of the characteristics of the most common mobile and wireless networks. Section 3 provides background information on the objective and subjective quality evaluation methods used in this paper. Section 4 describes the video characteristics, the setup, and the scenarios used to evaluate the transmission of streaming video in a wireless network. Section 5 presents the results of the objective and subjective evaluations. The paper ends with a last section on conclusions.

2 Characteristics of Mobile and Wireless Networks

2.1 Cellular Wireless Networks

Second Generation (2G) Cellular Networks. The main aim in the design of the 2G systems was the maximization of the system capacity, measured as the number of users per spectrum per unit area. 2G makes heavy use of digital technology through the use of digital vocoders, Forward Error Correction

(FEC), and high level digital modulation to improve voice quality, security and call reliability. The GSM technology has been a very stable, widely accepted and probably the most popular standard for mobile communication. The major drawback of GSM with respect to data and video is that GSM-enabled systems do not support high data rates. GSM supports only low rates for data services (up to 9.6 Kbps) and Short Message Services (SMS), thus, it is unable to handle complex data such as video. In addition, the GSM networks are not compatible with the current TCP/IP and other common networks because of differences in network hardware, software and protocols.

2.5G Cellular Networks (GPRS). The General Packet Radio Service (GPRS) is a standard developed by the European Telecommunications Standards Institute (ETSI) on packet data in GSM systems. GPRS is designed to provide a high data rate packet-switched bearer service in a GSM network. GPRS has a number of important benefits with respect to data and video. The most important are: (a) that it uses the same core infrastructure for different air interfaces, (b) it operates on an integrated telephony and Internet infrastructure, (c) it is always on, reducing the time spent in setting up and tearing down connections, (d) it is designed to support bursty applications, such as e-mail, telemetry, broadcast services and web browsing, and (e) it supports high-speed data services with rates up to 384Kbps.

3G Mobile Networks. 3G Systems are intended to provide a global mobility with wide range of services including telephony, paging, messaging, Internet and broadband data. UMTS offers teleservices and bearer services, which provide the capability for information transfer between access points. It is possible to negotiate the characteristics of a bearer service at session or connection establishment and renegotiate them during the session or connection. Bearer services have different QoS parameters for maximum transfer delay, delay variation and bit error rate. UMTS network services have different QoS classes for four types of traffic: Conversational class (voice, video telephony, video gaming) , Streaming class (multimedia, video on demand, webcast), Interactive class (web browsing, network gaming, database access), Background class (email, SMS, downloading).

Offered data rate targets are: 144 Kbps for satellite and rural outdoor, 384 Kbps for urban outdoor, and 2048 Kbps for indoor and low range outdoor. These are the maximum theoretical values in each environment for downlink speeds. The actual data rates may vary from 32Kbps, for a single voice channel, to 768 Kbps in urban low speed connections depending always on the class of service supported.

2.2 IEEE 802.11

Wireless local area networks (WLANs) based on the IEEE 802.11 standard are a significant and viable alternative to wireless connectivity. The standard has currently three variations widely deployed. The 802.11b operates on the 2.4GHz band and has a maximum theoretical data rate of 11Mbps, but operates also on 1, 2 and 5Mbps. The 802.11a and g operate on the 5GHz and 2.4GHz bands

respectively and both have a maximum theoretical data rate of 54Mbps. Using different modulation schemes they can also operate on the lower scales of 6, 10, 12, 18, 36, and 48 Mbps.

Based on CSMA/CA, a common resource sharing MAC protocol, 802.11 also adheres to the characteristic that the data rate allocated to each user is inversely proportional to the number of users in the local network. Therefore, the practical data rates are usually lower than those mentioned above.

3 Video Quality Assessment Schemes

3.1 Objective QoS Measures

In an optimal case, the quality of video is monitored during transmission. According to measurements, adjustment of parameters and possible retransmission of the data is carried out. Objective quality assessment methods of digital video can be classified into three categories. In the first category, the quality is evaluated by comparing the decoded video sequence to the original. The objectivity of this method is owed to the fact that there is no human interaction; the original video sequence and the impaired one are fed to a computer algorithm that calculates the distortion between the two. The second category contains methods that compare features calculated from the original and the decoded video sequences. The methods of the third category make observations only on decoded video and estimate the quality using only that information. The Video Quality Experts Group (VQEG) calls these groups the full, the reduced and the no reference methods [1]. Traditional signal distortion measures use an error signal to determine the quality of a system. The error signal is the absolute difference between the original and processed signal. The traditional quality metrics are the Root Mean Square Error (RMSE), the Signal-to-Noise Ratio(SNR), and the Peak Signal-to-Noise Ratio (PSNR) in dB. In this work we employ a Full reference method and use the PSNR as the objective quality metric.

3.2 Subjective QoS Measures

There are numerous metrics used to express the objective quality of an image or video, which cannot, however, characterize fully the response and end satisfaction of the viewer. Perceived measure of the quality of a video is done through the human "grading" of streams which helps collect and utilize the general user view. There is a number of perceived quality of service measurement techniques. Most of them are explained in [2]. The following are the most popular: a) DSIS (Double Stimulus Impairment Scale) b) DSCQS (Double Stimulus Continuous Quality Scale) c) SCAJ (Stimulus Comparison Adjectival Categorical Judgement) d) SAMVIQ (Subjective Assessment Method for Video Quality evaluation)

In this work we have used the SAMVIQ [3] method. SAMVIQ is based on random playout of the test files. The individual viewer can start and stop the evaluation process as he wishes and is allowed to determine his own pace for performing grading, modifying grades, repeating playout when needed, etc. With

the SAMVIQ method, quality evaluation is carried out scene after scene including an explicit reference, a hidden reference and various algorithms (codecs). As a result, SAMVIQ offers higher reliability, i.e. smaller standard deviations. A major advantage of this subjective evaluation scheme is in the way video sequences are presented to the viewer. In SAMVIQ video sequences are shown in multi-stimulus form, so that the user can choose the order of tests and correct their votes, as appropriate. As the viewers can directly compare the impaired sequences among themselves and against the reference, they can grade them accordingly. Thus, viewers are generally able to discriminate the different quality levels better with SAMVIQ than with the other methods. In addition, in this method there is only one viewer at a time, which alleviates a "group effect".

4 Evaluation Setup and Scenarios

4.1 Topology

The evaluation topology consists of one Video Streaming Server, two backbone routers and video clients of variable types and connectivity methods (fixed, mobile, wired, wireless) as shown in Fig. 1. The video streaming server is attached to the first backbone router with a link which has 10Mbps bandwidth and 10ms propagation delay. These values remain constant during all scenarios. This router is connected to a second router using a link with unspecified and variable bandwidth, propagation delay, and packet loss. The different parameter values used to characterize this variable link are shown in Table 1. Using this topology, we conducted several experiments for two different sample sequences and with fixed-wired clients, fixed-wireless clients and mobile-wireless clients.

4.2 Variable Test Parameters

The choice of the parameters used in the video quality evaluations (Table 1) was based on the typical characteristics of mobile and wireless networks, as these are

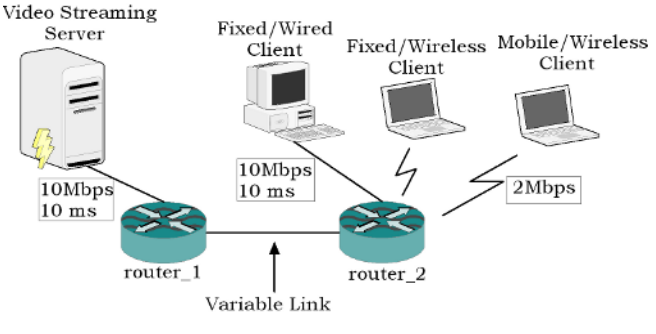


Fig. 1. Video Stream Evaluation Setup

Table 1. Variable Parameters

Video Stream Bit Rate	*Link Bandwidth*	*Propagation Delay*	*Packet Loss*
64 Kbps	64 Kbps	10 ms	
128 Kbps	100 Kbps	50 ms	10^{-5}
256 Kbps	256 Kbps	100 ms	10^{-3}
512 Kbps	512 Kbps	200 ms	
768 Kbps	1 Mbps	400 ms	

described in Section 2. For example, the Link Bandwidth can be considered as either the last hop access link BW or the available BW to the user. The values chosen can represent typical wired home access rates (modem, ISDN, xDSL) or different bearer rates for UMTS.

4.3 Test Sequences

The test sequences used in this work were the sample sequences Foreman and Claire. The sequences were chosen because of their different characteristics. The first is a stream with a fair amount of movement and change of background, whereas the second is a more static sequence. The characteristics of these sequences are shown in Table 2. The sample sequences were encoded in MPEG4 format with a free software tool called FFMPEG encoder [4]. The two sequences have temporal resolution 30 frames per second, and GoP (Group of Pictures) pattern IBBPBBPBBPBB. Each sequence was encoded at the rates shown in Table 1. The video stream bit rate[1] varies from 64Kbps to 768Kbps. This rate is the average produced by the encoder. Since the encoding of the sample video sequences is based on MPEG4, individual frames have variable sizes.

Table 2. Video Characteristics

Trace	*Resolution*	*Total Frames*	*# I Frames*	*# P Frames*	*# B Frames*
Foreman.yuv	176x144	400	34	100	266
Claire.yuv	176x144	494	42	124	328

4.4 Data Collection

All the aforementioned experiments were conducted with an open source network simulator tool NS2 [5]. Based on the open source framework called EvalVid [6] we were able to collect all the necessary information needed for the objective video quality evaluation like PSNR values, frames lost, packet end to end delay and packet jitter. Some new functionalities were implemented in NS2 from [7] in order to support EvalVid. The whole data collection procedure and PSNR evaluation is illustrated in Fig. 2.

[1] The terms video stream bit rate and video encoding rate are used interchangeably in this paper.

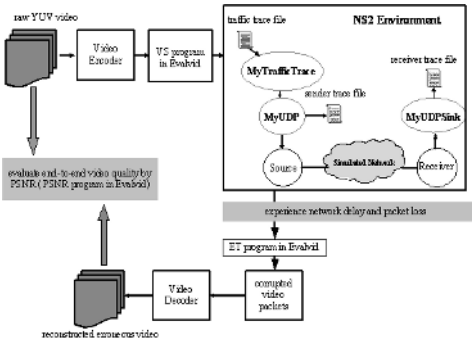


Fig. 2. PSRN calculation through evalvid

5 Results

In this section we analyze results obtained from the above scenario evaluations. Given the very large number of produced streams, we chose to present and analyze only one scenario. The results presented are for the following case: single user, single video stream, No background traffic, Foreman test sequence, mobile and wireless terminal. All other parameters are variable as shown in Table 1. To identify if and how the different parameters affect the objective value of PSNR we compare them in pairs.

5.1 Link Bandwidth and Propagation Delay

The effect of propagation delay and link bandwidth on the PSNR while keeping the encoding rate steady at 64Kbps and 256Kbps is presented in Fig. 3. These graphs show that the objective values remain relatively constant with the change in either variable, with a slight general increase for high link BW values and counter-intuitively in high delay values as well. There is also an overall upward shift by 1dB when the encoding rate is increased from 64Kbps to 256Kbps. The PSNR is extremely low in the case where the encoding rate is higher than the link BW, as it is evident by Fig. 3b.

5.2 Video Encoding Rate and Propagation Delay

The effect of propagation delay and video encoding rate on the PSNR when keeping the link BW constant at 500Kbps and 1Mbps is presented in Fig. 4. The results show that for the 1M case the results are similar to those of Fig. 3. For the 500K case we observe that the PSNR remains at the same levels with respect to delay, but is significantly reduced when the video encoding rate is at 512Kbps and 768Kbps with the PSNR of the latter being the worst at around 15dB. This leads us to believe that there is a stronger relationship between link BW and encoding rate, than between the link BW and the propagation delay.

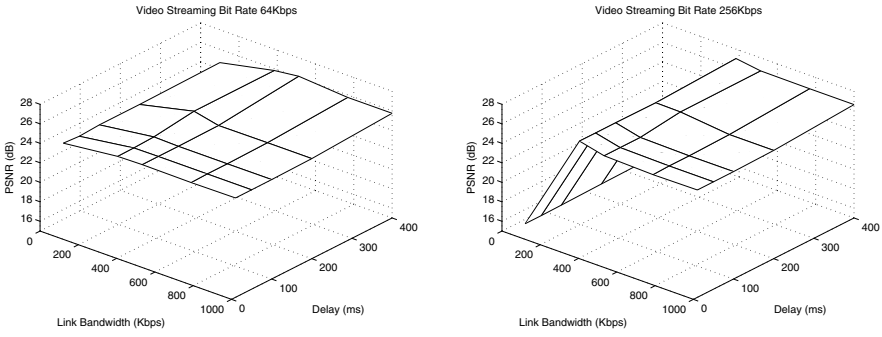


Fig. 3. Mean PSNR values vs Link Bandwidth and Delay (a) 64K Video Encoding Rate, (b) 256K Video Encoding Rate

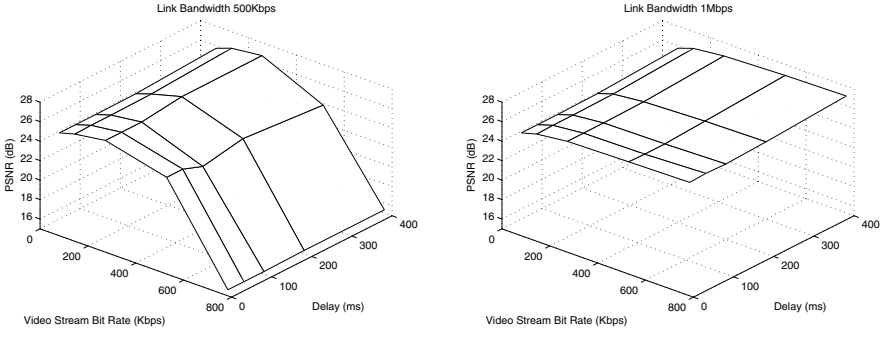


Fig. 4. Mean PSNR values vs Encoding Rate and Delay (a) 500K link BW, (b) 1M link BW

5.3 Link Bandwidth and Video Encoding Rate

Fig. 5 contains the most notable results. More specifically, for both values of delay considered (10ms and 400ms) the PSNR drops dramatically when the encoding rate is higher than the link bandwidth. This is somewhat intuitive if we consider that in those instances the packet losses of the video stream are very big, and approaching 100%, which in turn means that the PSNR is low as well.

5.4 Evaluation of Perceived Quality of Service

The set of video streams that were recorded on the receiving site of the evaluation setup was used as input to the PQoS evaluation method explained in Section 3.2. We used the software tool called "MSU Perceptual Video Quality tool" [8] which is a tool for subjective video quality evaluation implementing SAMVIQ. The score grades in this method range from 0 to 100. The videos

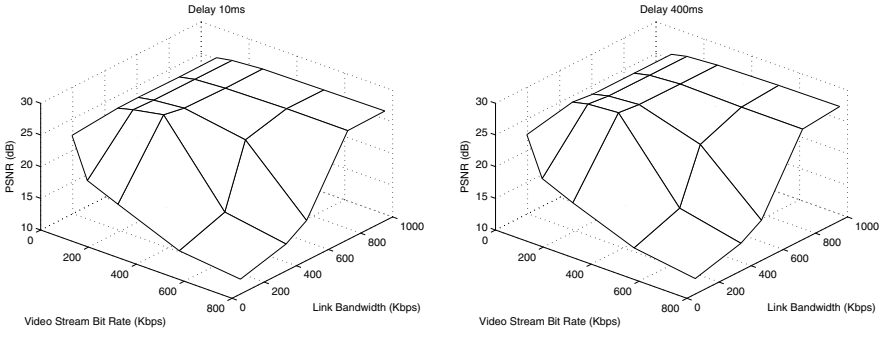


Fig. 5. Mean PSNR values vs Encoding Rate and Link Bandwidth (a) Delay 10ms, (b) Delay 400ms

Table 3. Relationship of PSNR with MOS

PSNR (dB)	*MOS*	P-QoS Category
> 27.2	81-100	1 Excellent
26.9 - 27.2	61-80	2 Good
26.1 - 26.9	41-60	3 Fair
16.2 - 26.1	21-40	4 Poor
< 16.2	0-20	5 Bad

were evaluated by a group of 20 students at the University of Cyprus. Table 3 presents the relationship between the average value of the students' subjective grading and the objective value obtained through EvalVid. values corresponding to each category is not similar and do not have a liner relationship with the MOS. The video streams which scored high had also an extremely high PSNR. The *Good* and *Fair* categories have also a small range of PSNR values (0.3dB and 0.8dB respectively) whereas the low categories get the bulk of the scores. This phenomenon illustrates clearly how inappropriate is PSNR to evaluate the actual QoS as perceived by the user.

6 Conclusions

In this paper we described the characteristics of mobile wireless networks and related these characteristics to the requirements of video transmission. The tests and simulations analyzed in this paper were designed to correlate objective video quality metrics with subjective video quality. Standard objective metrics such as PSNR were taken into consideration in order to evaluate objective quality.

A novel methodology called SAMVIQ was used for subjective evaluations. This method can be efficiently used for the evaluations of video sequences in both clear and error-prone environments. This set of values, when correlated with the conditions affecting PSNR help us reach some conclusions. Due to space limitations we could not include all additional metrics values for the resultant packet loss, delay, and jitter.

The experimental results show that the higher the video bit-rate the higher the QoS in terms of objective and subjective video quality evaluation measures. Of course the QoS depends primarily on the link bandwidth. The best quality in terms of PSNR as well as user-perceived quality is achieved when the encoding rate is less than or equal to the link BW or available BW. Needless to say that the most prevalent objective video quality metric does not correlate directly with viewer's perceived quality. Nevertheless the higher the PSNR values the higher the viewer perceived quality.

Through this work we establish guidelines for the transmission of video based on the limits of mobile and wireless networks. We believe that the results help researchers and professionals in the fields of video production and encoding to create videos of high efficiency, in terms of resource utilization, and of high performance, in terms of end-user satisfaction.

References

1. Rohaly M. et al. (2000) "Video Quality Experts Group: Current Results and Future Directions," In: SPIE Visual Communications and Image Processing, Perth, Australia, June 21-23, 2000, Vol. 4067, p.742-753.
2. ITU-R BT.500-11 "Methodology for the subjective assessment of the quality of television pictures"
3. F. Kozamernik, V. Steinmann, P. Sunna, E. Wyckens, "SAMVIQ: A New EBU Methodology for Video Quality Evaluations in Multimedia," SMPTE Motion Imaging Journal, April 2005, pp. 152-160.
4. FFMPEG Multimedia System. `http://ffmpeg.sourceforge.net/index.php`
5. Network Simulator 2 `http://www.isi.edu/nsnam/ns/`
6. J. Klaue, B. Rathke and A. Wolish, "EvalVid - A Framework for Video Transmission and Quality Evaluation," In Proc. of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, pp. 255-272, Urbana, Illinois, USA, September 2003
7. Chih-Heng Ke, Cheng-Han Lin, Ce-Kuen Shieh, Wen-Shyang Hwang, A Novel Realistic Simulation Tool for Video Transmission over Wireless Network, The IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC2006), June 5-7, 2006, Taichung, Taiwan.
8. MSU Perceptual Video Quality tool `http://compression.ru/video/quality_measure/`

Wavelet Based Estimation of Saliency Maps in Visual Attention Algorithms

Nicolas Tsapatsoulis[1] and Konstantinos Rapantzikos[2]

[1] Department of Computer Science,
University of Cyprus, CY 1678, Cyprus
phone: +357-2289-2747; fax: +357-2289-2701
nicolast@ucy.ac.cy
[2] School of Electrical Engineering,
National Technical University of Athens, 9 Iroon Polytechniou Str.,
15780, Zografou, Greece
phone: +30-210-7724351; fax: +30-210-7722492
rap@image.ntua.gr

Abstract. This paper deals with the problem of saliency map estimation in computational models of visual attention. In particular, we propose a wavelet based approach for efficient computation of the topographic feature maps. Given that wavelets and multiresolution theory are naturally connected the usage of wavelet decomposition for mimicking the center surround process in humans is an obvious choice. However, our proposal goes further. We utilize the wavelet decomposition for inline computation of the features (such as orientation) that are used to create the topographic feature maps. Topographic feature maps are then combined through a sigmoid function to produce the final saliency map. The computational model we use is based on the Feature Integration Theory of Treisman *et al* and follows the computational philosophy of this theory proposed by Itti *et al*. A series of experiments, conducted in a video encoding setup, show that the proposed method compares well against other implementations found in the literature both in terms of visual trials and computational complexity.

Keywords: Visual attention, saliency maps, perceptual video coding.

1 Introduction

In saliency-based visual attention algorithms efficient computation of the saliency map is critical for several reasons. First, the algorithm itself should model in an appropriate manner the visual attention process in humans. This is by no means easy. Visual attention theory has been constructed mainly by neuroscientists without taking into account computational modeling difficulties. On the other hand, computational models have been developed mainly by engineers and computer scientists which in several cases compromise theory in favor of implementation efficiency. Second, algorithm's implementation should conform to real life situations and settings. Perceptual based video coding is one of the areas that visual attention fits well.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 538 – 547, 2006.

However, in applications like video-telephony real-time video encoding is a requirement. Therefore, if a computational model of visual attention is to be used, then its implementation should be both fast and effective. Finally, integration of the topographic feature maps into the overall saliency map should be performed in a reasonable way and not ad hoc as it happens in most existing models where normalization and additions is the combination method of preference.

2 Saliency-Based Visual Attention

2.1 Existing Computational Models

The basis of many visual attention models proposed over the last two decades [1] – [3] is the Feature Integration Theory (FIT) of Treisman *et al* [4] that was derived from visual search experiments. According to this theory, features are registered early, automatically and in parallel along a number of separable dimensions (e.g. intensity, color, orientation, size, shape etc).

One of the major saliency-based computational models of visual attention is presented in [5] and deals with static color images. Visual input is first decomposed into a set of topographic feature maps. Different spatial locations then compete for saliency within each map, such that only locations that locally stand out from their surround can persist. All feature maps feed, in a purely bottom-up manner, into a master saliency map. Itti and Koch [6, 7] presented an implementation of the proposed saliency-based model. Low-level vision features (color channels tuned to red, green, blue and yellow hues, orientation and brightness) are extracted from the original color image at several spatial scales, using linear filtering. The different spatial scales are created using Gaussian pyramids, which consist of progressively low-pass filtering and sub-sampling the input image. Each feature is computed in a center-surround structure akin to visual receptive fields. Using this biological paradigm renders the system sensitive to local spatial contrast rather than to amplitude in that feature map. Center-surround operations are implemented in the model as differences between a fine and a coarse scale for a given feature. Seven types of features, for which evidence exists in mammalian visual systems, are computed in this manner from the low-level pyramids.

2.2 The Proposed Wavelet-Based VA Model Implementation

In this work we begin from the model of Itti & Koch and make use of the YCrCb colour model [8], instead of RGB, and the hierarchical wavelet decomposition of Mallat [9] to provide an efficient way of computing saliency maps in static images and video sequences.

Let's consider a colour image *f*, represented in using the *YCrCb* colour model. Channel *Y* corresponds to the illumination, and can be used for identifying outstanding regions according to illumination and orientation, while *Cr* (Chrominance Red) and *Cb* (Chrominance Blue) correspond to the chrominance components and can be used to identify outstanding regions according to colour.

In the proposed method salient areas based on intensity, orientation, and colour are computed in several scales. In this way, outstanding objects of different sizes are recognized as such. Combining the results of intensity, orientation, and colour feature maps at various scales provide the intensity (C_I), orientation (C_O) and colour (C_C) conspicuity maps. The motivation for the creation of the separate conspicuity maps is the hypothesis that similar features compete strongly for saliency, while different modalities contribute independently to the saliency volume. Hence, after the intra-feature competition the three conspicuity maps are normalized and summed into the saliency map. Both feature and conspicuity maps are combined using a saturation function (sigmoid) to preserve the independency and added value of each separate feature channel and scale.

The proposed method is analysed in detail in the following paragraphs.

3 Saliency-Map Computation

In order of multiscale analysis to be performed a pair of low-pass $h_\phi(\cdot)$ and high-pass filter $h_\psi(\cdot)$ are applied to each one of the image's colour channels *Y, Cr, Cb,* in both the horizontal and vertical directions. The filter outputs are then sub-sampled by a factor of two, generating the high-pass bands H (horizontal detail coefficients), V (vertical detail coefficients), D (diagonal detail coefficients) and a low-pass subband A (approximation coefficients). The process is then repeated to the *A* band to generate the next level of the decomposition.

The following equations describe mathematically the above process for the illumination channel *Y*. It is obvious that the same process applies also to *Cr* and *Cb* chromaticity channels:

$$\begin{aligned}
Y_A^{-(j+1)}(m,n) &= \left(h_\phi(-m) * \left(Y_A^{-j}(m,n) * h_\phi(-n)\right)\downarrow^{2n}\right)\downarrow^{2m} \\
Y_H^{-(j+1)}(m,n) &= \left(h_\psi(-m) * \left(Y_A^{-j}(m,n) * h_\phi(-n)\right)\downarrow^{2n}\right)\downarrow^{2m} \\
Y_V^{-(j+1)}(m,n) &= \left(h_\phi(-m) * \left(Y_A^{-j}(m,n) * h_\psi(-n)\right)\downarrow^{2n}\right)\downarrow^{2m} \\
Y_D^{-(j+1)}(m,n) &= \left(h_\psi(-m) * \left(Y_A^{-j}(m,n) * h_\psi(-n)\right)\downarrow^{2n}\right)\downarrow^{2m}
\end{aligned} \tag{1}$$

where $*$ denotes convolution, $Y_A^{-j}(m,n)$ is the approximation of *Y* channel at *j-th* level (note that $Y_A^{-0}(m,n) = Y$), and $\downarrow^{2m}$ and $\downarrow^{2n}$ denote down-sampling by a factor of two along rows and columns respectively.

Following the decomposition of each colour channel at specific depth we use *center-surround* differences to enhance regions that locally stand-out from the surround. Center-surround operations resemble the preferred stimuli of cells found in some parts of the visual pathway (lateral geniculate nucleus-LGN) [4]. Center-surround differences are computed in a particular scale (level *j*) using the morphological

gradient (difference between morphological opening and closing [8]) for the intensity and colour feature maps and the sum of differences of detail bands for the orientation feature map, as shown in the following equations:

$$I^{-j}(m,n) = Y_A^{-j}(m,n) \bullet b - Y_A^{-j}(m,n) \circ b \tag{2.1}$$

$$O^{-j}(m,n) = \left|Y_D^{-j}(m,n) - Y_H^{-j}(m,n))\right| + \left|Y_D^{-j}(m,n) - Y_V^{-j}(m,n))\right| + \left|Y_V^{-j}(m,n) - Y_H^{-j}(m,n))\right| \tag{2.2}$$

$$CR^{-j}(m,n) = Cr_A^{-j}(m,n) \bullet b - Cr_A^{-j}(,m,n) \circ b \tag{2.3}$$

$$CB^{-j}(m,n) = Cb_A^{-j}(m,n) \bullet b - Cb_A^{-j}(,m,n) \circ b \tag{2.4}$$

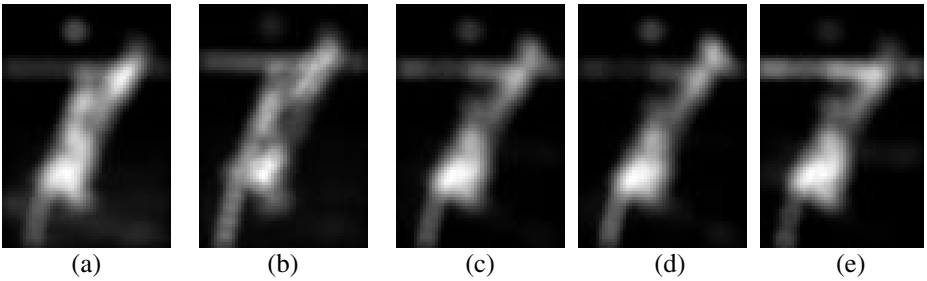
$$C^{-j} = CR^{-j} + CB^{-j} \tag{2.5}$$

In the above equations by $I^{-j}(m,n)$, $O^{-j}(m,n)$, $C^{-j}(m,n)$, we denote the intensity, orientation and colour feature maps computed at scale j while $\bullet$ and $\circ$ denote the closing and opening operators respectively.

The structuring element b is a disk of radius equal to *Jmax* where *Jmax* is maximum analysis depth and is computed as follows:

$$J\max = \left\lfloor \frac{1}{2}\log_2 N \right\rfloor,\ N = \min(R,C), \tag{3}$$

where in $y = \lfloor x \rfloor$ y is the highest integer value for which $x \geq y$, and R, C are the number of rows and columns of input image respectively.

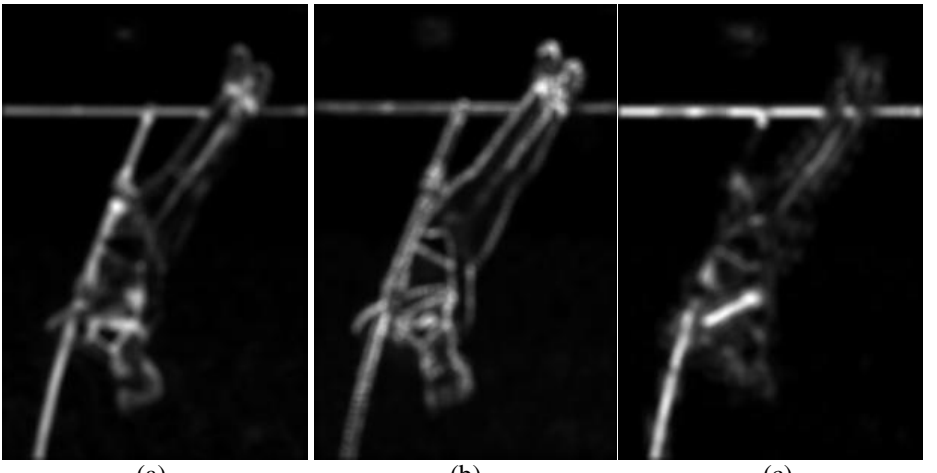


(a) (b) (c) (d) (e)

Fig. 1. Locally stand-out regions, at level 3, based on: (a) intensity, (b) orientation, and (c) colour. In (d) and (e) are shown the individual chromaticity feature maps (CR and CB).

Fig. 1 (a)-(c) shows the intensity, orientation and colour feature maps at scale 3 ($I^{-3}(m,n)$, $O^{-3}(m,n)$, $C^{-3}(m,n)$) along with the individual chromaticity feature maps ($CR^{-3}(m,n)$, $CB^{-3}(m,n)$) whose point by point addition produced the colour feature map.

In Fig. 2 (a)-(c) the intensity, orientation and colour feature maps at scale 1 are shown. It is important to note that the areas that stand-out from their surround are



(a) (b) (c)

Fig. 2. Locally stand-out regions, at level 1, based on: (a) intensity, (b) orientation, and (c) colour

significantly smaller (proportionally) than the ones shown in Fig. 1. Therefore, a combination of the features maps at the various scales (conspicuity maps) is needed to cover both small and large stand-out objects. Combination of different scales is achieved by interpolation to the finer scale, point-by-point subtraction and application of a saturate function to the final result. The following equations describe mathematically process of combining the results of two successive scales for the orientation conspicuity map. It is obvious that the same process applies also to intensity and colour conspicuity maps:

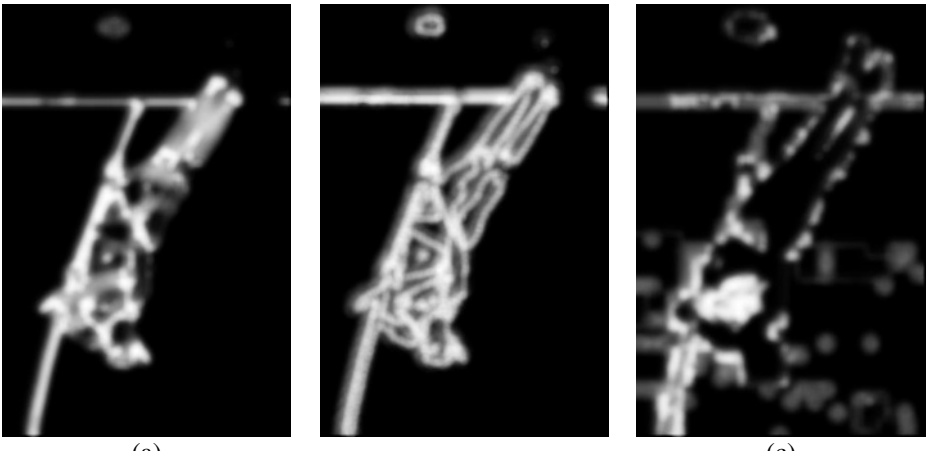
$$\hat{C}_O^{-j}(m,n) = \left(C_O^{-(j+1)}(m,n)) \uparrow^{2m} * h_\phi(m)\right) \uparrow^{2n} * h_\phi(n) \tag{4.1}$$

$$C_o^{-j}(m,n) = \frac{2}{1+e^{\left(-\left(\hat{C}_o^{-j}(m,n)+O^{-j}(m,n)\right)\right)}} - 1 \tag{4.2}$$

where $O^{-j}(m,n)$ is the orientation feature map computed at level j (see eq. 2.2), $C_o^{-j}(m,n)$ is the orientation conspicuity map at level j, $\hat{C}_o^{-j}(m,n)$ is the interpolation of $C_o^{-(j+1)}(m,n)$ at a finer scale j, and $\uparrow^{2m}$ and $\uparrow^{2n}$ denote up-sampling along rows and columns respectively.

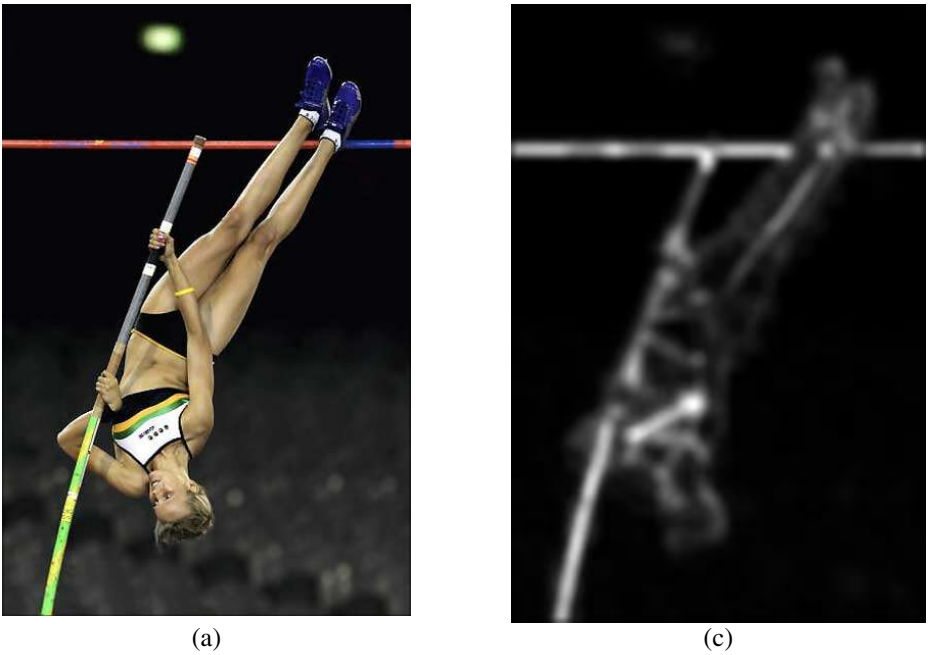
An example of intensity, orientation and colour conspicuity maps computed using analysis depth equal to 3 is shown in Fig.3.

After creating this intermediate multi-resolution representation (conspicuity maps per feature), where salient areas are enhanced and pop-out from the surround, an *across-scale combination* is applied to create a single *saliency* map. For this purpose a



(a) (c)

Fig. 3. Conspicuity maps computed using analysis depth (*Jmax*) equal to 3 (a) intensity (b) orientation, and (a) colour



(a) (c)

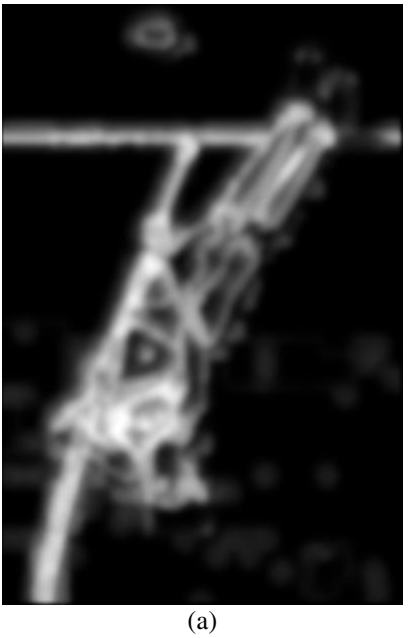
Fig. 4. (a) An input frame, (b) Saliency map computed at depth 1. Note that though the floodlight it is a clearly stand-out object it is not recognized as such at this level.

saturate function is applied so as to preserve the independency and added value of the particular concpicuity maps. This process is described mathematically by the following equation:

$$S(m,n) = \frac{2}{1+e^{\left(-\left(C_I^{-0}(m,n)+C_O^{-0}(m,n)+C_C^{-0}(m,n)\right)\right)}} - 1 \tag{5}$$

where $C_I^{-0}(m,n)$, $C_O^{-0}(m,n)$, and $C_C^{-0}(m,n)$ are the intensity, orientation and colour conspicuity maps respectively.

Figs. 4 and 5 show examples of saliency maps computed using depths 1, 3, and 4. In the latter case objects covering as much as 20% of the whole image can be identified as standing-out from their surround.



(a)

(c)

Fig. 5. Saliency maps computed using analysis depth (*Jmax*) equal to (a) 3 and (b) 4. In (a) the floodlight starts appearing as a stand-out object but the overall saliency maps is rather noisy. In (b) the floodlight it is clearly a stand-out object while the saliency map is smooth.

4 Visual Trials Tests and Experimental Results

To evaluate the algorithm, we simply use it as a front end; that is, once the VA-ROI areas identified the non-ROI areas in the video frames are blurred. Although this approach is not optimal in terms of expected file size gains, it has the advantage of producing compressed streams that are compatible with existing decoders [10].

Visual trial tests were conducted to examine the quality of the VA-ROI based encoded videos. These tests are based upon ten short video clips, namely: *eye_witness, fashion, grandma, justice, lecturer, news_cast1, news_cast2, night_interview, old_man, soldier* (see [11]). All video clips were chosen to have a reasonably varied content, whilst still containing humans and other objects that could be considered to be more important (visually interesting) than the background. They contain both indoor and outdoor scenes and can be considered as typical cases of news reports based on 3G video telephony. However, it is important to note that the selected video clips were chosen solely to judge the efficacy of VA ROI coding in MPEG-1 and are not actual video- telephony clips.

For each video clip encoding aiming at low-bit rate (frame resolution of 144x192, frame rate 24 fps, GOP structure: IBBPBBPBBPBB) has been taken place so as to conform to the constraints imposed by 3G video telephony. Two low-resolution video-clips were created for each case, one corresponding to VA based coding and the other to standard MPEG-1 video coding.

4.1 Experimental Methodology

The purpose of the visual trial test was to directly compare VA ROI based and standard MPEG-1 encoded video where the ROI is determined using the proposed VA algorithm. A two alternative forced choice (2AFC) methodology was selected because of its simplicity, i.e., the observer views the video clips and then selects the one preferred, and so there are no issues with scaling opinion scores between different observers [12]. There were ten observers, (5 male and 5 female) all with good, or corrected, vision and all observers were non-experts in image compression (students). The viewing distance was approximately 20 cm (i.e., a normal PDA / mobile phone viewing distance) and the video clip pairs were viewed one at a time in a random order.

The observer was free to view the video clips multiple times before making a decision within a time framework of 60 seconds. Each video pair was viewed twice, giving (10x10x2) 200 comparisons. Video-clips were viewed on a typical PDA display in a darkened room (i.e., daylight with drawn curtains). Prior to the start of the visual trial all observers were given a short period of training on the experiment and they were told to select the video clips they preferred assuming that it had been downloaded over a 3G mobile / wireless network.

4.2 Results

Table 1 shows the overall preferences, i.e., independent of (summed over) video clips for standard MPEG-1 and VA ROI-based encoded MPEG-1. It can be seen in that there is slight preference to standard MPEG-1 which is selected at 52.5% of the time as being of better quality. However, the difference in selections, between VA ROI-based and standard MPEG-1 encoding, is actually too small to indicate that the VA ROI-based encoding deteriorates significantly the quality of produced video. At the same time the bit rate gain, which is about 27% on average (see also Table II), shows clearly the efficiency of VA ROI based encoding.

Table 1. Overall preferences (independent of video clip)

Encoding Method	*Preferences*	*Average Bit Rate (Kbps)*
VA-ROI	95	224.4
Standard MPEG -1	105	308.1

Table 2. Comparison of VA-ROI based and Standard MPEG-1encoding in ten video seqs

Video Clip	*Encoding Method*	*Bit Rate (Kbps)*	*Bit Rate Gain*
Eye_witness,	VA-ROI Standard	319 386	17 (%)
fashion	VA-ROI Standard	296 354	16 (%)
grandma	VA-ROI Standard	217 256	15 (%)
justice	VA-ROI Standard	228 318	28 (%)
lecturer	VA-ROI Standard	201 274	27 (%)
news_cast1	VA-ROI Standard	205 297	31 (%)
news_cast2	VA-ROI Standard	170 270	37 (%)
night_interview	VA-ROI Standard	174 335	48 (%)
old_man	VA-ROI Standard	241 321	25 (%)
soldier	VA-ROI Standard	193 270	29 (%)
Average	VA-ROI Standard	224.4 308.1	27.2 (%)

Table 2 presents the bit-rates achieved for both the VA ROI based encoding and standard MPEG-1 in the individual video clips. It is clear that the bit rate gain obtained is significant, ranging from 15% to 48%. Furthermore, it can be seen from the results obtained in the *night_interview* video sequence, that increased bit-rate gain does not necessarily mean worse quality of the VA ROI encoded video.

Bit-rate gain achieved by JPEG encoding of the individual video frames (not shown in Table 2) is on average about 21% (ranging from 14% to 28%). This indicates that the bit-rate gain is mainly due to the compression obtained for Intra-coded (I) frames than for the Inter coded (P,B) ones. This conclusion strengthens the argument that smoothing of non-ROI areas may decrease the efficiency of motion compensation.

Acknowledgement. The study presented in this paper was supported (in part) by the research project “OPTOPOIHSH: Development of knowledge-based Visual Attention

models for Perceptual Video Coding", PLHRO 1104/01 funded by the Cyprus Research Promotion Foundation [13]

References

1. J.K. Tsotsos, S.M. Culhane, W.Y.K. Wai, Y. Lai, N. Davis, F. Nuflo, "Modeling visual attention via selective tuning", Artificial Intelligence, vol. 78, pp. 507-545, 1995.
2. E. Dickmanns, "Expectation-based dynamic scene understanding", in (eds.) Blake & Yuille, Active Vision, MIT Press, Cambridge Massachusetts, pp. 303-334.
3. Koch C., Ullman S., "Shifts in selective visual attention: towards the underlying neural circuitry", Human Neurobiology, vol. 4, pp. 219-227, 1985.
4. A. M. Treisman and G. Gelade, "A feature integration theory of attention," Cognitive Psychology, vol. 12(1), pp. 97-136, 1980.
5. Niebur, E. and Koch, C., "Computational architectures for attention" In Parasuraman, R.,editor, *The Attentive Brain*, chapter 9, pages 163–186. MIT Press, Cambridge, MA., 1998.
6. L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20(11), pp. 1254-1259, 1998.
7. L. Itti, and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," Vision Research, vol. 40, pp. 1489-1506, 2000.
8. R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, 2nd edition, Prentice Hall Inc, NJ, 2002, ISBN: 0-13-094650-8.
9. S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation", *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 11, pp 674-693, 1989.
10. Z. Wang, L. G. Lu, and A. C. Bovik, "Foveation scalable video coding with automatic fixation selection," *IEEE Transactions on Image Processing*, vol. 12, pp. 243–254, 2003.
11. [Online] http://www.cs.ucy.ac.cy/~nicolast/research/VAclips.zip
12. M. P. Eckert and A.P. Bradley, "Perceptual models applied to still image compression," *Signal Processing*, 70 (3), pp. 177–200, 1998.
13. [Online] http://www.optopiisi.com

Selective Tuning: Feature Binding Through Selective Attention

Albert L. Rothenstein and John K. Tsotsos

Dept. of Computer Science & Engineering and Centre for Vision Research
York University, Toronto, Canada
{albertlr, tsotsos}@cs.yorku.ca

Abstract. We present a biologically plausible computational model for solving the visual binding problem. The binding problem appears due to the distributed nature of visual processing in the primate brain, and the gradual loss of spatial information along the processing hierarchy. The model relies on the reentrant connections so ubiquitous in the primate brain to recover spatial information, and thus allow features represented in different parts of the brain to be integrated in a unitary conscious percept. We demonstrate the ability of the Selective Tuning (ST) model of visual attention [1] to recover spatial information, and based on this propose a general solution to the binding problem. The solution is demonstrated on two classic problems: recovery of form from motion and binding of shape and color. We also demonstrate how the method is able to handle difficult situations such as occlusions and transparency. The model is discussed in relation to recent results regarding the time course and processing sequence for form-from-motion in the primate visual system.

1 Introduction

Convergent evidence from many different kinds of studies suggests the visual cortex is divided into a large number of specialized areas processing different feature dimensions, organized into two main processing streams, a dorsal pathway, responsible for encoding motion, space, and spatial relations for guiding actions, and a ventral pathway, associated with object recognition and classification, conclusions supported by functional imaging, neurophysiology, and by strikingly selective localized lesions. This high selectivity of the various cortical areas has led to the obvious questions of how, despite this specialization, the visual percept is unitary, and what are the mechanisms responsible for, in effect, putting all this distributed information together. Following Roskies [2], "the canonical example of binding is the one suggested by Rosenblatt [3] in which one sort of visual feature, such as an object's shape, must be correctly associated with another feature, such as its location, to provide a unified representation of that object." Such explicit association is particularly important when more than one visual object is present, in order to avoid incorrect combinations of features belonging to different objects, otherwise known as "illusory conjunctions" [4]. Limiting the

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 548–557, 2006.

resources available for visual processing through increased loads and/or reduced time leads observers to erroneously associate basic features present in the image into objects that do not exist, e.g. a red X and a blue O are sometimes reported as a blue O and a red X. Studies have shown that these are real conjunction errors, and can not be attributed to guessing or memory. A general discussion of the binding problem appears in Neuron 24(1) (1999).

Three classes of solutions to the binding problem have been proposed in the literature. Proponents of the *convergence* solution suggest that highly selective, specialized neurons that explicitly code each percept (introduced as cardinal cells by Barlow [5] – also known as gnostic or grandmother neurons) form the basis of binding (e.g. [6,7]). The main problem with this solution is the combinatorial explosion in the number of units needed to represent all the different possible stimuli. Also, while this solution might be able to detect conjunctions of features in a biologically plausible network (i.e. a multi-layer hierarchy with pyramidal abstraction) it is unable to localize them in space on its own [8], and additional mechanisms are required to recover location information. *Synchrony*, the correlated firing of neurons, has also been proposed as a solution for the binding problem [9,10,11]. Synchrony might be necessary for signaling binding, but is not sufficient by itself, as it is clear that this phenomenon can at most tag bound representations, but not perform the binding process. The *colocation* solution proposed in the Feature Integration Theory (FIT) [12] simply states that features occupying the same spatial location belong together. Due to its purely spatial nature, this solution can not deal with transparency and other forms of spatial overlap. Also, since detailed spatial information is only available in the early areas of the visual system, simple location-based binding is agnostic of high-level image structure, which means that it can not impose boundaries (obviously, the different edges of an object occupy different spatial locations), and arbitrary areas that belong to none, one or more objects can be selected.

Selective Tuning (ST) [1] is a computational model of visual attention that integrates feedforward and feedback pathways into a network that is able to take high level decisions, and, through a series of winner-take-all (WTA) processes, identify all the neurons that have participated in that decision. This identification satisfies the key requirement for a kind of visual feature binding that ST was demonstrated to solve [13], despite the loss of spatial information inherent in a pyramidal system. The ST feedback process does not need collocation if neural convergence is guaranteed, so ST is able to select all parts of a stimulus, even if they do not share a location (e.g. stimuli with discontinuities due to overlap, or stimuli that are separated spatially due to the nature of the cortical feature maps). The partial solution to binding proposed in [13] is able to correctly bind all the activations that have contributed to a high level decision (*convergence*) and even non-convergent representations if the problem can be solved at the spatial resolution of the top level of the pyramid (a weak form of *collocation*) – i.e. there is sufficient spatial separation between the target and the distractors (see [14] for the importance of spatial separation in attention and recognition). Note that the feedback process will select only the units responding to the

selected stimulus, and not units that just happen to share locations with it, thus ensuring that overlapping and transparent stimuli will be handled correctly.

2 Proposed Solution

This section motivates and introduces an original approach to the binding problem. FIT [12] considers location as a feature that is faithfully represented in a "master map" of locations but, as Taraborelli [15] points out: "the idea of binding itself is nothing but a spatial conjunction of information concerning visual attributes of the same item." Tsotsos et al. [13] note that considering location as a feature can not be valid as location precision (resolution) changes layer to layer in any pyramid representation, and propose that location should be considered as the anchor that permits features to be bound together. At the same time, Robertson lists three phenomena that demonstrate the special role of spatial attention in binding [16]: illusory conjunctions under divided attention, dependence on number of distractors for conjunction searches, and the elimination of the influence of distractors with spatial cueing. In effect, a solution to the binding problem must address this seemingly incompatible requirement: binding is ultimately only a spatial conjunction, but at the same time it must be based on high-level information, allowing for object and feature-based selection.

The solution proposed is based on the general *collocation* principle of FIT, but using Selective Tuning to integrate high-level representations in the spatial selection process, and performing the spatial selection without requiring a "master map" of locations. The proposal, illustrated in Fig. 1 is to allow one feedforward pass through the network (arrow A in the figure), detect and select one salient high-level representation (in this case, one motion representation), and proceed backwards through the system in Selective Tuning manner (arrow B), selecting compatible representations that have contributed to the winning units, and inhibiting all the activations that are incompatible. As this feedback proceeds, lower level representations that participated in the salient activation are selected, and distractors inhibited, all the way to the first layer of processing. This allows further feedforward processing to be selectively directed towards the selected object (arrow C), eliminating the influence of spatially near competing stimuli and allowing the ventral pathway to detect the shape corresponding to the motion signal. When processing ends, the remaining active high-level representations all describe the selected stimulus in a sparse, distributed fashion ideal for maximizing the capacity of associative memories [17]. At the same time all the components of the attended stimulus will be selected throughout the visual system for recognition, and the location information can be used for the planning of actions towards the selected stimulus.

3 Examples

The general structure of the neural network used in the following examples is presented in Fig. 1, consisting of two biologically inspired processing pathways,

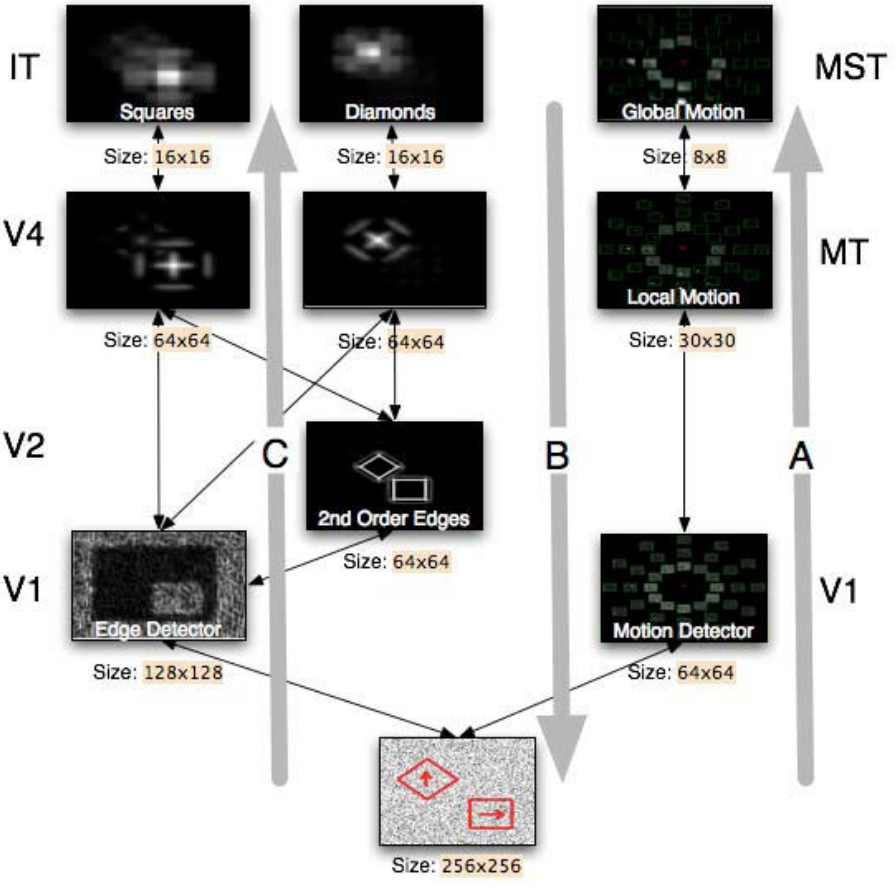
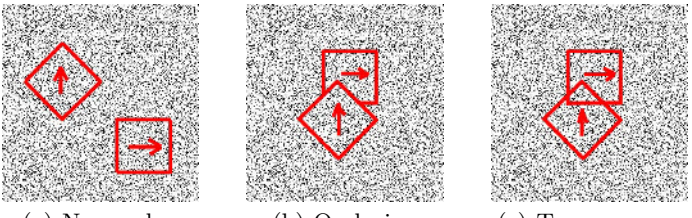


Fig. 1. Diagram of the network. On the left side is the shape recognition pathway, while on the right side the motion pathway. The arrows show the flow of information. See the text for details.

corresponding to the ventral and dorsal visual pathways. The pathways are pyramidal, meaning that successive layers represent more and more abstract concepts, and location, size and (direct) shape information is lost. The motion pathway recognizes affine motions, and is described in detail in [13]. The form pathway is an abstraction of the primate object recognition stream, and consists of layers that combine the edge information to detect simple geometric shapes.

As shape and motion are processed in different areas of the brain, the recovery of shape from motion is a particularly good illustration of binding. The subset of the motion processing hierarchy in Fig. 1 consists of a layer of motion sensitive neurons, followed by two layers of translation detection neurons, corresponding to visual areas V1, MT (local motion) and MST (global motion), respectively. The simplified shape processing hierarchy, detecting square and diamond shapes,

consists of two layers of first and second order edge detectors (four directions, one scale), and two layers of shape detectors, corresponding to visual areas V1 and V2 (edges), V4 (local shape) and IT (global shape), respectively. All the weights in the neural network are preset, no learning was used. Our system will process the image in parallel, along all the independent processing pathways, detecting the presence of the different shapes and motion patterns. The attentional process will select one top-level representation for further analysis, and the ST process will localize the corresponding pixels in the input image through feedback. ST will also inhibit pixels in the surround of the attended item, thus enhancing the relative saliency of the attended stimulus and introducing the contour information needed by the shape pathway. A second feedforward pass through the pyramids will refine the representation of the selected object, and at the same time select all the (distributed) representations that belong to it, thus achieving binding. The process can be repeated until the desired target is found, implementing a visual search mechanism. In the following experiments, the stimuli consists of random dot kinematograms with the dots in one or two windows performing translation motion in one or two different directions. The window can be square or diamond shaped – Fig. 2.



(a) No overlap (b) Occlusion (c) Transparency

Fig. 2. Random dot kinematograms used as stimuli in the form-from-motion experiments. Dots in one or two windows perform translation motion. The window can be square or diamond shaped.

3.1 Example 1 – Form from Motion

The stimulus consists of a random dot kinematogram with the dots in a square window performing translation motion to the right. After the initial feedforward and feedback passes, the moving dots will be localized in the input layer, and neighboring dots will be inhibited, as illustrated in Fig. 3(a). Once the neighboring neurons have been inhibited, the V1 edge detectors will detect these pseudo-edges, and the shape recognition pathway will become activated, detecting the presence of the square in the input sequence. Similarly, Fig. 3(b) illustrates the result of detecting motion in a diamond shaped window. To test the capabilities of the attentional system to deal with multiple input patterns we used an image sequence containing two moving regions: a square region of rightward moving dots and a diamond shaped region of upward moving dots. MST neurons will fire indicating the two incompatible motion patterns, and the attentional system

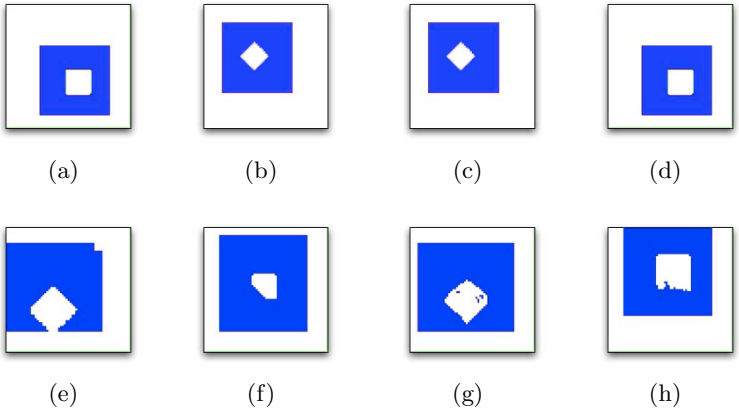


Fig. 3. Attentional selection windows for random-dot kinematograms. (a) Single stimulus, Square (b) Single stimulus, Diamond (c) Two objects, fixation 1 (d) Two objects, fixation 2 (e) Occluder, fixation 1 (f) Occluded, fixation 2 (g) Transparent, fixation 1 (h) Transparent, fixation 2. The internal white area represents the localized stimulus, blue (dark) the inhibitory surround.

will select the top level movements one after the other, allowing the system to detect first one, then the other shape – Fig. 3(c) and Fig. 3(d).

An important test that must be passed by any artificial vision system is its handling of occlusions and transparency. The two stimulus sequences from Example 1 were modified by making the two stimuli overlap spatially, either through occlusion or through transparency.

3.2 Example 2 – Occlusion

In this case, the diamond partially occludes the square. The process follows as illustrated above for the occluder – Fig. 3(e), but it is important to observe that for the occluded stimulus only the visible portion is selected Fig. 3(f). This is a very important point that highlights a key difference between ST and the Neocognitron [18] system. While in both systems, feedback "tunes" the processing pyramid, the latter increases the activation of units that correspond to the selected hypothesis by reducing their firing thresholds, thus introducing the risk of hallucinations. In ST the only manipulation permitted is the reduction of the activations of units that do not match the hypothesis, and so the risk of hallucination is eliminated, and if the hypothesis turns out to be incorrect, the responses of the selected output units should decrease, indicating the failure to find support for the hypothesis. This example clearly shows that based on the detected motion patterns, the object recognition pathway will get as input a correct representation of the shape present in the stimulus.

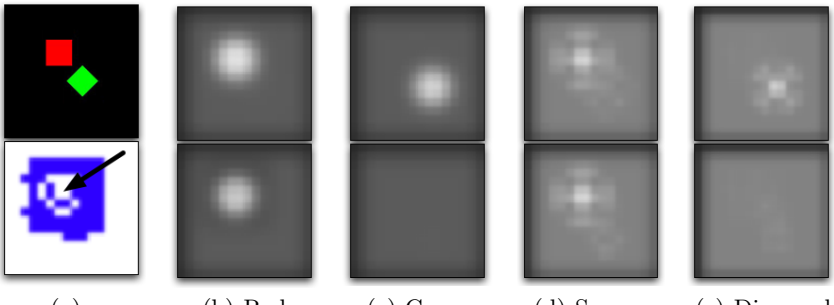
3.3 Example 3 – Transparency

In this case, the square partially overlaps the diamond, but the moving dots in the diamond shaped window remain visible. Again, the high level detection combined with the ST feedback process are able to highlight the correct areas in the input, thus allowing the object recognition pathway to correctly recognize the shapes – Fig. 3(g) and 3(h). While not an issue in this example, note that in some cases humans perceive two overlapping and different motion patterns as a single motion in a third direction (e.g. plaid motion), functionality currently missing from our motion model. Due to the fact that the ST process selects all inputs that have contributed to a high level decision as belonging to the stimulus, we expect the system to function correctly once this functionality is added.

3.4 Example 4 – Binding Color and Shape

The binding problem is often illustrated with images consisting of different geometric shapes, each of a different color [4]. Similar to the previous examples, we will use red square and green diamond objects (see Fig. 4(a) top), and we add a color detection pyramid (simple Gaussian blurring and downsampling). The system will initially detect the presence of the different shapes and colors by processing the whole image in parallel – Figs. 4(b) - 4(e) top. The attentional WTA process will select one top-level representation (the red representation, in this case), the ST process will localize the corresponding pixels in the input image and inhibit all nearby pixels, thus enhancing the relative saliency of the attended stimulus.

A second pass through the pyramids will refine the representation of the selected object, and select all the (distributed) representations that belong to it, while the green and diamond representations are strongly inhibited, thus achieving binding – Figs. 4(b) - 4(e) bottom. Fig. 4(a) bottom represents the difference between the activation of the red detector with and without attention, and it can be observed that in the attended condition the representation that was initially distributed (the dark inhibited area) is much more focused, as indicated by the arrow. The initial representation corresponded to all the items in the



(a) (b) Red (c) Green (d) Square (e) Diamond

Fig. 4. Effects of attentional selection on colored shape stimuli. See text for details.

field, but as a result of attention only the representations corresponding to red square remain active, making binding possible.

4 Discussion

While the importance of space in binding is captured in the Feature Integration Theory, high level representations, object- and feature-based attention mechanisms are not easily integrated into FIT. In this paper we have presented an original solution to the binding problem in visual perception, by recovering spatial information from high level representations through Selective Tuning style feedback. Another important contribution of this research is a process of recovering spatial information that does not require a "master map" of locations or any other separate representation of space. We have demonstrated this solution through a number of examples, including the difficult cases of occlusion and transparency. While these preliminary results are encouraging, the representations used (especially the shape recognition pathway) are very simplistic, and significant work needs to be done to prove the generality of the solution. It is important to observe that the mechanisms employed are very general, and could potentially be applied in the context of very different object recognition schemes, including structural and view based, as long as they have a multi-layer hierarchy with pyramidal abstraction structure (e.g. [19,20]).

A recent study regarding the time course and processing sequence of form-from-motion in humans using similar stimuli [21] concludes that dorsal activation (area V5/MT) precedes ventral activation (areas LO and IT) by 50-60ms. This time interval is consistent with the proposed mechanism [22]. The strongest indications about the nature of the internal representations used in object/shape perception comes from a series of imaging and neurophysiology experiments [23]. These results consistently show that for each in a very broad categories of stimuli, a small number of regions in IT become active, pointing to a sparse coarse coding. In order to solve the binding problem for object recognition, the various areas of activity corresponding to one stimulus must be selected together [13], but this is very difficult if not impossible at the level of IT due to the loss of spatial information. If, as indicated by recent studies (e.g. [24,25]), categoric information is available early in visual processing, the holistic nature of this representation might be able to act as anchor for the mechanism proposed in this paper, and select all the individual activations, while at the same time inhibiting competing representations, thus increasing the signal-to-noise ratio of the selected stimulus. The idea that attention binds together distributed cortical activations at multiple levels of the visual hierarchy involved in processing attended stimuli has recently received significant experimental support [26], and reentrant connections between extrastriate areas and V1 are gaining support as the substrate for attention and conscious perception – see [27] for a review.

Object recognition is one of the most important problems in computer vision, and in this context probably the biggest challenge is the representational gap between low-level image features and high-level concepts such as generic

models [28]. The proposal presented here allows a system to extract intermediate level representations based on available/extractable information and perceptual grouping, and use the feedback process to refine and bind together those intermediate level representations that belong to the same object. This would create a distributed sparse representation similar to that present in IT, representation known to be optimal for maximizing the capacity of associative memory networks [17]. Given the representational gap that computer vision systems must bridge, this method has the potential to make important contributions to the field.

References

1. Tsotsos, J.K., Culhane, S.M., Wai, W.Y.K., Lai, Y.H., Davis, N., Nuflo, F.: Modeling visual-attention via selective tuning. Artif. Intell. **78**(1-2) (1995) 507–545
2. Roskies, A.L.: The binding problem. Neuron **24**(1) (1999) 7–9
3. Rosenblatt, F.: Principles of Neurodynamics: Perceptions and the Theory of Brain Mechanisms. Spartan Books (1961)
4. Treisman, A., Schmidt, H.: Illusory conjunctions in the perception of objects. Cognit Psychol **14**(1) (1982) 107–41
5. Barlow, H.B.: Single units and sensation: A neuron doctrine for perceptual psychology? Perception **1**(4) (1972) 371–394
6. Ghose, G.M., Maunsell, J.: Specialized representations in visual cortex: a role for binding? Neuron **24**(1) (1999) 79–85
7. von der Malsburg, C.: The what and why of binding: the modeler's perspective. Neuron **24**(1) (1999) 95–104
8. Rothenstein, A.L., Tsotsos, J.K.: Attention links sensing to recognition. Image and Vision Computing (in press doi:10.1016/j.imavis.2005.08.011) (2006)
9. Milner, P.: A model for visual shape recognition. Psychol. Rev. **81** (1974) 521–535
10. von der Malsburg, C.: Nervous structures with dynamical links. Ber. Bunsenges. Phys. Chem. **89** (1985) 703–710
11. Singer, W.: Neuronal synchrony: a versatile code for the definition of relations? Neuron **24** (1999) 49–65
12. Treisman, A.M., Gelade, G.: Feature-integration theory of attention. Cognitive Psychology **12**(1) (1980) 97–136
13. Tsotsos, J.K., Liu, Y., Martinez-Trujillo, J.C., Pomplun, M., Simine, E., Zhou, K.: Attending to visual motion. Comput. Vis. Image Und. **100**(1-2) (2005) 3–40
14. Cutzu, F., Tsotsos, J.K.: The selective tuning model of attention: psychophysical evidence for a suppressive annulus around an attended item. Vision Research **43**(2) (2003) 205–219
15. Taraborelli, D.: Feature binding and object perception. Does object awareness require feature conjunction? In: 10th Annual Meeting of the European Society for Philosophy and Psychology - ESPP 2002, Lyon. (2002)
16. Robertson, L.: Space, Objects, Brains and Minds. Essays in Cognitive Psychology. Psychology Press (2004)
17. Foldiak, P., Young, M.: Sparse coding in the primate cortex. In Arbib, M.A., ed.: The Handbook of Brain Theory and Neural Networks. MIT Press (1995) 895–898
18. Fukushima, K., Imagawa, T., Ashida, E.: Character recognition with selective attention. In: International Joint Conference on Neural Networks. Volume 1., Seattle (1991) 593–598

19. Hummel, J.E., Stankiewicz, B.J.: An architecture for rapid, hierarchical structural description. In Inui, T., McClelland, J., eds.: Attention and Performance XVI: Information Integration in Perception and Communication. MIT Press. (1996) 93–121
20. Riesenhuber, M., Poggio, T.: Hierarchical models of object recognition in cortex. Nature Neuroscience **2**(11) (1999) 1019–1025
21. Schoenfeld, M.A., Woldorff, M., Duzel, E., Scheich, H., Heinze, H.J., Mangun, G.R.: Form-from-motion: MEG evidence for time course and processing sequence meg evidence for time course and processing sequence. Journal of Cognitive Neuroscience **15**(2) (2003) 157–172
22. Bullier, J.: Integrated model of visual processing. Brain Research Reviews **36**(2-3) (2001) 96–107
23. Tsunoda, K., Yamane, Y., Nishizaki, M., Tanifuji, M.: Complex objects are represented in macaque inferotemporal cortex by the combination of feature columns. Nature Neuroscience **4** (2001) 832–838
24. Grill-Spector, K., Kanwisher, N.: Visual recognition: as soon as you see it, you know what it is. Psychological Science **16**(2) (2005) 152–160
25. Wolfe, J.M.: Moving towards solutions to some enduring controversies in visual search. Trends in Cognitive Sciences **7**(2) (2003) 70–76
26. Haynes, J.D., Tregellas, J., Rees, G.: Attentional integration between anatomically distinct stimulus representations in early visual cortex. Proc. Natl. Acad. Sci. USA **102**(41) (2005) 14925–30
27. Pollen, D.A.: Explicit neural representations, recursive neural networks and conscious visual perception. Cerebral Cortex **13**(8) (2003) 807–14
28. Keselman, Y., Dickinson, S.J.: Generic model abstraction from examples. IEEE T. Pattern Anal. **27**(7) (2005) 1141 – 1156

Rotation Invariant Recognition of Road Signs with Ensemble of 1-NN Neural Classifiers

Bogusław Cyganek

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

Abstract. The paper presents a parallel system of two compound classifiers for recognition of the circular shape road signs. Each of the two classifiers is built of an ensemble of 1-nearest-neighbour (1-NN) classifiers and the arbitration unit operating in the winner-takes-all mode. For the 1-NN we employed the Hamming neural network (HNN) which accepts the binary input. Each HNN is responsible for classification within a single group of deformable prototypes of the road signs. Each of the two compound classifiers has the same structure, however they accept features from different domains: the spatial and the log-polar spaces. The former has an ability of precise classification for shifted but non-rotated objects. The latter exhibits good abilities to register the rotated shapes and also to reject the non road sign objects due to its high false negative detection properties. The combination of the two outperformed each of the single versions what was verified experimentally. The system is characterized by fast learning and recognition rates.

1 Introduction

In this paper we present our approach to the recognition problem of simple planar objects encountered in natural scenes. Our objective was to build a road sign (RS) classification system for circular shape signs based on the a priori prototypes given in the printed regulation (or norm) exclusively. Thus we do not assume any additional training patterns; specifically we do not use real examples to train our classifiers. This assumption has its advantages and drawbacks at the same time: on the one hand it is very appealing since the same approach can be used in many different situations, e.g. we can easily exchange the input data base of patterns with the rest of the system untouched and without any need of tedious gathering of real examples and the training. On the other hand, we may not be able to recognize some objects if they differ significantly from their printed prototypes. However, we noticed that humans can do such operation with no difficulty – one can simply read and memorize a page of symbols and then recognize these patterns in natural scenes with no or negligible mistakes. Although colour information helps in many situations, we assume that the signs can be recognized solely from the grey valued images of the natural scenes. The motivation behind this assumption comes also from the fact that humans can perform such recognition without any problems. The second reason is simplification of the

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 558 – 567, 2006.

processing path when dealing with scalar valued images only. However, colour plays a very important role in our system during RS detection from a natural scene [5].

Automatic detection and recognition of road signs have found much attention in the literature. For review one can refer to [4][7][10][17]. Usually the subject is divided into shape detection [3][5][10][4], then followed by classification, although a combined systems are also possible [2]. The back-propagation NN is proposed in [3][10]. The more robust classifier, which to some degree is resistive to occlusions and small shape rotations, is based on the Kohonen NN [3]. The other NNs which have been used for the sign recognition are the receptive field NNs [15], NNs with the radial basis functions [19], and the adaptive resonance ART NN [9].

The prototype based nearest-neighbours methods have been used with success in many classification systems since 1950s [12]. The great advantage of this class of methods comes from the potential incorporation of invariances under some transformation of input patterns. This is achieved by design of specific invariant manifolds in the pattern space and a metric used to measure the distances between elements in that space. Then, instead of Euclidean distance the distance of tangents is computed between feature points. Such approach, called a tangent distance, was devised by Simard for recognition of hand written digits [12].

In this paper we focus on the circular shaped RS from the group "B" and "C" of the Polish regulation system [18]. The other groups ("A" and "D") of RS were addressed in previous publications [6]. The main difference between these groups is that triangular "A" and rectangular "D" signs can be registered by the detection module to the common base line and therefore can have only small vertical or horizontal shifts with which we can deal quite easily. On the other hand, the circular "B" and "C" signs can be rotated by small angles, a phenomenon which is quite often encountered in natural scenes. Thus we had to develop different means of preprocessing to deal with small rotations. In our approach we employ two different systems that can cope with shifted and rotated patterns. The first one operates in the spatial domain as described in [6]. The second one operates after the log-polar transformation which alters rotation and scale into translations in the output space [20]. The data base of our RS prototypes is transformed into log-polar representation with origin placed at a centre of gravity computed on grey valued signal. Then the log-polar representation is binarized. Finally, the binary representation is shifted to produce deformable models representing small variations of angle and scale. By this we implement the idea of "hints" proposed in [1].

Based on the assumption of a single prototype for each pattern we decided to use the 1-nearest-neighbour approach for classification [12][8]. In this class of methods the training prototypes can be seen as points in the feature space. Each prototype point has associated a label; When a query pattern is put into the classifier it responds with the closest pattern in the sense of employed metric in that space. Each expert classifier in the system is responsible for recognition of all road signs from the group, however under a single deformation.

Since we have only one prototype per pattern, we expect some degree of generalization from our experts. In our system they are implemented as Hamming neural networks that efficiently compute the Hamming distance among binary patterns. The expert classifiers are then formed into a committee machine orchestrated by the winner-takes-all rule.

2 Feature Collection

Fig. 1. presents the stages of feature selection for the classifiers. There are two paths:

1. The binarized features from the original input space;
2. The binarized features from the log-polar representation of an input image;

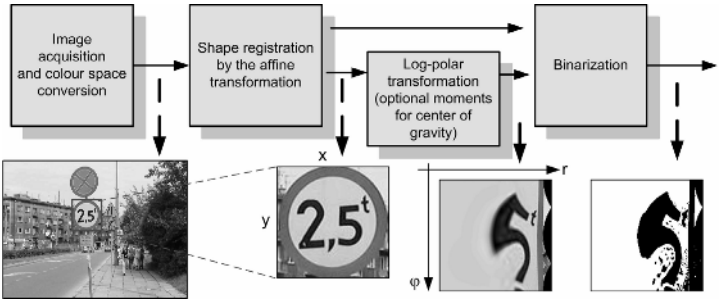


Fig. 1. Feature selection path in our system

The shape registration is performed by affine image transformation, as follows:

$$\mathbf{Ax} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ 1 \end{bmatrix} = \hat{\mathbf{x}} \tag{1}$$

where $x=(x_1,x_2,x_3)^T$ is a point in homogeneous coordinate system, $\mathbf{A}$ is the warping(affine) matrix, and $\hat{\mathbf{x}}$ denotes a point x after the warping. A value at x is determined by a bilinear interpolation, while matrix $\mathbf{A}$ by solving the linear system of equations composed from at least three non co-linear points. For circular shaped objects these are three corner points of a rectangle delimiting that shape – see Fig. 2.

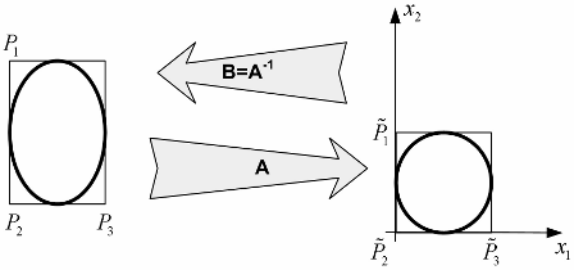


Fig. 2. Registration of the circular shapes by an affine mapping and interpolation

The points $\mathbf{P_1}$-$\mathbf{P_3}$ are mapped to $\tilde{\mathbf{P}}_1$ - $\tilde{\mathbf{P}}_3$ as follows:

$$\begin{cases} \mathbf{AP}_1 = \tilde{\mathbf{P}}_1 \\ \mathbf{AP}_2 = \tilde{\mathbf{P}}_2 \\ \mathbf{AP}_3 = \tilde{\mathbf{P}}_3 \end{cases} \qquad \begin{cases} \mathbf{P}_1 = \mathbf{B}\tilde{\mathbf{P}}_1 \\ \mathbf{P}_2 = \mathbf{B}\tilde{\mathbf{P}}_2 \\ \mathbf{P}_3 = \mathbf{B}\tilde{\mathbf{P}}_3 \end{cases} \tag{2}$$

where $\mathbf{B}=\mathbf{A}^{-1}$. The mapping with the matrix $\mathbf{B}$ is more practical since it allows the inverse warping scheme. To find $\mathbf{B}$ we need at least three pairs of matched points:

$$\begin{bmatrix} \tilde{p}_{11} & \tilde{p}_{12} & \tilde{p}_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \tilde{p}_{11} & \tilde{p}_{12} & \tilde{p}_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \tilde{p}_{11} & \tilde{p}_{12} & \tilde{p}_{13} \\ \tilde{p}_{21} & \tilde{p}_{22} & \tilde{p}_{23} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \tilde{p}_{21} & \tilde{p}_{22} & \tilde{p}_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \tilde{p}_{21} & \tilde{p}_{22} & \tilde{p}_{23} \\ \tilde{p}_{31} & \tilde{p}_{32} & \tilde{p}_{33} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \tilde{p}_{31} & \tilde{p}_{32} & \tilde{p}_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \tilde{p}_{31} & \tilde{p}_{32} & \tilde{p}_{33} \end{bmatrix} \begin{bmatrix} b_{11} \\ b_{12} \\ b_{13} \\ b_{21} \\ b_{22} \\ b_{23} \\ b_{31} \\ b_{32} \\ b_{33} \end{bmatrix} = \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{31} \\ p_{32} \\ p_{33} \end{bmatrix} \tag{3}$$

where b_{ij} are elements of $\mathbf{B}$. The above can be written as follows:

$$\tilde{\mathbf{P}}\mathbf{b} = \mathbf{P} \tag{4}$$

where $\mathbf{b}_{9\times1}$ contains aligned elements of $\mathbf{B}$, $\tilde{\mathbf{P}}_{9\times9}$ and $\mathbf{P}_{9\times1}$ are given in (3).

The log-polar transformation is given by the following equations [20]:

$$r = \log_L \left(\sqrt{(x-x_0)^2 + (y-y_0)^2} \right), \tag{5}$$

$$\varphi = \arctan \frac{y-y_0}{x-x_0}, \quad for \quad x \neq x_0 \tag{6}$$

for a point (x,y), where $C=(x_0,y_0)$ is a centre of transformation, L denotes base of a logarithm – it can be any positive value different from 1.0, however in practice it is chosen as to fit the maximum value of r_{max} to the maximal distance from the centre C and any point in the input image.

In our system we used an inverse warping scheme when transforming to the log-polar representation. In this scheme for a point in the output image a point in the input is computed and the value is interpolated (we used a bilinear interpolation). So, we had to use an inverse log-polar transformation given as follows:

$$x = \sqrt{\frac{L^{2r}}{1+tg^2\varphi}} + x_0 \quad , \quad y = \sqrt{\frac{L^{2r}}{1+tg^2\varphi}} tg\varphi + y_0 \ . \tag{7}$$

As alluded to previously, L is chosen as follows:

$$L = \exp\left(\frac{\ln(d_{\max})}{2r_{\max}} \right). \tag{8}$$

where $d_{max}=(x_{max}-x_0)^2+(y_{max}-y_0)^2$ is the maximal distance of a point from the centre. In our system we set the same size of an input image as well as its log-polar representation. In the latter the values of r and φ are quantized appropriately. In our experiments the centre C for the log-polar transformation is found as a centre of gravity $C=(x_0,y_0)=(m_{10}/m_{00}, m_{01}/m_{00})$ where m_{10}, m_{01}, and m_{00} are 2D moments [16]. Fig. 3. depicts relation of a rotation in the original space versus vertical image translation in the log-polar space on the example of the *B-13a* sign of resolution 164×164.

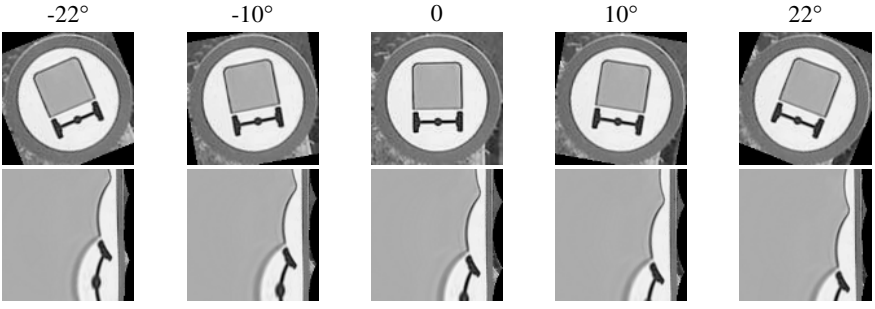


Fig. 3. Rotation in the input space is translation (vertical shift) in the log-polar space

The binarization process is carried out in the direct and log-polar spaces. However, it reflects intensity distribution of the original image. The binarization method must be the same for the whole class of images since for an input pattern we don't know beforehand what class it belongs to. In experiments the good results for the "B" group were achieved when binarization led to partitioning into white/non-white areas. In our experiments we found out that the best results are obtained with the following binarization methods:

1. Binarization around the median intensity value for the log-polar space;
2. Binarization around the mean intensity value for the (direct) feature space.

The sampling of the binarized version of a sign is performed line by line with a predefined horizontal and vertical margins (for 64×64 signs the margins are 4 and 4).

3 The 1-Nearest Neighbour Neural Classifier

The prototype and nearest-neighbours methods belong to the very common group of model-free classifiers [12]. These are called also memory-based methods since all the prototypes have to be stored prior to the recognition process. The classification is then based on finding the closest prototypes and majority voting scheme.

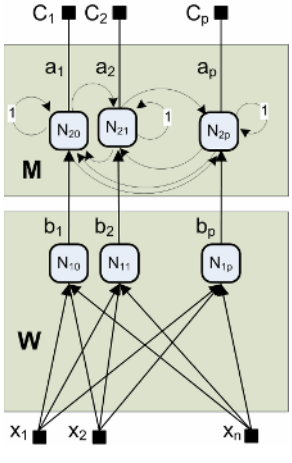


Fig. 4. The binary (Hamming) neural network as a classifier-expert module

For the binary inputs the very practical elementary single classifier showed to be the Hamming NN (HNN) [14]. In our system the HNN in Fig. 4. constitutes *the basic building block* of more complex classifiers. Each instance of HNN is designated for classification within *a single group of deformations* of the reference prototypes [6]. Training of the input layer consists of copying reference patterns into the weights of the matrix $\mathbf{W}_{pn}$ in Fig. 4. :

$$\mathbf{w}_i = \mathbf{x}_i \quad , \quad 1 \le i \le p \tag{9}$$

where p is the number of input patterns-vectors $\mathbf{x}$, each of the same length n, w_i is the *i-th* row of the matrix $\mathbf{W}$ of dimensions p rows and n columns. The recursive layer $N_{2.}$ (Fig. 4) selects a winning neuron. $\mathbf{M}_{pp}$ is initialized with negative values

$$m_{kl} = \varepsilon_k(t) = -(p-t)^{-1} \quad for \quad k \neq l, \quad 1 \quad for \quad k = l, \quad where \quad 1 \le k,l \le p, \quad p > 1 \tag{10}$$

where t is a classification time step, except the main diagonal which is set to 1.0 [11].

During the classification phase, the layer $N_{1.}$ of neurons computes the Hamming distance between the input pattern $\mathbf{z}$ and the training patterns, as follows:

$$b_i(\mathbf{z}, \mathbf{W}) = 1 - n^{-1} D_H(\mathbf{z}, \mathbf{w_i}) \quad , \quad 1 \le i \le p \tag{11}$$

where $b_i \in [0,1]$ is a value of an *i-th* neuron in the $N_{1.}$ layer, $D_H(\mathbf{z},\mathbf{w}_i) \in \{0,1,...,n\}$ is the Hamming distance. The $N_{2.}$ layer selects *only one* winning neuron in the process:

$$a_i[t+1] = \varphi\left(\sum_{j=1}^{n} m_{ij} a_j[t]\right) = \varphi\left(a_i[t] + \sum_{j=1, i \neq j}^{n} m_{ij} a_j[t]\right) \tag{12}$$

where $a_i[t]$ is an output of the *i-th* neuron of the $N_{2.}$ layer at the iteration step t, and $\varphi(x)=x$ for positive x and takes on *0* elsewhere.

The same winner-takes-all strategy is applied also in the arbitration unit to select a winning classifier (see Fig. 6). Some modifications are presented in [6].

4 The Combined Classifier for the Road Sign Classification

During experiments we noticed that classification exclusively in the log-polar space with binary inputs does not show the acceptable level of precision. This is caused by a fact of nonlinearity near the centroid of the processed sign (i.e. for small values of r) and also by the binarization.

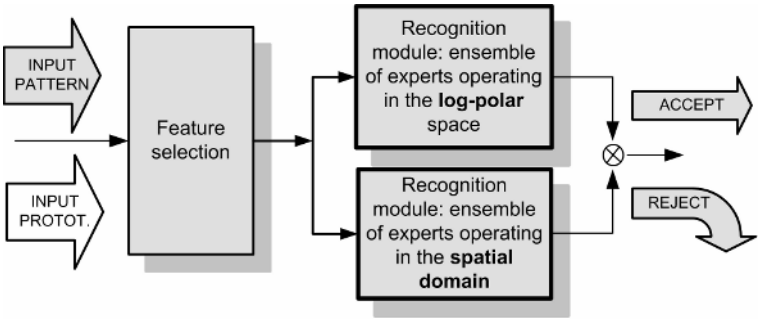


Fig. 5. Architecture of the system. Two ensembles of classifiers: log-polar and spatial domains.

The problem could be alleviated if the prototypes and test images had enough resolution. However, in such a case the number of features grows rapidly (a curse of dimensionality) what is not acceptable for the recognition with deformable models.

The log-polar classifier exhibits many desirable properties, as for instance low level of false negative matches [13].

On the other hand, the part of the system operating directly in the spatial domain also exhibits number of limitations [6]. Therefore we decided to build a combined system of classifiers, composed of the log-polar and the spatial domain classifiers operating in parallel – Fig. 5. A pattern is properly recognized if and only if the two recognition modules give *the same* answer. Otherwise a pattern is rejected.

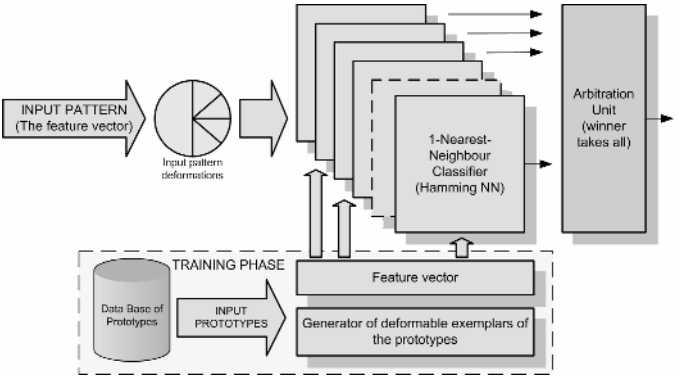


Fig. 6. Recognition module. Deformable prototypes are generated during the training phase.

The practical advantage of this system is that the two compound classifiers (ensembles of experts in Fig. 5,) have exactly the same structure which is depicted in Fig. 6. The only one difference is the processing space. The same is also the scheme of generation of deformable patterns from the reference prototypes: In both cases we generate only horizontally and vertically shifted versions of a pattern. For log-polar space transformations horizontal shifts denote rotation in the spatial space, while vertical – change of scale.

To cope with all possible deformations that a road sign can experience in real scenes we would need to generate deformations with three different parameters (two shifts plus rotation). However, this would result in excessive number of classifiers (experts) in the recognition modules. To avoid such situation we employ the serial-parallel approach for generation of the deformable patterns. Thus, deformations of two parameters are embedded into the ensemble of classifiers whereas the third one controls sequential deformations of the input pattern. For the recognition module operating in the log-polar space the embedded are shifts of the log-polar patterns (this is equivalent to variations of the angle and scale) while sequentially are generated minor shifts of the input pattern prior to being log-polar transformed. The situation is just reversed for the recognition module operating in the spatial domain. In this case the spatial shifts are embedded into the ensemble of classifiers whereas the input pattern is sequentially rotated before put into the classifiers. Such mixed operation was possible due to very fast response of the ensemble of classifiers [6].

The arbitration unit follows the already described winner-takes-all strategy (3) augmented by the mechanism promoting the most numerous group of unanimous experts; details given in [6].

5 Experimental Results

The system was implemented with the C++ and tested on the IBM PC with Pentium 4 3.4G and 2GB RAM. The system has two types of detectors for the circular signs: The first is based on the structural tensor [5] and is used during classification; The second is a manual detection where a user selects a rectangle containing a sign. It was used to create the data base of the "B" and "C" prototypes from their formal specification (Fig. 7a) [18]. From experiments we determined the optimal size of a single (binary) reference image to 64×64 pixels. This gives 2756 bits for each pattern in log-polar Fig. 7 b and spatial Fig. 7c representations. This is also a number of input neurons for each of the experts (Fig. 4). The smaller sizes gave much worse results because of lack of sufficient salient features. In this case classification was also more problematic since many different patterns had the same binary distance. On the other hand, bigger sizes gave little better results, however at increased computation costs.

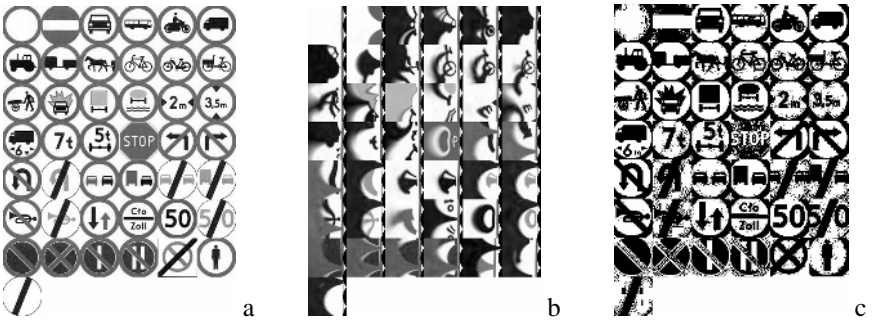


Fig. 7. The data base of the group "B" of RS: prototypes (a), log-polar (b), spatial features (c)

We tested the system for the groups "B" and "C" of road signs. Our maximal setup was as follows:

1. In the log-polar module for the ensemble of classifiers we had only vertical shifts (which correspond to the rotations in the spatial domain) from -8 to +8 (±45°) with a step of 2. The sequential change was limited to the ±4 pixels vertical and horizontal shifts with 2 pixels step of the input pattern before the log-polar. This gave a total of 9·5·5=225 tested deformations. The average exec. time was 0.9 s.
2. In the spatial domain the ensemble of classifiers supports ±8 pixels vertical and horizontal shifts whereas the input pattern was sequentially rotated from -30° to 30° with a step of 5°. This gave a total of 9·9·13=1053 possible deformations. The average execution time was also 0.9s due to lack of log-polar computations.

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

1. The first DB consist of 50% of deformable (-8 to 8 pixels shifted, -15-15° rotations with 5° step) road signs that were used during the system training and 50% of non-road-signs. Each image was added the Gaussian noise at certain level. Results are presented in Table 1 for different peek-signal-to-noise (PSNR).

Table 1. RS recognition accuracy from the DB with 50/50 of road-signs and non-road-signs images (4900 total). These road-signs were also used to train the system (LP = log-polar).

Parameters	"B" LP + spatial		"B" spatial		"C" LP + spatial		"C" spatial	
PSNR[dB]	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
100	1.0000	0.8922	0.5998	1.0000	1.0000	0.9005	0.6601	1.0000
80	1.0000	0.8788	0.6061	0.9900	1.0000	0.8788	0.6990	0.9833
70	1.0000	0.8149	0.6171	0.9890	1.0000	0.8147	0.6712	0.9901
60	0.9700	0.7700	0.5499	0.9747	1.0000	0.8454	0.6005	0.8999
50	1.0000	0.8113	0.5501	0.9600	0.9990	0.8740	0.5998	0.9213

2. The second DB is also 50/50, however the road-signs are taken from the real scenes. The Gaussian noise was also added. Results are contained in Table 2.

We found also that the best results are if the acceptance threshold is 0.001 (i.e. all lower scores are rejected) in the log-polar ensemble. For the ensemble operating with the spatial features this acceptance threshold is 0.05 - higher values of this parameter caused fall of recall value, since some "good" objects are classified as "don't know".

Table 2. Road signs recognition accuracy from the DB with 50/50 of road-signs and non-road-sign images (400 total). These road-signs in the DB where extracted from the real road scenes.

Parameters	"B" LP + spatial		"B" spatial		"C" LP + spatial		"C" spatial	
PSNR[dB]	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
100	0.8560	0.8239	0.4390	0.8990	0.7789	0.8911	0.6995	0.9211
80	0.8000	0.7996	0.4400	0.8171	0.7698	0.8910	0.6633	0.9049
70	0.7987	0.8013	0.4004	0.8200	0.7735	0.8880	0.6278	0.9002
60	0.7503	0.8200	0.3550	0.7998	0.6990	0.8100	0.5890	0.8756
50	0.7299	0.7098	0.3000	0.7099	0.6500	0.7927	0.5520	0.8234

Table 2 presents results of recognition accuracy for real images. The results are quite promising, although due to some geometric deformations and intrinsic noise they are a little bit worse than the presented in Table 1. For both tables it is visible that the system with the LP exhibits much better Precision at a little lower Recall compared to the version with the single classifier operating in the spatial domain only.

6 Conclusions

The paper presents a neural system for recognition of the circular road signs of the "B" and "C" groups. It is a part of the complex system for the advanced driving assistance. The recognition is performed by two committee machines operating in the spatial and log-polar input spaces, respectively. Both committee machines are built of the HNN operating as 1-nearest-neighbour classifiers. They operate on deformed versions of the input prototypes. In both cases the deformations are horizontal and vertical shifts – in the log-polar space they mean rotation and scale change in respect to the spatial domain. The crucial for the system is proper operation of the detector and feature extraction modules. In our system the input images were preprocessed by a simple binarization around the median (for the log-polar) and mean intensities. This

helped in data reduction however at a cost of the system performance (some signs can not be detected due to low feature discrimination). Thus, a future improvement can be achieved with most discriminative feature detectors.

Nevertheless, the experiments showed low computational demands, fast execution and high robustness of the system which allows reliable classification of the circular RS even from the highly deformed or partially occluded (<15% of occlusions) inputs.

Acknowledgement

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

References

1. Abu-Mostafa, Y.S.: Hints. Neural Computations, No. 7 (1995) 639-671
2. Amit, Y.: 2D Object Detection and Recognition, MIT Press (2002)
3. Aoyagi, Y., Asakura, T.: A study on traffic sign recognition in scene image using genetic algorithms and neural networks in IEEE Conf. Electronics, Control, (1996) 1838–1843
4. Chen, X., Yang, J., Zhang, J., Waibel, A.: Automatic Detection and Recognition of Signs From Natural Scenes. IEEE Trans. on Image Proc. v.13, no 1 (2004) 87-99
5. Cyganek, B.: Object Detection in Multi-Channel and Multi-Scale Images Based on the Structural Tensor Springer LNCS 3691 (2005) 570-578
6. Cyganek, B.: Recognition of Road Signs with Mixture of Neural Networks and Arbitration Modules. Proc. of ISNN, China, LNCS 3973, Springer (2006) 52 – 57
7. DaimlerChrysler, The Thinking Vehicle, http://www.daimlerchrysler.com (2002)
8. Duch, W., Grudziński, K.: A framework for similarity-based methods. Second Polish Conference on Theory and Applications of Artificial Intelligence (1998) 33-60
9. Escalera, A., Moreno, L., Salichs, M. A., Armingol, J. M.: Road traffic sign detection and classification, IEEE Trans. Ind. Electron., v. 44, (1997) 848–859
10. Escalera, A., Armingol, J.A.: Visual Sign Information Extraction and Identification by Deformable Models. IEEE Tr. On Int. Transportation Systems, v. 5, no 2, (2004) 57-68
11. Floréen, P.: Computational Complexity Problems in Neural Associative Memories. PhD Thesis, University of Helsinki, Department of Computer Science, Finland (1992)
12. Hastie, T., Tibshirani, R., Friedman: The Elements of Statistical Learning. Springer (2001)
13. Kara, L.,B., and Stahovich, T.,F.: An image-based, trainable symbol recognizer for hand-drawn sketches. Computers & Graphics, Volume 29, Issue 4, August (2005) 501-517
14. Lippman, R.: An introduction to computing with neural nets. IEEE Transactions on Acoustic, Speech, and Signal Processing, v.ASSP-4 (1987) 4-22
15. Luo, R. C., Potlapalli, H.: Landmark recognition using projection learning for mobile robot navigation, in Proc. IEEE Int. Conf. Neural Networks, v.4, (1994) 2703–2708
16. Klette, R., Rosenfeld, A.: Digital Geometry. Morgan Kaufman (2004)
17. Piccioli, G., Micheli, E.D., Parodi, P., Campani, M.: Robust method for road sign detection and recognition. Image and Vision Computing, v.14 (1996) 209-223
18. Road Signs and Signalization. Directive of the Polish Ministry of Infrastructure, Internal Affairs and Administration (Dz. U. Nr 170, poz. 1393) (2002)
19. Zheng, Y. J., Ritter, W., Janssen, R.: An adaptive system for traffic sign recognition, in Proc. IEEE Intelligent Vehicles Symp. (1994) 165–170
20. Zokai, S., Wolberg, G.: Image Registration Using Log-Polar Mappings for Recovery of Large-Scale Similarity. IEEE Transactions on Image Processing, 14(10) (2005) 1422-1433

Computer Aided Classification of Mammographic Tissue Using Independent Component Analysis and Support Vector Machines

Athanasios Koutras[1], Ioanna Christoyianni[1], George Georgoulas[2], and Evangelos Dermatas[1]

[1] WCL, Electrical & Computer Engineering Dept., University of Patras
26100 Patras, Hellas
[2] LAR, Electrical & Computer Engineering Dept., University of Patras
26100 Patras, Hellas
koutras@giapi.wcl2.ee.upatras.gr

Abstract. In this paper a robust regions-of-suspicion (ROS) diagnosis system on mammograms, recognizing all types of abnormalities is presented and evaluated. A new type of statistical descriptors, based on Independent Component Analysis (ICA), derive the source regions that generate the observed ROS in mammograms. The reduced set of linear transformation coefficients, estimated from ICA after principal component analysis (PCA), compose the features vector that describes the observed regions in an effective way. The ROS are diagnosed using support-vector-machines (SVMs) with polynomial and radial basis function kernels. Taking into account the small number of training data, the PCA preprocessing step reduces the dimensionality of the features vector and consequently improves the classification accuracy of the SVM classifier. Extensive experiments using the Mammographic Image Analysis Society (MIAS) database have given high recognition accuracy above 87%.

1 Introduction

Breast cancer has been a leading cause of fatality among all cancers for women. X-ray mammography is the most effective, low-cost, and highly sensitive technique for detecting small lesions [1]. The radiographs are searched for signs of abnormality by expert radiologists but complex structures in appearance and signs of early disease are often small or subtle. That's the main cause of many missed diagnoses that can be mainly attributed to human factors [1,2]. However, the consequences of errors in detection or classification are costly, so there has been a considerable interest in developing methods for automatically classifying suspicious areas of mammography tissue, as a means of aiding radiologists by improving the efficacy of screening programs and avoiding unnecessary biopsies.

Among the various types of breast abnormalities clustered microcalcifications and mass lesions are the most important ones. Masses and clustered microcalcifications often characterize early breast cancer [3] that can be detectable before a woman or the physician can palp them. Masses appear as dense regions of varying sizes and properties and can be characterized as circumscribed (Fig 1a), spiculated (Fig 1b), or

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 568 – 577, 2006.

ill defined (Fig 1c). On the other hand, microcalcifications (Fig 1f), appear as small bright arbitrarily shaped regions on the large variety of breast texture background. Finally, asymmetry, and architectural distortion (Fig 1e-d), are also very important and difficult abnormalities to detect.

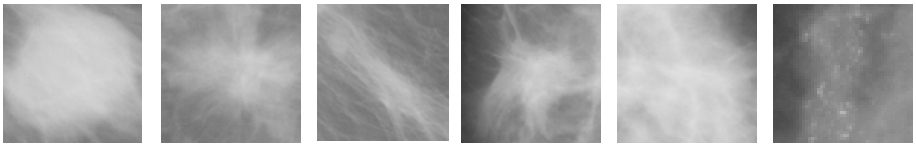


Fig. 1. Types of breast cancer on mammograms

Computer-aided methods in the field of digital mammography are divided into two main categories; computer aided detection methods [4-6] that are capable of pinpointing regions of suspicion (ROS) in mammograms for further analysis from an expert radiologist and computer aided diagnosis methods [5-7] which are capable of making a decision whether the examined ROS consist of abnormal or healthy tissue. However, the development of methods for recognizing the identity of a ROS and the exploration of other types of classifiers [8,9], especially in the case of all kinds of abnormalities has been very limited [7,10]. In this study, a method to classify regions of suspicion (ROS) that contain abnormal or healthy tissue using Support Vector Machines (SVM) is proposed.

SVMs is a new technique for data classification and regression, which has gained great interest during the last decade [11,12]. The main idea behind SVMs for classification is to map the input space into a feature space of much greater dimension and then construct a hyperplane in such a way that the margin of separation is maximized. However this is the ideal scenario and it has to be slightly modified to encompass the non-separable case. In that case the SVM can provide a good generalization performance and seems to be quite insensitive to overfitting.

An important role in the performance of the SVM plays the selection of the so called "inner-product kernel". Depending on how this kernel is generated, different machines with different nonlinear decision surfaces in the input space can be constructed (in the feature space the decision surface is always a hyperplane). The most common machines are the polynomial learning machines, the radial basis function networks and two layer perceptrons [12]. In this study, the polynomial learning machines and the radial basis function implementation is evaluated for the binary classification problem of discrimination between abnormal and healthy tissue in digital mammograms.

The descriptors of normal and abnormal tissue were estimated directly from the image data using ICA, a signal processing technique employed in various signal processing applications [13]. The purpose of ICA is to estimate a linear non-orthogonal coordinate system in multivariate data. The directions of the axes are determined not only by the data's first and second order statistics, but also by higher order statistics. In our approach we consider the normal and the abnormal regions of mammograms to be generated by a set of independent images, namely the source regions that are

estimated using standard ICA techniques. The coefficients of the linear combination of the independent source regions are the features that are fed into the SVM classifier. Additionally, a preprocessing step implementing Principal Component Analysis (PCA) is presented for reducing the dimensionality of these features without affecting the classification accuracy. Extensive experiments have shown great accuracy of 87.39% in recognizing normal and abnormal breast tissue in mammograms which is similar to the performance of neural network classifiers [7].

The structure of this paper is as follows: In the next section, a detailed description of the features extracted from mammograms using ICA techniques is given. Additionally, the implemented technique for reducing the dimensionality of the extracted features is presented. In section 3, a brief description of the SVM classifier is given. In section 4 the data set and the experimental results are presented and finally, in section 5 some conclusions are provided.

2 Feature Extraction from Mammograms

2.1 ICA Based Feature Extraction

The aim of the proposed feature extraction technique is to estimate a set of descriptors that can be used to describe the healthy and tumorous regions of mammograms in an effective way. To this direction, we assume that the observed regions of mammograms are generated by a linear combination of an unknown set of statistically independent source regions according to the equation

$$X = A \cdot S \,, \tag{1}$$

where A is the mixing matrix that generates the X observed regions from the S independent source regions, with the coefficients used in the linear combination being in the rows of A. These coefficients are employed as feature descriptors that describe uniquely the abnormal and the normal regions in X, in a most fitting way. The source regions are estimated using standard ICA techniques as follows:

Let us consider a set of N regions of mammograms used for the training procedure, containing normal and abnormal tissue with dimensions KxL pixels. The regions are first converted to one-dimensional vectors with length $D=\{KxL\}$. These vectors form the rows of the observation matrix X_{train} with dimensions NxD and are fed into the ICA neural network seen in Figure 2.

The source regions are estimated by: $S = W \cdot X_{train}$, where W is the NxN ICA separating matrix. To estimate this matrix in an unsupervised manner, we have applied the Maximum Likelihood Estimation criterion (MLE). The log-likelihood of the observed regions is given by:

$$L = \log(p_X(X_{train}; W)) = \log(|W|) + \log(p_S(S)) \tag{2}$$

The weights of the ICA network are estimated using the stochastic gradient of L with respect to the matrix W:

$$\frac{\partial L}{\partial W} = \left[W^{-1}\right]^T - \Phi(S) \cdot X_{train}^T \,, \tag{3}$$

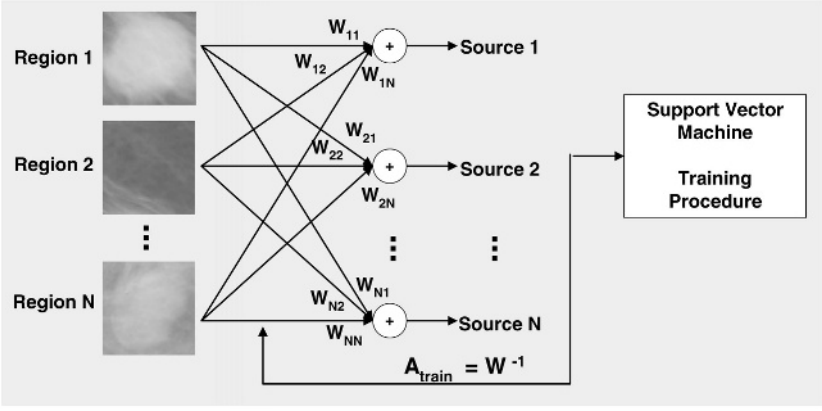


Fig. 2. The ICA based feature extraction scheme for the training procedure

where $\Phi(S) = -\left[\frac{p_1'(s_1;W)}{p_1(s_1;W)}, \cdots, \frac{p_N'(s_N;W)}{p_N(s_N;W)}\right]^T$ and $p_i(s_i;W)$ is the probability density function of the i^{th} source region. The marginal pdf of the source regions was chosen experimentally to follow the hyperbolic cosine distribution with $p_i(s_i;W) \propto 1/\cosh(s_i)$, so $\Phi(s_i) = \tanh(s_i)$. From equation (3) and using the natural gradient approach, we conclude to the following weight adaptation rule for W:

$$\Delta W = -n\frac{\partial L}{\partial W}W^T W = n\left[I - \Phi(S)S^T\right]\cdot W \tag{4}$$

By applying the above learning rule, we can find the separating matrix W with dimensions NxN and the N independent source regions in the matrix S with dimensions $\{KxL\}$. The features that are used to describe the observed regions of mammograms are contained in the rows of the inverse of the separating matrix $A_{train} = W^{-1}$.

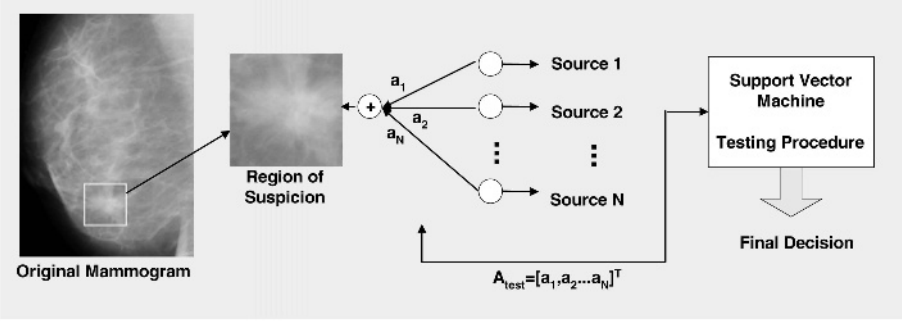


Fig. 3. The ICA based feature extraction scheme for the testing procedure

For the testing procedure presented in Figure 3, each observed ROS is generated by a linear combination of the learned source regions S using the coefficients $A_{test} = X_{test} \cdot S^{\#}$, where the operator # denotes the pseudoinverse of a matrix. The feature vectors in A_{test} are then fed into the SVM classifier and the final decision is made whether the tested ROS is normal or abnormal.

2.2 Dimensionality Reduction

The dimensionality of the extracted features depends strongly on the number of the regions used in the training procedure. In case of a large training set, the dimensionality of the extracted features vector increases enormously, which complicates the task of the implemented classifier. On the other hand, when using a small training set, the performance of the ICA network deteriorates and the independent source regions cannot be estimated correctly. In order to face this problem a PCA preprocessing step is added. In this case, instead of performing ICA on the N observed regions, ICA is applied on a subset of K linear combinations where $K<N$. This technique does not significantly affect the ICA network's performance, as the initial regions have been replaced with another linear combination. The use of the PCA preprocessing step does not destroy the higher order relationship between the initial regions, but eliminates only the second order dependences (same as the whitening technique). The higher order relations still exist in the data and are not separated.

PCA can be implemented using eigenvalue decomposition on the covariance matrix of the observed regions in X_{train}. Let P be the matrix DxN with the N principal components in its columns, sorted by descending order with respect to their variances. By taking the first K more significant principal components and performing ICA on the data in P^T coefficients, the K independent source images in the rows of S are estimated. The new feature vectors of the observed regions in X_{train} are determined as follows:

The representation R_m of X_{train} based on the principal components in P is defined as $R_m = X_{train} \cdot P$. The regions in X_{train} can be approximated using the minimum squared error [14] as $X_{rec} = R_m \cdot P^T$. By applying the rule in equation (4) on the first K principal components, the matrix W is estimated such that $S = W \cdot P^T$, therefore $P^T = W^{-1} \cdot S$. Using the above equation we find that $X_{rec} = R_m \cdot P^T = R_m \cdot W^{-1} \cdot S$. The rows of the transformation matrix

$$B_{train} = R_m \cdot W^{-1} \tag{5}$$

contain the coefficients of the linear combination of the statistically independent regions in S that generate the observations in X_{rec}, therefore they can be used as feature vectors with reduced dimensionality K to describe the observed regions in a compact and more efficient way.

In the testing procedure, each region in X_{test} is processed with the principal components P_m estimated from the training procedure $R_{test} = X_{test} \cdot P_m$. The feature vector is calculated by

$$B_{test} = R_{test} \cdot W^{-1}. \tag{6}$$

3 Support Vector Machines

SVMs are learning systems that are trained using an algorithm from optimization theory [11]. The main idea behind SVMs, when dealing with a pattern classification problem, is to find an "optimal" hyperplane as the solution to the learning problem. By the term "optimal", it is suggested that for a separable classification task, the hyperplane ($\mathbf{w}$, b) with the maximum margin from the closest data points belonging to the different classes is selected.

Consider a data set $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ of labeled examples $y_i \in \{-1,1\}$. In the simple case of linearly separable patterns, a hyperplane $\mathbf{w}_o \cdot \mathbf{x} + b_o = 0$ can be constructed where the vector $\mathbf{W}_o$ minimizes $\frac{1}{2}\|\mathbf{w}\|^2$ subject to the constraints

$$y_i((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq 1 \quad i = 1,...,n \tag{7}$$

As a linear function is often not adequate in real problems to perform this separation, a mapping of the input space into a high dimensional feature space is involved via a non linear mapping $\boldsymbol{\varphi}(\cdot)$. Therefore for each training example $\mathbf{x}_i$, a non-linear mapping $\boldsymbol{\varphi}(\mathbf{x}_i)$ is considered, and in order to achieve perfect classification the following condition should be satisfied

$$y_i((\mathbf{w} \cdot \phi(\mathbf{x}_i)) + b) \geq 1 \quad i = 1,...,n \tag{8}$$

Still the quantity to be minimized is $\frac{1}{2}\|\mathbf{w}\|^2$. However, only the mapping into a higher feature space through a nonlinear function does not guaranty perfect separation of the classes, therefore we can introduce slack-variables ξ_i that measure the deviation of a data point from the ideal condition of pattern separability and relax the hard margin constraints as follows:

$$y_i((\mathbf{w} \cdot \phi(\mathbf{x}_i)) + b) \geq 1 - \xi_i, \ \xi_i \geq 0, \ i = 1,...,n \tag{9}$$

allowing some misclassifications. Now the new quantity that has to be minimized is

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i, \tag{10}$$

where C is a positive user specified parameter that penalizes margin errors, i.e. patterns that lie within the margin as well as those that are on the wrong side of the decision surface. The solution to this optimization problem subject to the constraints is given by the saddle point of the primal Lagrangian equation:

$$L_p(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left(y_i\left(\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}_i) + b\right) - 1 + \xi_i\right) - \sum_{i=1}^{n}\beta\xi_i \tag{11}$$

This leads to the dual maximization problem of the dual Langrangian equation:

$$L_d(\boldsymbol{\alpha}) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j (\boldsymbol{\varphi}(\mathbf{x}_i) \cdot \boldsymbol{\varphi}(\mathbf{x}_j)) \tag{12}$$

subject to the constraints

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \text{ and } C \geq \alpha_i \geq 0 \quad , \; i = 1,...,n \tag{13}$$

The solution of the above optimization problem leads to the "optimal" discriminating function:

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{n} y_i a_i \left(\boldsymbol{\varphi}^T(\mathbf{x}_i) \boldsymbol{\varphi}(\mathbf{x}) \right) + b \right) \tag{14}$$

The points for which $a_i > 0$ are called Support Vectors and they are usually a small portion of the original data set. The inner product in the feature space can be written in the form of:

$$\boldsymbol{\varphi}^T(\mathbf{x}_i) \boldsymbol{\varphi}(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j) \tag{15}$$

where K is called the inner-product kernel. Instead of calculating the inner product in the feature space $\boldsymbol{\varphi}^{\mathrm{T}}(\mathbf{x}_i)\boldsymbol{\varphi}(\mathbf{x}_j)$, one can indirectly calculate it using the kernel function $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)$. Therefore, by selecting an appropriate symmetric positive semi-definite kernel function, it is not necessary to know the actual mapping [12].

Depending on the choice of the kernel function, different learning machines with different nonlinear decision surfaces can be constructed. Among others the most popular are the polynomial learning machines, the radial basis function networks and the two-layer perceptrons. In our experimental procedure we have employed polynomial learning machines (the power p is specified a priori by the user [15,11])

$$K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}^T \mathbf{x}_i + \mathbf{1})^p, \; i = 1,...n \tag{16}$$

and radial basis function machines (the width σ^2, which is common to all kernels is specified also a priori by the user [15])

$$K(\mathbf{x}, \mathbf{x}_i) = \exp(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{x}_i\|^2), \; i = 1,...n \tag{17}$$

4 Experimental Results

4.1 The MIAS Data Set

In our experiments the MIAS MiniMammographic Database [16], provided by the Mammographic Image Analysis Society (MIAS), was used. The mammograms are digitized at 200- micron pixel edge, resulting to a 1024x1024-pixel resolution.

In the MIAS Database there is a total of 119 ROS containing all kinds of existing abnormal tissue from masses to clustered microcalcifications. The smallest abnormality extends to 3 pixels in radius, while the largest one to 197 pixels. These 119 ROS along with another 119 randomly selected sub-images from entirely normal mammograms were used throughout our experiments.

4.2 Classification Results

From a total number of 238 ROS included in the MIAS database, 119 regions are used for the training procedure (*jackknife* method): 60 groundtruthed abnormal regions along with 59 randomly selected normal ones. In the evaluation procedure, the remaining 119 regions are used that contain 59 groundtruthed abnormal regions together with 60 entirely normal regions. Therefore, no ROS was used both in the training and testing procedure.

In order to reduce the dimensionality of the extracted feature vectors (119 dimensional), we implemented the PCA preprocessing step as described earlier. The principal components were estimated by calculating the eigenvectors of the covariance matrix of the training set. ICA was then performed successively on the first 5 to 15 most significant of these eigenvectors, which resulted in 5 to 15 dimensional feature vectors and 119 independent source regions with 1125 pixels length. These features were used to train the implemented SVM classifier.

For the testing procedure, the remaining 119 ROS were first preprocessed with the eigenvectors estimated from the previous step and then their features were extracted using equation (6).

The extracted features were used in the SVM classifier, the percent recognition rate of which is shown in Table 1. Specifically, the implemented feature extraction technique, resulted in features that can describe effectively both healthy and tumorous regions achieving high recognition accuracy above 80% in all of the cases. The best results were obtained when using only 5 principal components, derived in the PCA preprocessing step, which resulted in an extremely low-dimensionality feature vector and a low complexity neural and SVM classifier, achieving a total recognition accuracy of 87.39% in the SVM-RBF kernel configuration.

On the other hand, in order to find the best configuration for our SVM classifier (the values for (p,C) for the case of polynomial machine and (σ^2,C) for the RBF machine where the recognition accuracy is maximized), a "grid-search" approach was used in a systematic manner with different values for the parameters followed by a cross validation to pick the best combination.

In detail, the SVM based classifier using the RBF kernel recognized correctly 53 out of 59 abnormal and 51 out of 60 normal ROS, giving a total number of 104 out of 119 correct classifications for the case we retain the five most significant principal components. The SVM based classifier using the Polynomial Learning Machines recognized correctly 50 out of 59 abnormal and 49 out of 60 normal ROS, giving a total number of 99 out of 119 correct classifications. These were obtained using a polynomial kernel of degree 2.

Further experiments were performed using more than 15 components in the PCA preprocessing step, but the results showed no further improvement of the recognition accuracy. The experimental results using the complete features set (the 119 dimensional features vector) without the PCA preprocessing step showed a recognition accuracy far below that achieved when only the first 5 principal components were used.

The SVM based classifier's accuracy can be compared to that of the Radial Basis Function neural networks [7] (total of 88.23% recognition rate; 88.13% for normal tissue and 88.33% for the abnormal one) when used on the exact same descriptors.

Table 1. Recognition accuracy for the ICA features and both types of SVM kernels

		15 components	10 components	5 components
Polynomial Learning Machines	Normal	81.66%	81.66%	81.66%
	Abnormal	83.05%	84.74%	84.74%
	Total	82.35%	83.19%	83.19%
RBF kernel	Normal	81.66%	81.66%	**85%**
	Abnormal	88.13%	88.13%	**89.83 %**
	Total	84.87%	84.87%	**87.39 %**

5 Conclusion

In this paper we investigated the performance of a classifier based on Support Vector Machines and a new set of statistical features based on ICA, in the problem of recognizing breast cancer in ROS of digital mammograms. It is well known that the disease diagnosis on mammograms is a very difficult task even for experienced radiologists due to the great variability of the mass appearance. The experimental results showed quite similar accuracy for both implemented SVM kernels (RBF and polynomial kernels) when used to recognize all different types of breast abnormalities. Nevertheless, the achieved recognition accuracy is promising and needs to be improved in order to become a great assistance for radiologists in their diagnosis.

References

1. Martin, J., Moskowitz, M. and Milbrath, J.: Breast cancer missed by mammography. AJR, Vol. 132. (1979) 737
2. Kalisher, L.: Factors influencing false negative rates in xero-mammography. Radiology, Vol.133. (1979) 297
3. Tabar, L. and Dean, B.P.: Teaching Atlas of Mammography.2nd edition, Thieme, NY (1985)
4. Christoyianni, I., Dermatas, E., and Kokkinakis, G.: Fast Detection of Masses in Computer-Aided Mammography. IEEE Signal Processing Magazine, vol. 17, no 1. (2000) 54-64
5. Sonka, M., Fitzpatrick, J. : Handbook of Medical Imaging. SPIE Press (2000).
6. Doi, K., Giger, M., Nishikawa, R., and Schmidt, R. (eds.): Digital Mammography 96. Elsevier Amsterdam (1996)
7. Christoyianni, I., Koutras, A., Dermatas, E., and Kokkinakis, G.: Computer Aided Diagnosis of Breast Cancer in Digitized Mammograms", Computerized Medical Imaging and Graphics. Elsevier, vol. 26, no 5. (2002) 309-119
8. Bazzani, A., Bevilacqua, A., Bollini, D., Brancaccio, R., Campanini, R., Lanconelli, N., Riccardi, A., Romani, D., Zamboni, G. : Automatic detection of clustered microcalcifications in digital mammograms using a SVM classifier. European Symposium on Artificial Neural Networks, Bruges, Belgium, (2000) 195-200

9. Wei L., Yang Y., Nishikawa R. M., Jiang Y., "A Study on Several Machine-Learning Methods for Classification of Malignant and Benign Clustered Microcalcifications", IEEE Transactions on Medical Imaging, pp. 1-10, Jan 2005.
10. Christoyianni, I., Dermatas, E. and Kokkinakis, G.: Neural Classification of Abnormal Tissue in Digital Mammography Using Statistical Features of the Texture. IEEE Int. Conference on Electronics, Circuits and Systems. vol 1, (1999) 117-120
11. Schφlkopf, B., Burges, C.J.C. and Smola, A.J.: Advances in Kernel Methods. Support Vector Learning. London The MIT Press (1999).
12. Burges, C.J.C : A Tutorial on Support Vector Machines for Pattern Recognition. Data mining and Knowledge Dicovery. vol. 2. (1998) 121-167
13. Lee, Te-Won: Independent Component Analysis: Theory and Applications. Kluwer Academic Publishers (1998)
14. Bartlett, M., Lades, M., Sejnowski, T.: Independent component representation for face recognition. Proc. SPIE Symposium on Electronic Imaging: Science and Technology (1998)
15. Haykin, S.: Neural Networks: A Comprehensive Foundation. Englewoo Cliffs, NJ: Prentice Hall, (1999)
16. http://peipa.essex.ac.uk/info/mias.html

Growing Neural Gas for Vision Tasks with Time Restrictions

José García, Francisco Flórez-Revuelta, and Juan Manuel García

Department of Computer Tecnology and Computation. University of Alicante.
Apdo. 99. 03080 Alicante, Spain
{jgarcia, florez, juanma}@dtic.ua.es

Abstract. Self-organizing neural networks try to preserve the topology of an input space by means of their competitive learning. This capacity is being used for the representation of objects and their motion. In addition, these applications usually have real-time constraints. In this work, diverse variants of a self-organizing network, the Growing Neural Gas, that allow an acceleration of the learning process are considered. However, this increase of speed causes that, in some cases, topology preservation is lost and, therefore, the quality of the representation. So, we have made a study to quantify topology preservation using different measures to establish the most suitable learning parameters, depending on the size of the network and on the available time for its adaptation.

1 Introduction

Self-organizing neural networks, by means of a competitive learning, make an adaptation of the reference vectors of the neurons, as well as, of the interconnection network among them; obtaining a mapping that tries to preserve the topology of an input space. Besides, they are able of a continuous re-adaptation process even if new patterns are entered, with no need to reset the learning.

These capacities have been used for the representation of objects [1] (figure 1) and their motion [2] by means of the Growing Neural Gas (GNG) [3], that has a learning process more flexible than other self-organizing models, like Kohonen maps [4].

These two applications, representation of objects and their motion, have in many cases high temporal constraints, reason why the adaptation of the network within the available time cannot be assured. This is feasible by modifying the learning parameters of the GNG to conclude into the deadline. Nevertheless, this can affect the quality of the adaptation, measured as the topology preservation of the input space [5].

In other applications, no deadline has to be accomplished, but it is allowed the interruption of the adaptation. This means that to give a correct representation of the input space, a good preservation of the topology should be maintained throughout the learning process.

In this work first we present in section 2 the original algorithm of GNG and the different learning parameters and conditions of finalization tested for different input spaces, in section 3 we present different topology preservation measures for self-organizing maps, next in section 4 the results of the experiments are showed and finally we extract some conclusions from the experiments and further work.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 578 – 586, 2006.

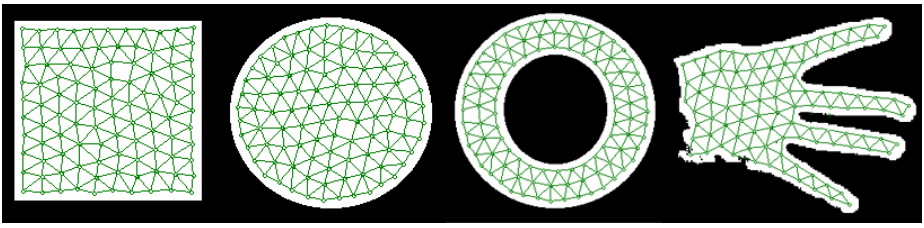


Fig. 1. Representation of two-dimensional objects with a self-organizing network

2 Growing Neural Gas

The Growing Neural Gas is an incremental neural model that avoids the necessity to previously specify the network size, as other methods require. On the contrary, from a minimal network, a growth process takes place that is continued until an ending condition is fulfilled. Also, learning parameters are constant in time, in contrast to other methods whose learning falls basically in decaying parameters.

2.1 GNG Algorithm

The GNG learning algorithm to approach the network to the input manifold is as follows:

1. Start with two neurons a and b at random positions w_a and w_b in $\mathcal{R}^d$.
2. Generate an input signal ξ according to a density function $P(\xi)$.
3. Find the nearest neuron (winner neuron) s_1 and the second nearest s_2.
4. Increase the age of all the edges emanating from s_1.
5. Add the squared distance between the input signal and the winner neuron to a counter error of s_1:

$$\Delta error(s_1) = \left\| w_{s_1} - \xi \right\|^2 \tag{1}$$

6. Move the winner neuron s_1 and its topological neighbours (neurons connected to s_1) towards ξ by a learning step ε_w and ε_n, respectively, of the total distance:

$$\Delta w_{s_1} = \varepsilon_w (\xi - w_{s_1}) \tag{2}$$

$$\Delta w_{s_n} = \varepsilon_n (\xi - w_{s_n}) \tag{3}$$

7. If s_1 and s_2 are connected by an edge, set the age of this edge to 0. If it does not exist, create it.
8. Remove the edges larger than a_{max}. If this results in isolated neurons (without emanating edges), remove them as well.
9. Every certain number λ of input signals generated, insert a new neuron as follows:

- Determine the neuron q with the maximum accumulated error.
- Insert a new neuron r between q and its further neighbour f:

$$w_r = 0.5(w_q + w_f) \tag{4}$$

- Insert new edges connecting the neuron r with neurons q and f, removing the old edge between q and f.
- Decrease the error variables of neurons q and f multiplying them with a constant α. Initialize the error variable of r with the new value of the error variable of q and f.

10. Decrease all error variables by multiplying then with a constant β.
11. If the stopping criterion is not yet achieved, go to step 2. (In our case the insertion of 100 neurons or 1 second of available time)

2.2 Modification of the Parameters of the Growing Neural Gas to Accelerate the Learning Process

The conclusion of the competitive learning of the GNG usually comes determined by the insertion of all the neurons until obtaining a predetermined size. Nevertheless, if a temporal factor is included as condition of conclusion, it will not be possible, in some cases, to complete the adaptive process with the consequent loss of topology preservation; creating connections between neurons that would not have to be joined (figure 2) or removing other that should be created. So, there will be differences between the final configuration of the network and the Delaunay triangulation that must have been established.

If a complete network, with all its neurons, wants to be obtained in a predetermined time, its learning algorithm has to be modified to accelerate its conclusion. The main factor in the learning time is the number of input signals generated by iteration, since new neurons are inserted (step 9 of the learning process) at smaller intervals, taking less time in completing the network.

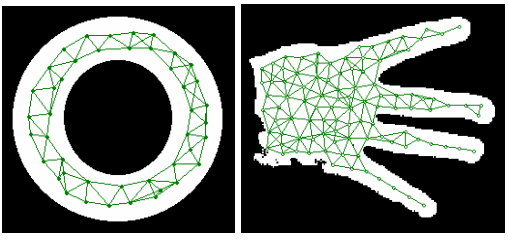


Fig. 2. Incomplete (incorrect) adaptations due to an early conclusion of the learning process

Another alternative is to insert more than a neuron by iteration, repeating the step 9 of the learning algorithm in several occasions. There is a work in this line [6] in where two neurons are inserted by iteration, according to diverse circumstances. In our case, step 9 is repeated several times, inserting neurons in those zones where greater accumulated error exists, creating the corresponding connections.

Nevertheless, these alternatives cause that the topology preservation of the input space, that is to say, the quality of the representation is affected (figure 3). For that reason, different measures of topology preservation are used to evaluate the correction of different adaptations throughout the time.

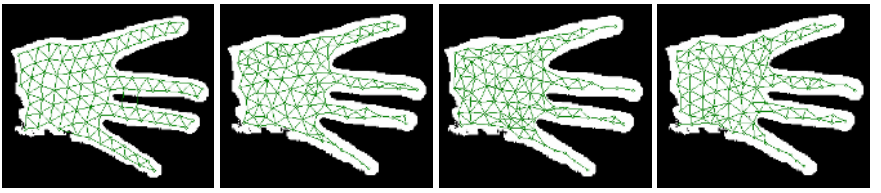


Fig. 3. Final adaptations depending on the number of neurons inserted by iteration: 1 (a), 2 (b), 5 (c) y 9 (d)

3 Measures of Topology Preservation

The adaptation of a self-organizing neural network is made mainly from two points of view: its resolution and its topology preservation of an input space.

The measure of resolution usually employed is the quantization error [4], expressed like:

$$\mathcal{E} = \sum_{\forall \xi \in \mathbb{R}^d} \left\| w_{s_\xi} - \xi \right\| \bullet p(\xi) \tag{5}$$

where s_ξ is the nearest neuron to the input pattern ξ.

One of the first developed measure to evaluate topology preservation was the topographic product [7] that compares the neighbourhood relationship among all pair of neurons of the network with concerning, on one hand to their position inside the map, and on the other hand, according to their reference vectors:

$$\mathcal{P} = \frac{1}{\mathcal{N}(\mathcal{N}-1)} \sum_{j=1}^{\mathcal{N}} \sum_{k=1}^{\mathcal{N}-1} log \left(\left(\prod_{l=1}^{k} \frac{d^{\mathcal{V}}\left(w_j, w_{n_l^{\mathcal{A}}(j)}\right)}{d^{\mathcal{V}}\left(w_j, w_{n_l^{\mathcal{V}}(j)}\right)} \cdot \frac{d^{\mathcal{A}}\left(j, n_l^{\mathcal{A}}(j)\right)}{d^{\mathcal{A}}\left(j, n_l^{\mathcal{V}}(j)\right)} \right)^{1/2k} \right) \tag{6}$$

where j is a neuron, w_j is its reference vector, n_l^V is the l-th closest neighbour to j in the input manifold $\mathcal{V}$ according to a distance $d^{\mathcal{V}}$ and n_l^A is the l-th nearest neuron to j in the network $\mathcal{A}$ according to a distance $d^{\mathcal{A}}$. In order to use this measure to non-linear input spaces the geodesic distance [8] is employed as $d^{\mathcal{V}}$.

On the other hand, the topographic function [9] compares the resulting neural network with the Delaunay triangulation induced by the input space, measuring the number of neurons that have adjacent receptive fields but are not connected and vice versa.

It would be desirable that all the measures considered both aspects: resolution and preservation of the topology. This is not true for the measures above: the topographic product and the topographic function. However, their resolution aspect is implicit in the competitive learning of self-organizing models.

Kaski and Lagus [10] proposed a goodness measure C that combines both aspects, obtaining the closest reference vector to every input pattern, and thereafter to the second-closest reference vector along the map. The result is the sum of these distances.

Deviations of these three measures from the zero value indicate a loss of topology preservation, indicating their sign, in the case of the product and the topographic function, if the dimensionality of the network is greater or smaller than the one of the input space represented.

4 Experiments and Results

In this section we are going to compare the quality of the representation of different networks, where their learning parameters have been modified to accelerate their adaptation. So, some of the learning parameters have been fixed ($\varepsilon_1 = 0.1$, $\varepsilon_2 = 0.01$, $\alpha = 0.5$, $\beta = 0.0005$, $a_{max} = 250$), modifying for each alternative the number of input signals and the neurons inserted by iteration. The different alternatives will be denoted like GNG_{χ}^{λ} where λ indicates the number of input signals and χ represent the amount of neurons inserted by iteration.

Different networks have been adapted to the input spaces displayed in figure 1. Since the results are very similar in all the cases, next we only present the results obtained for one of the objects, the ring.

Figure 4 shows the legend used for all the graphs where the first number represents the random input signals (step 2) and the second number the number of neurons inserted by iteration (step 9). For example 1000p1npi means 1000 input signals and 1 neuron inserted by iteration that in the notation used is GNG_1^{1000}.

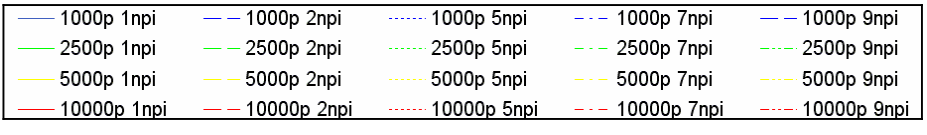


Fig. 4. Legend used for the graphs that present the results of the experiments

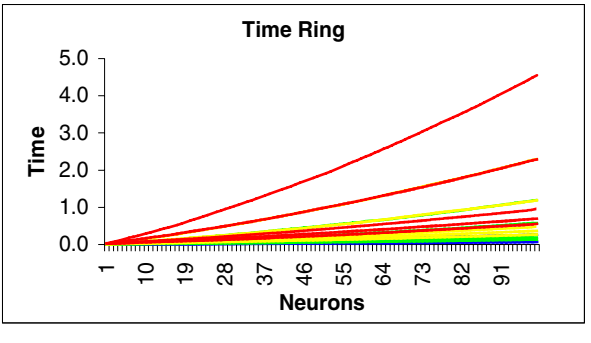


Fig. 5. Time used in the insertion of 100 neurons

4.1 Topology Preservation Depending on the Number of Neurons

In figure 5 learning time for different options is indicated. In this case the condition of conclusion is a pre-established size of the network (100 neurons).

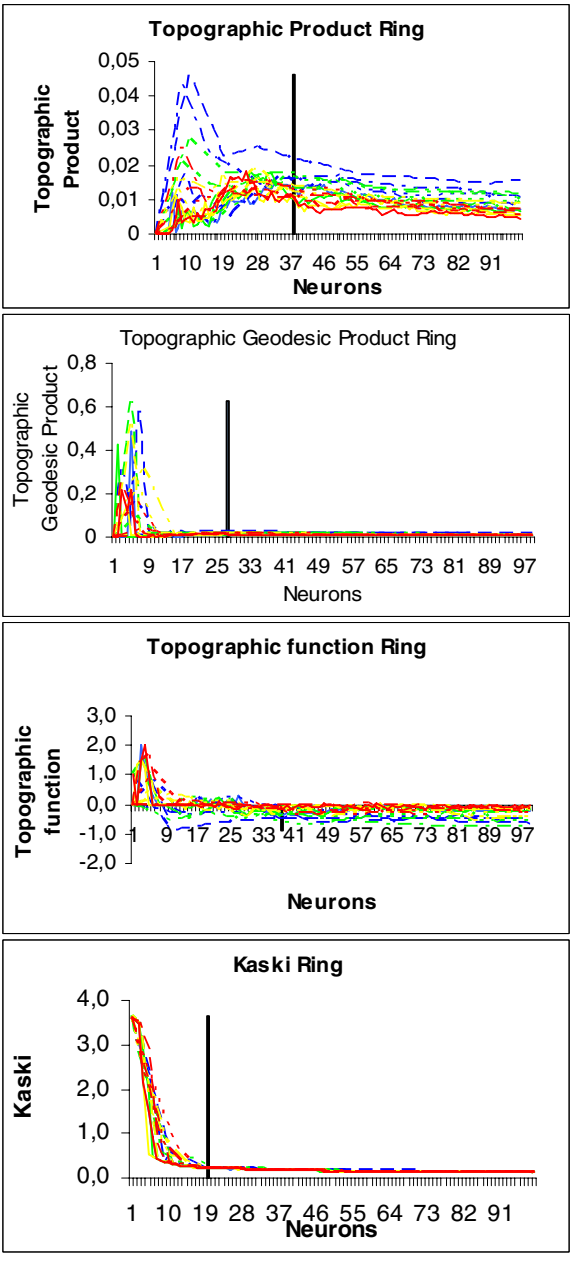


Fig. 6. Topology preservation depending on the number of neurons

Figure 6 shows topology preservation of the diverse variants depending on the number of neurons that the network has, throughout the learning process. This study is of interest in the case that temporal constraints that limit the adaptive process do not exist but a possible maximum size of the network is established, e.g. 100 neurons.

In initial stages of the adaptation the networks try to represent broadly the input space, this is the reason why topology preservation fluctuates considerably. When a small number of neurons are inserted, it is stabilized. Nevertheless, if the fastest options are employed, their topology preservation stays worse throughout the adaptive process, since there are edges between neurons that should not be connected and vice versa.

We represent with a vertical line the number of neurons that we consider necessary to achieve a correct topology preservation with any of the different measures.

4.2 Topology Preservation Depending on the Available Time

In figure 7 is showed the number of neurons that each network is able to insert in 1 second time. As it is obvious, inserting several neurons or reducing the signals by iteration allow to obtain networks of greater size in less time.

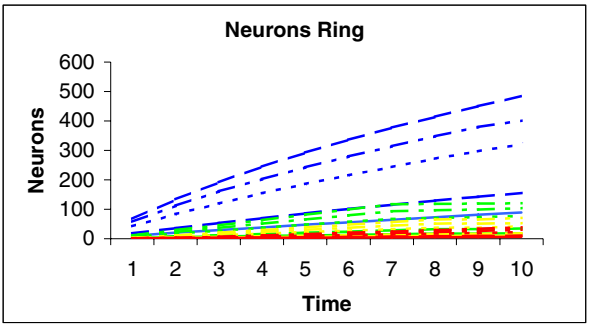


Fig. 7. Number of neurons inserted in 1 second time

Figure 8 shows topology preservation throughout the adaptation process of the network without establishing limits in the number of neurons, but with a time of 1 second as temporal limit.

Differences in topology preservation of the different options are not too significant when measuring with the topographic product. Nevertheless, topology preservation is lost when the number of neurons is high, because the number of input signals by iteration is insufficient to adapt all those neurons.

On the other hand, the topographic function shows differences in the topology preservation, indicating that the fastest networks have incorrect connections.

As in the case of a fixed number of neurons, we represent with a vertical line the necessary time to obtain good topology preservation with any of the different measures.

In general the results are quite unstable because in most of the cases only a few neurons had been inserted by the time first samples was taken.

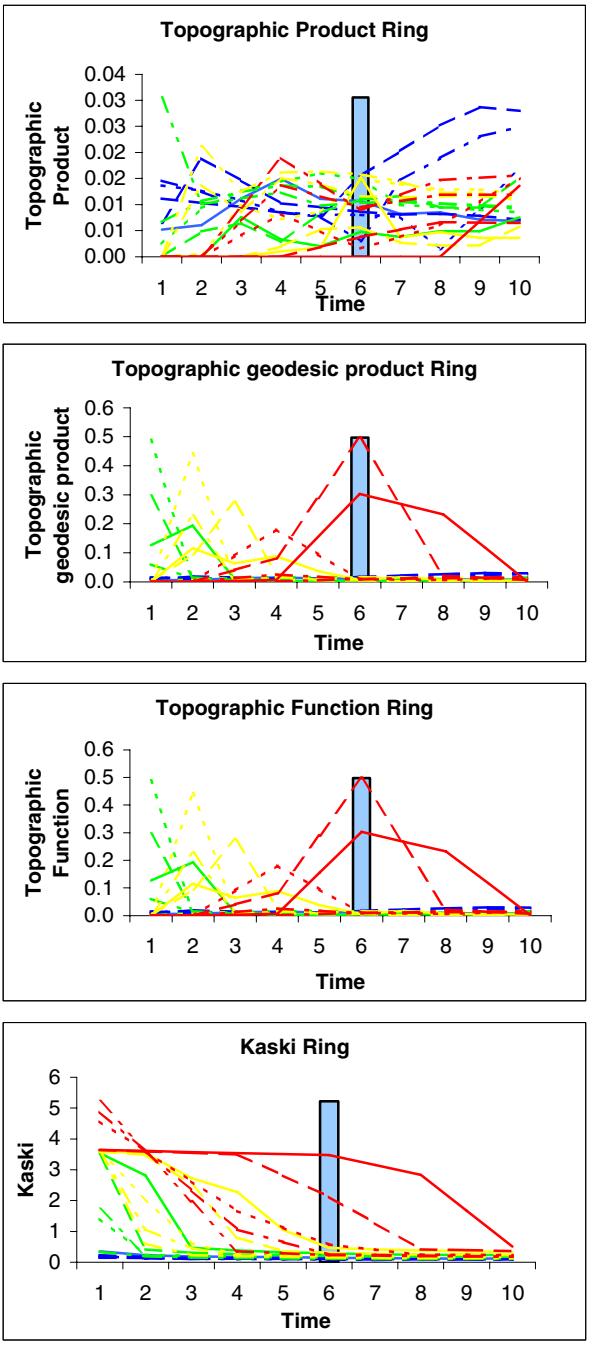


Fig. 8. Topology preservation depending on the available time

5 Conclusions and Further Work

Topology preservation of the Growing Neural Gas is affected by the learning parameters and available time. Faster methods, in many cases, deteriorate topology preservation, because the relation between number of neurons and input signals by iteration decreases.

From a practical point of view the study is interesting to define the necessary number of neurons or time to be used when trying to represent different objects with GNG keeping good topology preservation.

At the moment, we are doing similar studies with other self-organizing models (Neural Gas [11], GWR [12]), studying their degree of topology preservation. We want to extract which are the characteristics of these networks that allow a suitable and fast representation of an input space, in order to develop a new self-organizing neural network based on the combination of them.

References

1. Flórez, F., García, J.M., García, J., Hernández, A.: Representation of 2D Objects with a Topology Preserving Network. In Proceedings of the 2nd International Workshop on Pattern Recognition in Information Systems (PRIS'02), Alicante. ICEIS Press (2001) 267-276
2. Flórez, F., García, J.M., García, J., Hernández, A.: Hand Gesture Recognition Following the Dynamics of a Topology-Preserving Network. In Proc. of the 5th IEEE Intern. Conference on Automatic Face and Gesture Recognition, Washington, D.C. IEEE, Inc. (2001) 318-323
3. Fritzke, B.: A Growing Neural Gas Network Learns Topologies. In Advances in Neural Information Processing Systems 7, G. Tesauro, D.S. Touretzky y T.K. Leen (eds.), MIT Press (1995) 625-632
4. Kohonen, T.: Self-Organizing Maps. Springer-Verlag, Berlin Heidelberg (1995)
5. Martinetz, T., Schulten, K.: Topology Representing Networks. Neural Networks, 7(3) (1994) 507-522
6. Cheng, G., Zell, A.: Double Growing Neural Gas for Disease Diagnosis. In Proceedings of Artificial Neural Networks in Medicine and Biology Conference (ANNIMAB-1), Goteborg, Vol. 5. Springer (2000) 309-314
7. Bauer, H.-U., Pawelzik, K.R.: Quantifying the Neighborhood Preservation of Self-Organizing Feature Maps. IEEE Transactions on Neural Networks, 3(4) (1992) 570-578
8. Flórez, F., García, J.M., García, J., Hernández: Geodesic Topographic Product: An Improvement to Measure Topology Preservation of Self-Organizing Neural Networks. Lecture Notes in Computer Sciences, 3315. Springer, Berlin (2004) 841-850
9. Villmann, T., Der, R., Herrmann, M., Martinetz, T.M.: Topology Preservation in Self-Organizing Feature Maps: Exact Definition and Measurement. IEEE Transactions on Neural Networks, 8(2) (1997) 256-266
10. Kaski, S., Lagus, K.: Comparing Self-Organizing Maps. Lecture Notes in Computer Sciences, 1112. Springer, Berlin (1996) 809-814
11. Martinetz, T, Schulten, K.: A "Neural-Gas" Network Learns Topologies. In Artificial Neural Networks, T. Kohonen, K. Mäkisara, O. Simula y J. Kangas (eds.) (1991) 1:397-402
12. Marsland, S., Shapiro, J.; Nehmzow, U.: A self-organising network that grows when required. Neural Networks, 15 (2002) 1041-1058

A Fixed-Point Algorithm of Topographic ICA

Yoshitatsu Matsuda[1] and Kazunori Yamaguchi[2]

[1] Department of Integrated Information Technology,
Aoyama Gakuin University,
5-10-1, Fuchinobe, Sagamihara-shi, Kanagawa, 229-8558, Japan
matsuda@it.aoyama.ac.jp
http://www-haradalb.it.aoyama.ac.jp/~matsuda
[2] Department of General Systems Studies,
The University of Tokyo,
3-8-1, Komaba, Meguro-ku, Tokyo, 153-8902, Japan
yamaguch@graco.c.u-tokyo.ac.jp

Abstract. Topographic ICA is a well-known ICA-based technique, which generates a topographic mapping consisting of edge detectors from natural scenes. Topographic ICA uses a complicated criterion derived from a two-layer generative model and minimizes it by a gradient descent algorithm. In this paper, we propose a new simple criterion for topographic ICA and construct a fixed-point algorithm minimizing it. Our algorithm can be regarded as an expansion of the well-known fast ICA algorithm to topographic ICA, and it does not need any tuning of the stepsize. Numerical experiments show that our fixed-point algorithm can generate topographic mappings similar to those in topographic ICA.

1 Introduction

Independent component analysis (ICA) is a recently-developed method in the fields of signal processing and artificial neural networks, and has been shown to be quite useful for the blind separation problem [1,2,3,4]. The linear ICA is formalized as follows. Let $\boldsymbol{s} = (s_i)$ and $\boldsymbol{A}$ are N-dimensional source signals and an $N \times N$ mixing matrix, respectively. Then, an N-dimensional vector $\boldsymbol{x} = (x_i)$ of observed signals is defined as

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}. \tag{1}$$

The purpose is to find out $\boldsymbol{A}$ (or the inverse matrix $\boldsymbol{W}$) when only observed (mixed) signals are given. In other words, ICA blindly extracts source signals from observed signals as follows:

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}, \tag{2}$$

where $\boldsymbol{W} = (w_{ij})$ is an $N \times N$ mixing matrix to be estimated and $\boldsymbol{y} = (y_i)$ is an N-dimensional vector of estimated source signals. This is a typical ill-conditioned problem, but ICA can solve it by assuming that the source signals are generated

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 587–594, 2006.

according to independent and non-Gaussian probability distributions. In general, the ICA algorithms find out $\boldsymbol{W}$ by maximizing a criterion (called the contrast function) such as the higher-order statistics (e.g. the kurtosis) of each component of $\boldsymbol{y}$.

Topographic ICA is a well-known ICA-based technique, which can generate interesting topographic mappings of whitened components from natural scenes [5,6]. It assumes a two-layer generative model and minimizes a rather complicated contrast function by a gradient descent algorithm. Recently, we have proposed a simple criterion for topographic mappings by a novel information-theoretic approach named "InfoMin" [7,8,9]. In this paper, we propose a simple criterion based on InfoMin and construct a fixed-point algorithm minimizing the criterion. The new algorithm is derived in the similar way as fast ICA [10,11] and it can generate topographic mappings from natural scenes without the stepsize control.

This paper is organized as follows. In Section 2, topographic ICA and fast ICA are briefly explained. In Section 3, we propose a new contrast function for topographic ICA, then we construct a new fixed-point algorithm minimizing it. In Section 4, numerical experiments show that our new algorithm can generate topographic mappings from natural scenes in the similar way as topographic ICA. Lastly, this paper is concluded in Section 5.

2 Background

2.1 Topographic ICA

Topographic ICA [5] assumes that observed signals are given by a two-layer generative models and the variances of sources are dependent on each other through neighborhood functions. As a consequence, topographic ICA is given as the following update equation:

$$w_{ij} := w_{ij} + \alpha E\left(x_j y_i r_i\right) \tag{3}$$

where $E\left(u\right)$ is the expectation operator, α is the stepsize, $y_i = \sum_j w_{ij} x_j$, and

$$r_i = \sum_k h\left(i,k\right) g\left(\sum_j h\left(k,j\right) y_j^2\right). \tag{4}$$

$h\left(i,j\right)$ is a neighborhood function, and $g\left(u\right)$ is given as a nonlinear function such as $\tanh\left(u\right)$.

2.2 Fast ICA

Fast ICA [11] is a technique for minimizing a contrast function ϕ, which is given in the following form:

$$\phi = \sum_i E\left(G\left(y_i\right)\right) \tag{5}$$

where $G(u)$ is a nonlinear function such as u^4 or $\log(\cosh(u))$. In addition, it is assumed that $\boldsymbol{x}$ is whitened (which means that $E\left(\boldsymbol{x}\boldsymbol{x}^T\right)$ is the $N \times N$ identity matrix $\boldsymbol{I}$) and $\boldsymbol{W}$ is orthogonal. For each i, the update equation of fast ICA is derived as follows. Let $\boldsymbol{w}_i$ be the i-th row of $\boldsymbol{W}$. According to the Kuhn-Tucker conditions in the optima of $E\left(G\left(y_i = \boldsymbol{w}_i\boldsymbol{x}\right)\right)$ under the constraint $E\left(y_i^2\right) = \boldsymbol{w}_i\boldsymbol{w}_i^T = 1$, the optimal $\boldsymbol{w}_i$ needs to satisfy

$$E\left(\boldsymbol{x}g\left(y_i\right)\right) - \beta\boldsymbol{w}_i^T = 0 \tag{6}$$

where $g(u)$ is the derivative of $G(u)$ w.r.t u and β is a constant given as $E\left(y_i g\left(y_i\right)\right)$. Then, Eq. (6) is optimized by Newton's method. Its Jacobian matrix $\boldsymbol{J}$ w.r.t $\boldsymbol{w}_i$ is given as

$$\boldsymbol{J} = E\left(\boldsymbol{x}\boldsymbol{x}^T g'\left(y_i\right)\right) - \beta\boldsymbol{I} \tag{7}$$

where $g'(u)$ is the derivative of $g(u)$. The crucial approximation used in fast ICA is given as

$$E\left(\boldsymbol{x}\boldsymbol{x}^T g'\left(y_i\right)\right) \simeq E\left(\boldsymbol{x}\boldsymbol{x}^T\right) E\left(g'\left(y_i\right)\right) \simeq E\left(g'\left(y_i\right)\right)\boldsymbol{I}. \tag{8}$$

Thus, because the Jacobian matrix $\boldsymbol{J}$ becomes diagonal, the inversion of $\boldsymbol{J}$ is easily calculated. In consequence, the following update equation based on Newton's method is derived:

$$\boldsymbol{w}_i := E\left(\boldsymbol{x}^T g\left(y_i\right)\right) - E\left(g'\left(y_i\right)\right)\boldsymbol{w}_i \tag{9}$$

where $\boldsymbol{w}_i$ is normalized by $\boldsymbol{w}_i := \frac{\boldsymbol{w}_i}{\sqrt{\boldsymbol{w}_i\boldsymbol{w}_i^T}}$ at each update. Note that Eq. (9) does not depend on β any longer. In the frequently-used deflation approach, each $\boldsymbol{w}_i$ is estimated one by one under the constraint that $\boldsymbol{w}_i$ is orthogonal to previously estimated $\boldsymbol{w}_p$'s $(p < i)$. By contrast, in the symmetrical approach, every $\boldsymbol{w}_i$ is simultaneously updated by Eq. (9) and then $\boldsymbol{W}$ is orthonormalized by

$$\boldsymbol{W} := \left(\boldsymbol{W}\boldsymbol{W}^T\right)^{-\frac{1}{2}}\boldsymbol{W}. \tag{10}$$

In our algorithm in Section 3, the symmetrical approach is employed.

3 Fixed-Point Algorithm

3.1 New Criterion

We have previously proposed the InfoMin principle in [7,8,9], and the following criterion η was derived:

$$\eta = -\sum_{\omega\in\Omega}\left(\sum_{i\in\omega}\left(E\left(y_i^4\right) - 3\right)^2 + 3\sum_{i\in\omega}\sum_{j\in\omega, j\neq i}\left(E\left(y_i^2 y_j^2\right) - 1\right)^2\right) \tag{11}$$

where $\boldsymbol{y}$ is whitened, and every component y_i is placed on a two-dimensional array. ω is a neighborhood area on the array and Ω is a set of ω. It was shown that

topographic mappings could be generated from natural scenes by the criterion η. See [9] for the details of InfoMin. η is regarded as a basic criterion for forming topographic mappings in this paper. Eq. (11) is rewritten as

$$\eta = -\sum_i a_i \left(E\left(y_i^4\right) - 3\right)^2 - \sum_{i,j\neq i} b_{ij} \left(E\left(y_i^2 y_j^2\right) - 1\right)^2 \tag{12}$$

where a_i and b_{ij} are constants depending on Ω. Because it is difficult to deal with $\left(E\left(u\right)\right)^2$ directly, it is first assumed that all sources are super-gaussian. It is easily shown that $\left(E\left(y_i^4\right) - 3\right)$ and $\left(E\left(y_i^2 y_j^2\right) - 1\right)$ are always positive under this assumption. Next, $\eta\ (< 0)$ is replaced with $-\sqrt{-\eta}$. Then, $-\sqrt{-\eta}$ is approximated as

$$-\sqrt{-\eta} \simeq -\sum_i \sqrt{a_i} \left(E\left(y_i^4\right) - 3\right) - \sum_{i,j\neq i} \sqrt{b_{ij}} \left(E\left(y_i^2 y_j^2\right) - 1\right). \tag{13}$$

Lastly, by generalizing y_i^4 in the first term, the following contrast function ψ is given:

$$\psi = -\sum_i c_i E\left(G\left(y_i\right)\right) - \sum_{i,j\neq i} h\left(i,j\right) E\left(y_i^2 y_j^2\right) \tag{14}$$

where c_i is a constant and $h\left(i,j\right)$ is a neighborhood function determined by Ω.

3.2 Derivation of Algorithm

Here, a fixed-point algorithm minimizing ψ in Eq. (14) is derived in the same way as for ϕ in fast ICA (see Section 2.2). By replacing ϕ with ψ, Eqs. (6), (7), and (8) are modified as follows:

$$E\left(\boldsymbol{x}\left(c_i g\left(y_i\right) + 2\sum_{j\neq i} h\left(i,j\right) y_i y_j^2\right)\right) - \beta \boldsymbol{w}_i^T = 0, \tag{15}$$

$$\boldsymbol{J} = E\left(\boldsymbol{x}\boldsymbol{x}^T\left(c_i g'\left(y_i\right) + 2\sum_{j\neq i} h\left(i,j\right) y_j^2\right)\right) - \beta \boldsymbol{I}, \tag{16}$$

and

$$\begin{aligned} E\left(\boldsymbol{x}\boldsymbol{x}^T\left(c_i g'\left(y_i\right) + 2\sum_{j\neq i} h\left(i,j\right) y_j^2\right)\right) &\simeq E\left(c_i g'\left(y_i\right) + 2\sum_{j\neq i} h\left(i,j\right) y_j^2\right)\boldsymbol{I} \\ &= E\left(c_i g'\left(y_i\right) + 2\sum_{j\neq i} h\left(i,j\right)\right)\boldsymbol{I} \end{aligned} \tag{17}$$

where y_i^2 is removed because of $E\left(y_i^2\right) = 1$. Thus, the following update equation is derived:

$$\boldsymbol{w}_i := E\left(\boldsymbol{x}^T\left(c_i g\left(y_i\right) + 2\sum_{j\neq i} h\left(i,j\right) y_i y_j^2\right)\right) - E\left(c_i g'\left(y_i\right) + 2\sum_{j\neq i} h\left(i,j\right)\right)\boldsymbol{w}_i. \tag{18}$$

Because every y_i depends on each other through the neighborhood function $h(i,j)$ in this algorithm, the symmetrical approach is employed for updating $\boldsymbol{W}$. In other words, Eq. (18) is calculated simultaneously for every i and then $\boldsymbol{W}$ is orthonormalized by Eq. (10).

3.3 Discussion

The computation at each update in the above fixed-point algorithm needs some matrix manipulations, the dominant ones of which are multiplications of an $N \times N$ matrix and an $N \times M$ one (M is the number of samples). It is easily shown that the computational complexity of each update in our fixed-point algorithm is the same as those in topographic ICA (Section 2.1) and the symmetrical approach of fast ICA (Section 2.2). Therefore, the estimation of the number of updates is crucial for comparing these algorithms. Because both our algorithm and fast ICA are approximations of Newton's method, they are expected to rapidly converge to minima. On the other hand, it is expected that the convergence of topographic ICA using a simple gradient algorithm is slower. In addition, it is a significant advantage in practical applications that our fixed-point algorithm does not need any additional techniques for the stepsize control.

It is also worthwhile to examine whether this fixed-point approach can be applied directly to the original topographic ICA. It is difficult because topographic ICA utilizes a convolution to incorporate a neighborhood function with ICA. This convolution seems to be derived necessarily from the two-layer generative model. On the other hand, our algorithm avoids this problem by using ψ in Eq. (14), where a typical contrast function ($\sum_i c_i E(G(y_i))$) is separated from the term depending on the neighborhood function ($\sum_{i,j \neq i} h(i,j) E\left(y_i^2 y_j^2\right)$).

4 Results

Here, our fixed-point algorithm is applied to processing natural scenes. The results are shown in Fig. 1. The three functions u^3, $\tanh(u)$, and $u \exp(-\frac{u^2}{2})$ were given as $g(u)$ in our method. Though the initial $\boldsymbol{W}$ was given randomly, the same initial $\boldsymbol{W}$ was used for each $g(u)$. 1000 updates by Eq. (18) were done. At the final stage of updates, the rate of the fluctuation of $\boldsymbol{W}$ was less than 10^{-5} for every $g(u)$. Each experiment took about 4 hours with 2.8GHz CPU. For comparison, topographic ICA was done for the same natural scenes.

Figs 1-(a), (b), and (c) show that our fixed-point algorithm could form topographic mappings where edge detectors with similar orientation preferences are nearer. Though there is no obvious difference among them, it seems that the topography in Fig 1-(a) is weaker than Figs 1-(b) and (c). The mapping by topographic ICA is also shown in Fig. 1-(d). Topographic ICA generated a topographic mapping of distinct but short edge filters. On the other hand, the edge filters in our methods are noisy but longer.

It is interesting that a topographic mapping was formed even if $g(u) = u^3$, for it is known to be difficult for such a cumulant-based ICA algorithm to extract

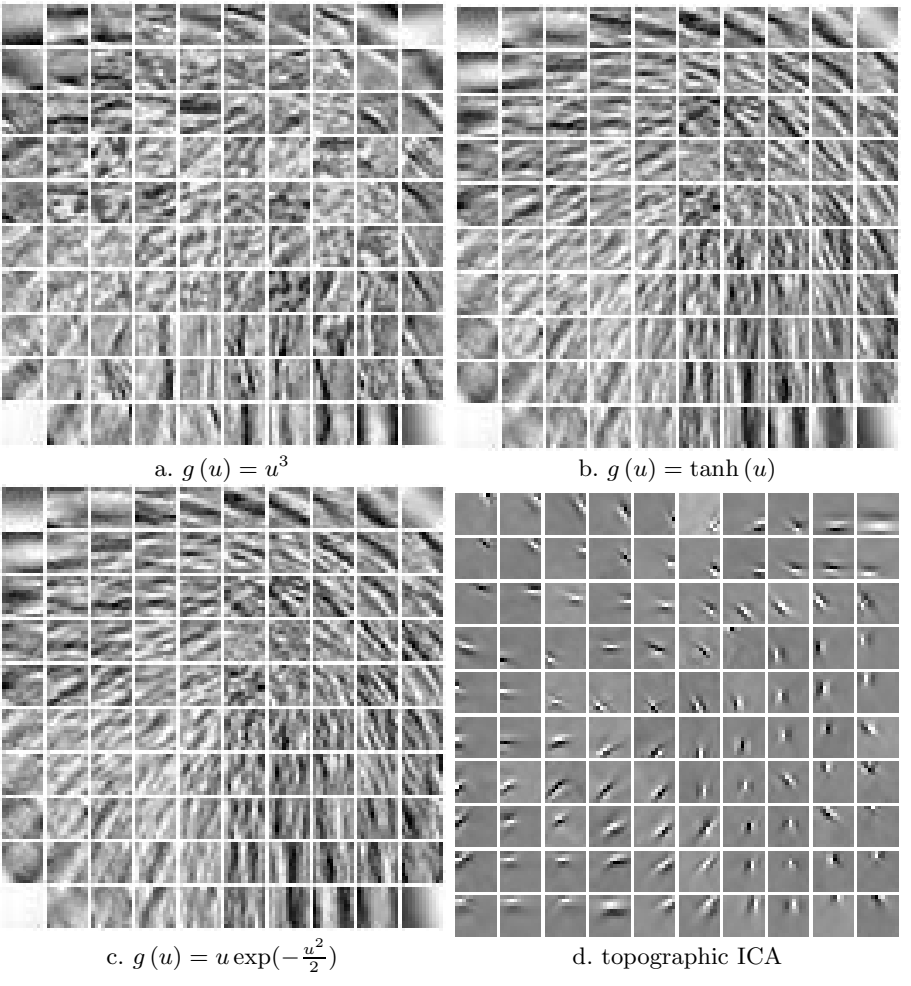


Fig. 1. Formation of topographic mappings: Here, 100 PCA-whitened components of 30000 samples of natural scenes of 12×12 pixels were used as $\boldsymbol{x}$. Every y_i is placed on 10×10 array. Ω was given as the set of all the areas of 5×5 components over the array. 1000 updates were done for each $g(u)$. The estimated mixing matrix $\boldsymbol{A}$ is visualized. (a): A topographic mapping generated by our fixed-point algorithm with $g(u) = u^3$. (b) Fixed-point topographic ICA with $g(u) = \tanh(u)$. (c) Fixed-point topographic ICA with $g(u) = u\exp(-\frac{u^2}{2})$. (d) A mapping generated by topographic ICA with a 3×3 neighborhood ones matrix and 10000 updates.

edge detectors from natural scenes. On the other hand, an obvious disadvantage of our algorithm is that every result was rather noisy. Though it is well known that cumulant-based ICA is quite sensitive to outliers, the results were also noisy even if robust nonlinear functions such as $g(u) = \tanh(u)$ were used.

This noise could not be removed by increasing the number of updates. In order to generate noiseless results, we may have to also replace $E\left(y_i^2 y_j^2\right)$ with some robust functions.

5 Conclusion

In this paper, we proposed a new fixed-point algorithm of topographic ICA. First, we proposed a new criterion ψ which is the sum of a typical contrast function and the terms based on a neighborhood function. Then, we derived a fixed-point algorithm minimizing ψ, which is an extension of fast ICA. Numerical experiments showed that our methods could generate topographic mappings similar to those in topographic ICA.

Our algorithm currently lacks the theoretical foundations of the definition of the criterion ψ. Though it is roughly based on the InfoMin principle, rather coarse approximations are applied. Its relation to the model of the original topographic ICA is also unclear. So, further theoretical analysis would be needed. Besides, numerical experiments show that the results of our algorithm were noisy even if a robust $g(u)$ was utilized. So, we may have to replace $E\left(y_i^2 y_j^2\right)$ with some robust functions as well. It is easily shown that our method is applicable even if $E\left(y_i^2 y_j^2\right)$ is extended to a form $f(y_i) f(y_j)$, where $f(u)$ is an arbitrary differentiable function. We are now trying to find a useful $f(u)$.

References

1. Jutten, C., Herault, J.: Blind separation of sources (part I): An adaptive algorithm based on neuromimetic architecture. Signal Processing **24** (1991) 1–10
2. Comon, P.: Independent component analysis - a new concept? Signal Processing **36** (1994) 287–314
3. Bell, A.J., Sejnowski, T.J.: An information-maximization approach to blind separation and blind deconvolution. Neural Computation **7** (1995) 1129–1159
4. Cardoso, J.F., Laheld, B.: Equivariant adaptive source separation. IEEE Transactions on Signal Processing **44** (1996) 3017–3030
5. Hyvärinen, A., Hoyer, P.O., Inki, M.: Topographic independent component analysis. Neural Computation **13** (2001) 1527–1558
6. 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** (2001) 2413–2423
7. Matsuda, Y., Yamaguchi, K.: The InfoMin principle: a unifying information-based criterion for forming topographic mappings. In: ICONIP2001 Proceedings, Shanghai, China (2001) 14–19
8. Matsuda, Y., Yamaguchi, K.: The InfoMin criterion: an information theoretic unifying objective function for topographic mappings. In: Artificial Neural Network and Neural Information Processing - ICANN/ICONIP 2003. Volume 2714 of LNCS., Istanbul, Turkey, Springer-Verlag (2003) 401–408

9. Matsuda, Y., Yamaguchi, K.: The infomin principle for ica and topographic mappings. In: Independent Component Analysis and Blind Signal Separation, 6th International Conference, ICA 2006. Volume 3889 of LNCS., Charleston, SC, USA, Springer-Verlag (2006) 958–965
10. Hyvärinen, A., Oja, E.: A fast fixed-point algorithm for independent component analysis. Neural Computation **9** (1997) 1483–1492
11. Hyvärinen, A.: Fast and robust fixed-point algorithms for independent component analysis. IEEE Transactions on Neural Networks **10** (1999) 626–634

Image Compression by Vector Quantization with Recurrent Discrete Networks

Domingo López-Rodríguez[1], Enrique Mérida-Casermeiro[1],
Juan M. Ortiz-de-Lazcano-Lobato[2], and Ezequiel López-rubio[2]

[1] Department of Applied Mathematics
University of Málaga
Málaga, Spain
{dlopez, merida}@ctima.uma.es
[2] Department of Computer Science and Artificial Intelligence
University of Málaga
Málaga, Spain
{jmortiz, ezeqlr}@lcc.uma.es

Abstract. In this work we propose a recurrent multivalued network, generalizing Hopfield's model, which can be interpreted as a vector quantifier. We explain the model and establish a relation between vector quantization and sum-of-squares clustering. To test the efficiency of this model as vector quantifier, we apply this new technique to image compression. Two well-known images are used as benchmark, allowing us to compare our model to standard competitive learning. In our simulations, our new technique clearly outperforms the classical algorithm for vector quantization, achieving not only a better distortion rate, but even reducing drastically the computational time.

1 Introduction

Compressing an image is a significantly different task than compressing raw binary data. Although general purpose compression techniques can be used to compress images, the result is less than optimal. The reason is that images have certain statistical properties which in turn may be exploited by encoders specifically designed for this task. Also, some of the finer details in the image can be sacrificed for the sake of saving a little more bandwidth or storage space. This fact also means that lossy compression techniques can be used in this area.

Lossless compression involves with compressing data which, when decompressed, will be an exact replica of the original data. Lossless compression is applied to binary data as executables or documents, which need to be exactly reproduced when decompressed. On the other hand, images need not to be reproduced exactly in their original form, but an approximation of the original image is enough for most purposes, as long as the error, obtained in the compression phase, between the original and the reproduced image is tolerable.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 595–605, 2006.

Some error measures, commonly used in image compression, are:

- The mean square error (MSE), given by:

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} [I(i,j) - I'(i,j)]^2$$

  where I is the original image, I' is the approximated version (which is actually the decompressed image) and M, N are the dimensions of the images. A lower value for MSE means lesser error.
- The Peak Signal to Noise Ratio (PSNR), given by:

$$PSNR = 20 \log_{10} \left(\frac{255}{\sqrt{MSE}} \right)$$

  which achieves high value when MSE is low. So a good technique will obtain a high value for PSNR.
- The mean distortion, used in Vector Quantization (VQ), which will be defined in the next section.

We can use any of these three error measures to quantify the goodness of a compression technique. In the present work, we use the mean distortion measure, since it is more appropriate when dealing with VQ.

According to Egmont-Petersen et al. [5], two different types of image compression approaches with neural networks (ANNs) can be identified: direct pixel-based encoding-decoding by one ANN [2,7,16,17] and pixel-based encoding-decoding based on a modular approach [3,4,12,18,20,21]. Different types of ANNs have been trained to perform image compression: feed-forward networks [3,4,16,17,18,20,21], Kohonen Self-Organizing Maps (SOMs) [2,7], adaptive fuzzy leader clustering (a fuzzy ART-like approach) [12], a learning vector quantifier [21] and a radial basis function network [16].

Other approaches are based on competitive neural networks. The aim of competitive neural networks is to cluster the input vectors and it can be used for data coding and compression through vector quantization. It has been shown that competitive learning is an appropriate algorithm for VQ of unlabeled data. Ahalt, Krishnamurthy and Chen [1] discussed the application of competitive learning neural networks to VQ and developed a new training algorithm for designing VQ codebooks which yields near-optimal results and can be used to develop adaptive vector quantifiers. Yair, Zeger and Gersho [22] have proposed a deterministic VQ design algorithm, called the soft competition scheme, which updates all the codevectors simultaneously with a step size that is proportional to its probability of winning. In [15], Pal, Bezdek and Tsao proposed a generalization of learning VQ for clustering which avoids the necessity of defining an update neighbourhood scheme and the final centroids do not seem sensitive to initialization. Ueda and Nakano presented a new competitive learning algorithm with a selection mechanism based on the equidistortion principle for designing

optimal vector quantizers [19]. The selection mechanism enables the system to escape from local minima.

Recently, Muñoz-Perez et al. [13] proposed an expansive and competitive learning for VQ capable to avoid local minima of the distortion function, and presented some optimality conditions for the set of codewords.

ANN approaches have to compete with well-established compression techniques such as JPEG, which should serve as a reference. The major advantage of ANNs is that their parameters are adaptable, which may give better compression rates when trained on specific image material. However, such a specialization becomes a drawback when novel types of images have to be compressed.

In this work, we propose a vector quantization approach to image compression by means of a discrete recurrent model, comparing its efficiency to that of the classical competitive learning.

2 Vector Quantization and Competitive Learning

A vector quantifier of dimension d and size K is a mapping Q from the d-dimensional Euclidean space $\mathbb{R}^d$ into a finite subset $C = \{\boldsymbol{c_1}, \ldots, \boldsymbol{c_K}\}$ of $\mathbb{R}^d$ containing K output or representative vectors, called code vectors, reference vectors, reproduction vectors, prototypes or codewords. The collection of all possible reproduction vectors is called the reproduction alphabet or more commonly the codebook. Hence, the input vector space, $\mathbb{R}^d$, is divided into K disjoint regions, $C_1, \ldots, C_K$, where

$$C_k = \{\boldsymbol{x} \in \mathbb{R}^d : Q(\boldsymbol{x}) = \boldsymbol{c_k}\}$$

All inputs vectors in C_k are approximated by $\boldsymbol{c_k}$. The cost introduced by this approximation is given by a nonnegative distortion measure, usually the Euclidean distance between $\boldsymbol{x}$ and the corresponding $\boldsymbol{c_k} = Q(\boldsymbol{x})$.

For a finite training set, $X = \{\boldsymbol{x_1}, \ldots, \boldsymbol{x_N}\}$, the vector quantization is a combinatorial problem that attempts to represent X (with large information contents) by a reduced set of codewords C. In other words, the goal is to select a set C of codewords such that the mean distortion function:

$$D(C) = \frac{1}{N} \sum_{k=1}^{K} \sum_{i|\boldsymbol{x_i} \in C_k} ||\boldsymbol{x_i} - \boldsymbol{c_k}||^2 \tag{1}$$

is minimum. This distortion function is generally not convex.

The standard competitive learning algorithm is a stochastic gradient descent approach to minimize this function. It consists in:

1. Selecting a point $\boldsymbol{x} \in X$ and determining $\boldsymbol{c_k} = Q(\boldsymbol{x})$.
2. Updating $\boldsymbol{c_k}$ with the rule $\Delta \boldsymbol{c_k} = \alpha_n(\boldsymbol{x} - \boldsymbol{c_k})$, where α_n is the learning rate at the n-th training epoch.
3. Repeat the previous points until a maximum of training epochs is reached or convergence is detected.

With this algorithm, it is guaranteed that $\boldsymbol{c_k}$ is the centroid of C_k, and it is the best representative vector of C_k.

3 The MREM Model

Let us consider a recurrent neural network formed by N neurons, where the state of each neuron $i = 1, \ldots, N$ is defined by its output s_i taking values in any finite set $\mathcal{M} = \{m_1, m_2, \ldots, m_L\}$. This set does not need to be numerical.

The state of the network, at time t, is given by a N-dimensional vector, $\boldsymbol{S}(t) = (s_1(t), s_2(t), \ldots, s_N(t)) \in \mathcal{M}^N$. Associated to every state vector, an energy function, is defined:

$$E(\boldsymbol{S}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij} f(s_i, s_j) + \sum_{i=1}^{N} \theta_i(s_i) \tag{2}$$

where $w_{i,j}$ is the weight of the connection from the j-th neuron to the i-th neuron, $f : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ can be considered as a measure of similarity between the outputs of two neurons, usually verifying the conditions mentioned in [9]:

1. For all $x \in \mathcal{M}$, $f(x, x) = c \in \mathbb{R}$.
2. f is a symmetric function: for every $x, y \in \mathcal{M}$, $f(x, y) = f(y, x)$.
3. If $x \neq y$, then $f(x, y) \leq c$.

and $\theta_i : \mathcal{M} \to \mathbb{R}$ are the threshold functions. Since thresholds will not be used for image compression, therefore we will consider θ_i to be the zero function for all $i = 1, \ldots, N$.

The introduction of this similarity function provides, to the network, of a wide range of possibilities to represent different problems [9,10]. So, it leads to a better and richer (giving more information) representation of problems than other multivalued models, as SOAR and MAREN [6,14], since in those models most of the information enclosed in the multivalued representation is lost by the use of the signum function that only produces values in $\{-1, 0, 1\}$.

If function $f(x, y) = 2\delta_{x,y} - 1$, which equals 1 if and only if its two parameters coincide, and -1 in the rest of cases, is used and $\mathcal{M} = \{-1, 1\}$, MREM reduces to Hopfield's bipolar model (BH) [8]. So, MREM is a powerful generalization of BH and other multivalued models, because it is capable of representing the information more accurately than those models.

The energy function characterizes the dynamics of the net, as happened in BH. In every instant, the net evolves to reach a state of lower energy than the current one.

In this work, we have considered discrete time and semi-parallel dynamics, where only one neuron is updated at time t. The next state of the net will be the one that achieves the greatest descent of the energy function by changing only one neuron output.

Let us consider a total order in $\mathcal{M}$. The potential increment when p-th neuron changes its output from s_p to $l \in \mathcal{M}$ at time t, is

$$U_p(l) = -\Delta E = \frac{1}{2}\sum_{i=1}^{N} [w_{p,i} f(l, s_i(t)) + w_{i,p} f(s_i(t), l) -$$

$$-(w_{p,i}f(s_p(t), s_i(t)) + w_{i,p}f(s_i(t), s_p(t)))] - \frac{1}{2}w_{p,p}[f(l,l) - f(s_p(t), s_p(t))] \quad (3)$$

If f verifies the similarity conditions and if matrix W is symmetric and $w_{p,p} = 0$ (as in the case of the problem studied in this paper, it will be made clearer in the next section), then the *reduced potential increment* is obtained:

$$U_p^*(l) = \frac{1}{2}\sum_{j=1}^{N} w_{p,j}\,[f(s_p, s_j) - f(l, s_j)] \quad (4)$$

We use the following updating rule for the neuron outputs:

$$s_p(t+1) = \begin{cases} l, & \text{if } U_a(l) \geq U_q(k)\,\forall k \in \mathcal{M} \text{ and } \forall q \in \{1, \ldots, N\} \\ s_p(t), & \text{otherwise} \end{cases} \quad (5)$$

This means that each neuron computes in parallel the value of a L-dimensional vector of potentials, related to the energy decrement produced if the neuron state is changed. The only neuron changing its current state is the one producing the maximum decrease of energy.

It has been proved that the MREM model with this dynamics always converges to a minimal state [9]. This result is particularly important when dealing with combinatorial optimization problems, where the application of MREM has been very fruitful [9,10].

4 Two-Stage Image Compression with MREM

In this section we will describe the two-stage VQ algorithm that uses the multi-valued model MREM in its first phase.

4.1 Clustering with MREM

In the first stage, MREM is used to obtain a good clustering of the input pattern set.

In order to apply MREM at this step, this clustering problem must be formulated as an optimization task.

Although there are lots of possible formulations for this clustering problem, one of the most used formulations consists in minimizing the sum of intra-cluster distances, that is, if $X = \{\boldsymbol{x_1}, \ldots, \boldsymbol{x_N}\}$ is the pattern set to be clustered into K groups, we look forward to minimizing the quantity:

$$d = \sum_{k=1}^{K} \sum_{i|\boldsymbol{x_i} \in C_k} \sum_{j|\boldsymbol{x_j} \in C_k} ||\boldsymbol{x_i} - \boldsymbol{x_j}||$$

which is the sum of the distances between patterns in the same cluster. With this formulation, we will obtain homogeneous clusters.

The above expression can be easily re-written as

$$d = \sum_{i=1}^{N} \sum_{j=1}^{N} ||\boldsymbol{x_i} - \boldsymbol{x_j}|| \rho_{\boldsymbol{x_i},\boldsymbol{x_j}} \tag{6}$$

where $\rho_{\boldsymbol{x},\boldsymbol{y}}$ equals 1 if and only if $\boldsymbol{x}$ and $\boldsymbol{y}$ belong to the same cluster, otherwise it will be 0.

This new expression can be used to an energy function for the MREM model.

Thus, let us consider a neural network with N neurons. The output s_i of the i-th neuron belongs to the set $\mathcal{M} = \{1, \ldots, K\}$, meaning that the pattern $\boldsymbol{x_i}$ is assigned to the s_i-th cluster.

If we compare Eq. (2) and Eq. (6), and taking into account that θ_i is the zero function for all i, we can obtain the value for the synaptic weights $w_{i,j}$ and an appropriate definition of the similarity function.

This comparison leads us to define:

$$w_{i,j} = -2||\boldsymbol{x_i} - \boldsymbol{x_j}||$$

and

$$f(a,b) = \delta_{a,b} = \begin{cases} 1, \text{ if } a = b \\ 0, \text{ otherwise} \end{cases}$$

So, the energy function will be as follows:

$$E = \sum_{i=1}^{N} \sum_{j=1}^{N} ||\boldsymbol{x_i} - \boldsymbol{x_j}|| \delta_{s_i,s_j} \tag{7}$$

that is, the sum of intra-cluster distances.

In order to minimize this energy function, we propose semi-parallel dynamics for the network, as mentioned before:

- In parallel, each neuron computes a vector of reduced potential increments, $\boldsymbol{V_p} = (U_p^*(1), \ldots, U_p^*(K))$, by using Eq. (4), which in this case is

$$U_p^*(l) = \frac{1}{2} \sum_{j=1}^{N} ||\boldsymbol{x_p} - \boldsymbol{x_j}|| \left[\delta_{s_p,s_j} - \delta_{l,s_j}\right]$$

- Each neuron computes in parallel the maximum potential in its corresponding $\boldsymbol{V_p}$. It will be stored in $v_p = \max(\boldsymbol{V_p})$ and n_p will be the value of $l \in \{1, \ldots, K\}$ which produces the maximum potential increment in $\boldsymbol{V_p}$.
- The scheduling selects the neuron q for which $v_q \geq v_p$ for all $p \in \{1, \ldots, N\}$, and updates its output according to $s_q = n_q$. This last step is not made in parallel.

With this dynamics, the energy function is minimized and therefore a clustering of the input pattern space is obtained.

4.2 Computation of the Codebook

In this second stage, we use the recently obtained clustering to compute the set of codewords.

As we want the mean distortion, given by Eq. (1), to be minimized, we compute $\boldsymbol{c_k}$ as the centroid of the k-th cluster C_k, that is,

$$\boldsymbol{c_k} = \frac{1}{N_k} \sum_{i|\boldsymbol{x_i} \in C_k} \boldsymbol{x_i}$$

where N_k is the number of patterns that belong to C_k.

So, we have guaranteed the (local) optimality of the codebook.

(a) (b)

Fig. 1. Test images used in this work: (a) cameraman, (b) lenna

5 Experimental Results

Two well-known images have been used in this work to show the efficiency of the proposed technique: cameraman and lenna, see Fig 1.

The size of these images was 256x256 pixels, with 256 graylevels. Each image was divided into windows of size $L \in \{8, 10, 12, 16\}$, obtaining a total of $\frac{256^2}{L^2}$ windows. Every window is represented by a L^2-dimensional vector.

Every component of these vectors is normalized to avoid the negative effect of a bad scaling.

This set of vectors is then clustered to obtain $K \in \{16, 32\}$ prototypes and the mean distortion is measured. The results of mean distortion achieved in these experiments are shown in Tables 1 and 2.

In these Tables, a comparison with Standard Competitive Learning (SCL) is made. The learning rate α_n of SCL decreased from 0.9 to 0.01 for 100 training epochs, and 10 executions were performed for each image and algorithm. Columns labeled Min. and Av. show the minimum and average mean distortion

Table 1. Mean distorsion for cameraman image

			MREM			SCL			
L	K	N	Min.	Av.	t	Min.	Av.	t	Impr.
16	16	256	5.31	5.35	0.1946	14.65	14.75	16.4758	175.7%
16	32	256	4.68	4.78	0.3860	14.61	14.69	29.9586	207.3%
12	16	441	3.52	3.56	0.4532	10.79	10.85	20.2672	204.7%
12	32	441	3.19	3.21	0.8867	10.66	10.76	39.5268	235.2%
10	16	625	2.76	2.81	1.1248	8.80	8.88	27.2470	216.0%
10	32	625	2.47	2.50	1.9042	8.84	8.91	50.5668	256.4%
8	16	1024	2.07	2.08	3.7546	6.94	7.02	47.9990	237.5%
8	32	1024	1.86	1.88	7.7969	6.89	6.94	64.9956	269.1%

Table 2. Mean distorsion for lenna image

			MREM			SCL			
L	K	N	Min.	Av.	t	Min.	Av.	t	Impr.
16	16	256	7.12	7.19	0.1725	16.37	16.50	20.2954	129.4%
16	32	256	6.33	6.37	0.3377	16.28	16.41	35.6092	157.6%
12	16	441	4.69	4.72	0.4157	12.15	12.23	22.2540	159.1%
12	32	441	4.12	4.14	0.8644	12.08	12.13	45.0909	192.9%
10	16	625	3.72	3.75	0.8182	9.84	9.99	27.6838	166.4%
10	32	625	3.24	3.26	1.7484	9.86	9.94	46.1765	204.9%
8	16	1024	2.66	2.67	3.2199	7.75	7.82	46.6706	192.8%
8	32	1024	2.32	2.34	7.9586	7.75	7.78	76.5742	232.4%

achieved by the two algorithms. Columns labeled t contain the time spent by each of them. In the last column, Impr., a measure of the improvement achieved by MREM over SCL:

$$\text{Impr.} = \frac{\text{Av}_{SCL} - \text{Av}_{MREM}}{\text{Av}_{MREM}} \cdot 100$$

It is remarkable that MREM highly outperforms SCL on average quality in all cases, achieving improvements of about 150-200%. The time spent by MREM is also a fraction of the spent by SCL. So, MREM is much more efficient than SCL.

In order to show the efficiency of this technique, we have made a simulation in which $L = 4$. If $K = 32$ representatives are used, and $L = 4$, then 128 bits are needed to represent each window, but only 5 to represent the codewords, so we may obtain a compression rate of 128 to 5, that is, 25 to 1 approximately. By using JPG compression, we obtained 45Kb for the original cameraman image, 34Kb for the SCL-compressed and 29Kb for the MREM-compressed. For lenna image, these quantities were 43, 32 and 35Kb, respectively. In Fig. 2, the compressed images, obtained by both techniques, are shown.

(a) (b)

Fig. 2. Compressed test images with $L = 4$ and $K = 32$: (a) by using Standard Competitive Learning (distortions=3.18 and 3.56, from up to down) and (b) by using MREM (distortions=0.67 and 0.81, respectively)

6 Conclusions

In this work we have proposed an alternative method to competitive learning in vector quantization tasks.

This approach is based on a multivalued recurrent network suitable for combinatorial optimization problems, as proved in other works. The intrinsic semi-parallelism provided by this model improves the efficiency of the net when compared to SCL, since the time consumption is drastically reduced. We have applied this approach to image compression, achieving great advantages over SCL, not only on computational time, but even on quality of the quantization, obtaining improvements above 100%. One of the reasons for this improvement is that our algorithm divides the entire task of vector quantization into a two-stage problem: first, it finds a (locally) optimal clustering of the input pattern space, and then it computes the optimal codebook associated to the given

partition. Our future work in this problem consists in finding new formulations to help MREM avoid local minima in the clustering task, which will lead to an improvement of the quantization results.

References

1. S.C. Ahalt, A.K. Krishnamurthy, P. Chen, and D.E. Melton, *Competitive learning algorithms for vector quantization*, Neural Networks, **3**, 277-290, 1990.
2. C. Amerijckx, M. Verleysen, P. Thissen et al., *Image compression by self-organized Kohonen map*, IEEE Trans. Neural Networks **9(3)**, 503–507, 1998.
3. J.G. Daugman, *Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression*, IEEE Trans. Acoustics, Speech Signal Process. **36(7)**, 1169-1179, 1988.
4. R.D. Dony, S. Haykin, *Optimally adaptive transform coding*, IEEE Trans. Image Process. **4(10)**, 1358-1370, 1995.
5. M. Egmont-Petersen, D. de Ridder and H. Handels, *Image processing with neural networksa review*, Pattern Recognition **35**, 2279-2301, 2002.
6. M. H. Erdem and Y. Ozturk, *A New family of Multivalued Networks*, Neural Networks **9,6**, 979–989, 1996.
7. G. Hauske, *A self organizing map approach to image quality*, Biosystems **40(1-2)**, 93-102, 1997.
8. J.J. Hopfield, *Neural networks and physical systems with emergent collective computational abilities*, Proc. of National Academy of Sciences USA, **79**, 2254–2558, 1982.
9. E. Mérida Casermeiro, *Red Neuronal recurrente multivaluada para el reconocimiento de patrones y la optimización combinatoria*, Ph. D. dissertation (in Spanish). Univ. Málaga, España, 2000.
10. E. Mérida Casermeiro, J. Muñoz Pérez and R. Benítez Rochel, *A recurrent multivalued neural network for the N-queens problem*, Lecture Notes in Computer Science **2084**, 522–529, 2001.
11. E. Mérida-Casermeiro and D. López-Rodríguez, *Multivalued Neural Network for Graph MaxCut Problem*, Lecture Series on Computer and Computational Sciences, **1**, 375–378, 2004.
12. S. Mitra and S.Y. Yang, *High fidelity adaptive vector quantization at very low bit rates for progressive transmission of radiographic images*, J. Electron. Imaging **8(1)**, 23-35, 1999.
13. J. Muñoz-Perez, J.A. Gomez-Ruiz, E. Lopez-Rubio and M.A. Garcia-Bernal, *Expansive and Competitive Learning for Vector Quantization*, Neural Processing Letters **15**, 261-273, 2002.
14. Y. Ozturk and H. Abut, *System of associative relationships (SOAR)*, Proceedings of ASILOMAR, 1997.
15. N.R. Pal, J.C. Bezdek, and E.C. Tsao, *Generalized clustering networks and Kohonens self-organizing scheme*, IEEE Trans. Neural Networks, **4(4)**, 549-557, 1993.
16. S.A. Rizvi, L.C. Wang and N.M. Nasrabadi, *Nonlinear vector prediction using feedforward neural networks*, IEEE Trans. Image Process. **6(10)**, 1431-1436, 1997.
17. W. Skarbek and A. Cichocki, *Robust image association by recurrent neural subnetworks*, Neural Process. Lett. **3**, 131-138, 1996.
18. D. Tzovaras and M.G. Strintzis, *Use of nonlinear principal component analysis and vector quantization for image coding*, IEEE Trans. Image Process. **7(8)**, 1218-1223, 1998.

19. N. Ueda, and R. Nakano, *A new competitive learning approach based on an equidistortion principle for designing optimal vector quantizers*, Neural Networks, **7(8)**, 1211-1227, 1994.
20. L.C. Wang, S.A. Rizvi and N.M. Nasrabadi, *A modular neural network vector predictor for predictive image coding*, IEEE Trans. Image Process. **7(8)**, 1198-1217, 1998.
21. A. Weingessel, H. Bischof, K. Hornik et al., *Adaptive combination of PCA and VQ networks*, IEEE Trans. Neural Networks **8(5)**, 1208-1211, 1997.
22. E.K. Yair, K. Zeger and A. Gersho, *Competitive learning and soft competition for vector quantizer design*, IEEE Trans. Signal Processing, **40(2)**, 294-308, 1992.

Feature Extraction Using Class-Augmented Principal Component Analysis (CA-PCA)

Myoung Soo Park, Jin Hee Na, and Jin Young Choi

School of Electrical Engineering and Computer Science, ASRI,
Seoul National University, Seoul 151-744, Korea
meister1@gmail.com

Abstract. In this paper, we propose a novel feature extraction method called Class-Augmented PCA (CA-PCA) which uses class information. The class information is augmented to data and influences the extraction of features so that the features become more appropriate for classification than those from original PCA. Compared to other supervised feature extraction methods LDA and its variants, this scheme does not use the scatter matrix including inversion and therefore it is free from the problems of LDA originated from this matrix inversion. The performance of the proposed scheme is evaluated by experiments using two well-known face database and as a result we can show that the performance of the proposed CA-PCA is superior to those of other methods.

1 Introduction

Feature extraction is an important issue for classification of data with large input dimension such as face images. The purpose of feature extraction is to generate a set of features that have smaller dimension than original data and include the data characteristics sufficient to classify data. These extracted features can reduce the computation for classification and improve the classification performance by removing non-relevant characteristics in a data set.

The Principal Component Analysis (PCA) [1], also called as Karhunen-Loeve transform (KLT), is a well-known statistical method to extract the features for face recognition. This method is very effective to find the features for reducing the dimension, however, these features may be inappropriate for classification since the class information is not considered during the determination of features. Another well-known method is the Linear Discriminant Analysis (LDA) [2][3] which can resolve the difficulty of PCA by using the scatter matrix including the class information, however, it has its own drawbacks such as singularity problem originated from the use of scatter matrix including the matrix inversion [4]. To resolve the difficulties and improve the performance, some variants of LDA have been developed such as Fisherface [5], Direct-LDA [6], Generalized LDA [7], etc.

In this paper, a novel supervised feature extraction scheme is proposed to ease the difficulties mentioned in the above. In order to utilize the class information, the new dimension which encodes the class information is augmented to

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 606–615, 2006.

the original data. This augmented data should be carefully normalized in order to maximize the effect of class information in feature extraction. By applying the PCA to this augmented data, we can find a transformation matrix to generate the feature which can describe the data distribution well and improve the classification performance. This scheme does not include matrix inversion, and therefore it is free from the problems of LDA mentioned in the above. We will call this scheme as Class-Augmented Principal Component Analysis (CA-PCA) and describe in the subsequent sections. The improved classification performance by using the features from CA-PCA will be demonstrated by experiments and explained.

2 Key Concept of CA-PCA

The goal of PCA is to find a set of orthogonal axis such that the variance of the data along which is maximized. The values of data along these axis are called *principal components* and they can be determined by the following equation.

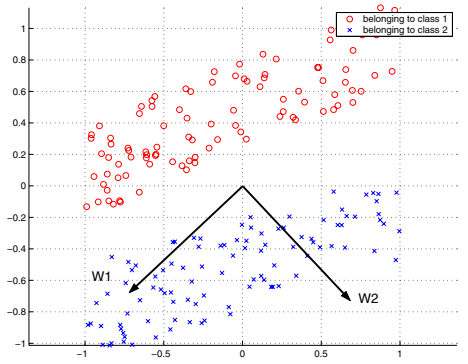
$$X_{feature} = W^T X \tag{1}$$

where X is the original representation of data and $X_{feature}$ is the vector consisting of principal components of X. $W = [w_1\ w_2\ ...\ ]$ is a transformation matrix consisting of a set of basis vector corresponding to the axis for principal components. A small number of principal components can describe the most of variance in the original data set and they can be used as feature for replacing the original data representation.

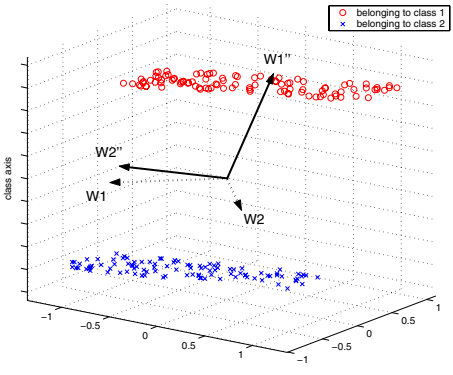
However, this feature from PCA may not be inappropriate for classification. If PCA is applied to the data set presented in Fig.1a, two principal components along axis w_1 and w_2 are obtained. For reducing the dimension of data representation we need to select only one axis w_1 among w_1 and w_2, and use the corresponding principal component as the feature of X. This feature is not suitable for classiciation since the data belonging to different classes cannot be separated by the value of this feature. If we can find the other axis w_1' in Fig.1c instead of w_1, the value of data along w_1' can be used as the feature approriate for classification since the data set can be easily separated by this value. To find out w_1' instead of w_1, the class information of the data set is necessary to be considered in the application of PCA.

In this paper, we will propose a scheme for enabling PCA to select the appropriate axis for classification. The following is an example to show overall procedure of this scheme applied to the data set in Fig.1a.

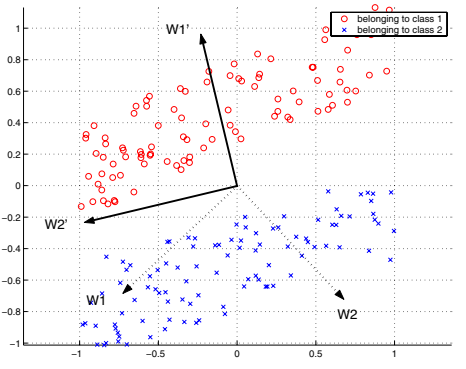
STEP 1. A new data representation is defined by augmenting a new axis which is orthogonal to all original axis and assigning a value along this new axis according to the class information of each data. According to this new representation, each data is plotted as a point in Fig.1b. Along this new class axis the data belonging to class 1 has different value from the value which the data belonging to class 2 has.



(a) the axis w_1, w_2 extracted by PCA.



(b) the axis w_1'', w_2'' extracted by PCA applied to the new data.



(c) the axis w_1', w_2' selected from w_1'', w_2''.

Fig. 1. An exemplary application of the proposed scheme. New class axis orthogonal to the data plane is augmented and the new data is defined from the original data value and its value on class axis which is determined according to the class of data. Then, the PCA is applied to new data and from the resultant principal components w_1'' and w_2'', the basis w_1' and w_2' for extracting feature can be obtained.

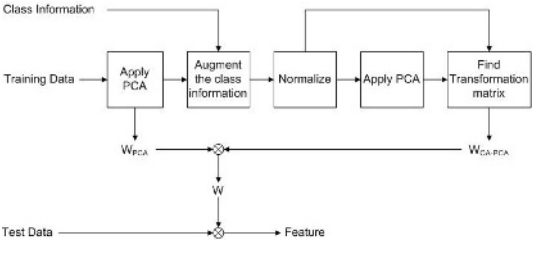


Fig. 2. Overall scheme of CA-PCA. CA-PCA consists of followsing steps: augmenting the encoded class information, normalizing the data, applying PCA, and determining the transformation matrix W

STEP 2. By applying PCA to these new data representation, we can find the axis w_1'' and w_2'' in Fig.1b. Since the variance along each axis is the sum of the variance along the orginal data axis and the variance along the class axis, the directions of new axis can be influenced by the variance along the class axis. In Fig.1b the value of each data along class axis is carefully adjusted such that w_1'' and w_2'' can be selected.

STEP 3. The projection onto the axis w_1'' and w_2'' requires the value of data on class axis, and therefore these axis cannot be immediately applied to the test data whose class information is unknown. If the value of each data along class axis is carefully adjusted such that the variance along class axis is very small, w_1'' and w_2'' can be approximated to w_1' and w_2' which are composed of the original data axis and they can be used for determining the feature as in in Fig.1c.

This simple example shows the overall steps of the proposed scheme which are depicted in Fig.2, however, it includes the issues whose details should be clearly specified: 1) encoding and normalizing of class information on class axis; 2) selecting of basis on the data plane from the principal components in the augmented space. They will be explained in the following subsections.

3 CA-PCA: Class-Augmented PCA

In this section, we will explain the detail of the proposed scheme. For augmenting the class information, the 1-of-n encoding scheme will be used and the normalization, for maximizing the effect of class information in feature extraction, will be applied. The PCA will extract principal components appropriate for classification, and among them the transformation matrix for feature extraction will be constructed.

3.1 Encoding of Class Information

There are various coding schemes which can be used for class labels, among which the thermometer coding and 1-of-n coding (n is the number of classes) are well known and widely used. With thermometer coding some class labels are closer to others than to the others and the difference between class labels can result in the side-effect which is not desirable for our scheme. For example, if the two data set belonging to two class are far while the labels for these two class are near from each other, the difference between data with class label become smaller than those without class label. This change can make it hard to find out the feature appropriate for classification. To avoid this kind of difficulty, we use the 1-of-n coding for other scheme. In this coding, each class label has the same distance from each other, therefore, the side-effect described in the above can be avoided.

Let the number of classes be n_{class}. Each class information for data X is represented by

$$C(X) = [c_1 \;\; c_2 \;\; \ldots \;\; c_{n_{class}}]^T \tag{2}$$

where class label c_i of X is a constant p if X belongs to class i, otherwise c_i is another constant n_i. Values of n_i are determined so that the mean of c_i becomes 0. p is determined so that the sum of the variances $\sum_{i=1}^{n_{class}} var(c_i)$ becomes σ^2 in which σ is selected as a scalar value much less than 1. The role of σ will be specified in the subsequent subsection. Since the number of equation is $n_{class}+1$ and the number of variables is $n_{class}+1$, all n_i and p can be determined.

Example: Assume that three data X_1, X_2, X_3 are given and the number of total classes n_{class} is 2. X_1 belongs to the first class, and the others belong to the second class. In this case the class information $C(X)$ for input data is expressed as follows:

$$\begin{array}{l} C(X_1) = [p \quad n_2]^T \\ C(X_2) = [n_1 \; p]^T \\ C(X_3) = [n_1 \; p]^T \end{array} . \tag{3}$$

For this representation, if $p = \sqrt{10}\sigma/5$, $n_1 = -\sqrt{10}\sigma/10$, and $n_2 = -2\sqrt{10}\sigma/5$, the average of c_1 and c_2 becomes 0 and the sum of the variances $\sum_{i=1}^{2} var(c_i)$ become σ^2.

3.2 Normalization

The purpose of normalization is to maximize the effect of class information in the selection of principal components. The principal components in Fig. 1a are rotated to those in Fig. 1c by the variances on the class axis. For this rotation can be possible between any of principal components, **the variance of data along the class axis needs to be set larger than the difference between the variances along axis on data plane**. For example in Fig.1b, if the variance along class axis is equal to or greater than the difference between variances along w_1 and w_2, w_1'' can be selected and w_1' can be determined from w_1''.

This condition can be described by the following equations. For all i and j,

$$\sigma^2 \geq |\sigma_{w_i}^2 - \sigma_{w_j}^2| \tag{4}$$

where $\sigma_{w_i}^2$ means the variance along the axis w_i.

This condition can be simultaneously satisfied for all pairs (i, j) by normalizing the variance along each data axis to be 1. After normalizing of data, the difference between variances along each axis becomes 0 and therefore the condition specified in the above can be satisfied for any value of σ which is larger than 0. There are another condition σ needs to satisfy, however, it will be explained in the later subsection. The normalization is carried out by the following equation for $j = 1, 2, ..., n_{input}$,

$$\overline{x}_j = x_j/\sigma_j \tag{5}$$

where n_{input} is the data dimension, $\overline{X}$ is the normalized data of X and x_j means the j^{th} element of X, and σ_j is the standard deviation of the j^{th} element over the entire data set. The variance of each element in X is changed to 1 by applying this scaling equation.

The encoded class information for each X is augmented to the normalized data as follows,

$$\overline{X}_i^a = \begin{bmatrix} \overline{X}_i \\ C(X_i) \end{bmatrix} \tag{6}$$

where $\overline{X}_i$ is the normalized data with dimension $n_{input} \times 1$ that corresponds to class information $C(X_i)$.

3.3 Application of PCA

After normalization and class augmentation, the normal PCA is applied to the set of $\overline{X}^a$ in order to obtain the principal components. The dimension of each principal component is $(n_{input} + n_{class})$, and the maximum number of principal components is also $(n_{input} + n_{class})$.

If reduction of the input dimension from n_{input} to $n_{feature} < n_{input}$ ie desired, the $n_{feature}$ principal components are selected along which the variance of the data is large. The obtained transformation matrix $\overline{W}^a$ is as follows.

$$\overline{W}^a = \begin{bmatrix} \overline{w}_1^a & \overline{w}_2^a & \dots & \overline{w}_{n_{feature}}^a \end{bmatrix} \tag{7}$$

where the dimension of principal component $\overline{w}_i^a$ is $(n_{input} + n_{class}) \times 1$. The dimension of $\overline{W}^a$ is $(n_{input} + n_{class}) \times n_{feature}$. The feature can be found by

$$X_{feature} = \overline{W}^{aT} \overline{X}^a \quad . \tag{8}$$

To avoid repeating the normalization for new data, the scaling factor of normalization can be included in W by the following equation for $i = 1, 2, ..., n_{feature}$ and $j = 1, 2, ..., n_{input}$,

$$w_{ij}^a = \overline{w}_{ij}^a / \sigma_j \tag{9}$$

where $\overline{w}_{ij}^a$ is the j^{th} element of $\overline{w}_i^a$ and w_{ij}^a is the j^{th} element of w_i^a. The other elements corresponding to class information, need not be modified for $j = n_{input} + 1, n_{input} + 2, ..., n_{input} + n_{class}$. It is because these elements are multiplied with class information and the class information of test data will not be given. These elements will be removed by the procedure explained in the next subsection.

With the calculated w_i, equation(8) can be rewritten as

$$X_{feature} = W^{aT} X^a. \tag{10}$$

where $W^a = [w_1^a \;\; w_2^a \;\; \dots \;\; w_{n_{feature}}^a] = [W_{input}^T \;\; W_{class}^T]^T$

3.4 Finding Transformation Matrix from Principal Components

If the obtained W^a is used to the augmented data consisting of datum X and class information $C(X)$, the augmented data can be transformed into features with a reduced dimension. This transformation is as follows.

$$X_{feature} = W^{aT} X^a \tag{11}$$

$$= [W_{input}^T \quad W_{class}^T] \begin{bmatrix} X \\ C(X) \end{bmatrix} \tag{12}$$

$$= W_{input}^T X + W_{class}^T C(X). \tag{13}$$

For the data given for finding a transformation matrix, called *training data*, the class information is known and C(X) can be determined. However, for the data given for classification called *test data*, class information is not given or permitted to be used and therefore C(X) cannot be determined. The above equation needs to be modified for being applicable to test data.

If σ^2 in the previous subsection is set to a much smaller value than 1, the elements from $W_{class}^T C(X)$ tend to become much smaller than those from $W_{input}^T X$ and the second term in (13) can be omitted. Experimentally we verified that the variance $var\{W_{class}^T C(X)\}$ from class information can be made less than 0.1% of the variance $var\{W_{input}^T X\}$ from data X for $\sigma = 0.01$, and the omittance of the second term does not significantly affect the effectiveness of features. The selected features from $\sigma = 0.1$ and these from $\sigma = 0.001$ were almost the same.

After omitting second term, the equation finally becomes as follows:

$$X_{feature} = W_{input}^T X + W_{class}^T C(X) \tag{14}$$

$$\simeq W_{input}^T X \tag{15}$$

$$= W^T X. \tag{16}$$

This equation can be applied for any test data X without class information, in order to obtain the corresponding value on the feature space.

4 Experiment

In this section, the CA-PCA is applied to face recognition problems and the classification performance of features extracted by CA-PCA is compared with those by the other methods such as PCA, LDA, ICA, and their variants.

4.1 Data Sets

We use two well-known face database for our experiments: YALE face database and ORL database. There exist two kinds of YALE face databases, a closely cropped set and a full face set, and among them we select the cropped set for our experiments. The YALE Face database contains 165 grayscale images of 15 subjects (individuals). There are 11 images per subject obtained under different facial expression or illumination conditions: center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised, and wink. ORL face database contains 400 grayscale images of 40 subjects. There are 10 images per subject with different poses.

There are experimental results using these databases, however, the direct comparison of our performance with these results is not easy since these results are obtained by using the differently downsampled images and by using different evaluating strategies. In this paper, we survey the recent results evaluated by the leave-one-out strategy (or called one-against-all strategy) and compare our result with them. Leave-one-out strategy is an appropriate cross-validation strategy for a data set in which the number of data is less than the dimension of data as in YALE and ORL face databases. For showing the result more clearly, we provide the results for images which are downsampled into two different scales.

4.2 Performance

The result using two databases are summarized in the following Table.1. In tables, the first column indicates the feature extraction methods which include CA-PCA, PCA variants (Eigenface, Kernel Eigenface, 2DPCA), LDA variants (Fisherface, Kernel Fisherface, RLDA, Generalizaed LDA, LDA/GSVD), DCT, and etc. The next columns indicate the image scale in which original image is downsampled, the number of features used for classification, and the type of classifier. Nearest Neighborhood Classifier is used for all cases except one cases using Radial Basis Function Network. The accuracies for the feature extraction method and the references in which the results are reported given in the last two columns.

In the first table, the result of experiments on the YALE face database is given. We can observe that the best classification accuracy 100.0% can be achieved by using the features extracted by CA-PCA from the image whose scale is 40x30. Compared to other methods which use images with the scale similar to or larger than 40x30, it is also noted that the number of features necessary for achieving this accuracy is the smallest in CA-PCA among those in the other methods presented in this table. For comparison with the performance of ICA-FX which is performed on the images with scale 30x21, the experiment for the images with the same scale is performed and given in the table. CA-PCA can achieve the best result 100.0% also for this case and the number of features for CA-PCA is a little smaller than that for ICA-FX.

In the second table, the experiment on the ORL face database shows the similar result with that from the experiments on YALE face database. 99.75%, the best classification accuracy among those presented in this table is achieved by using the features extracted by CA-PCA from the image whose scale is 40x30. Compared to other methods which uses images with similar or larger scale than 40x30, it is also noted that the number of features necessary for achieving this accuracy is much smaller in CA-PCA than those in the other methods presented in this table. For comparison with the performance of other methods which is performed on the images with scale 28x23, the experiment for the images with the same scale is also performed and given in the table. CA-PCA can achieve the best accuracy 99.75% also for this case. The number of features for CA-PCA is not the best since the feature for ICA-FX is a little smaller than that for CA-PCA.

Table 1. The reported classification performance on two face databases using different subspace methods and leave-one-out strategy. * means that the corresponding value or method is not specified in the report.

Feature Extraction Methods	Image Scale	Feature	Classifier	Accuracy	Reference
CA-PCA	30x21	9	NN	**100.0%**	
CA-PCA	40x30	8	NN	**100.0%**	
Eigenface	41x29	30	NN	71.52%	[8]
Kernel Eigenface (d=3)	41x29	60	NN	72.73%	[8]
2DPCA	100x80	*	NN	**84.24%**	[9]
Fisherface	41x29	14	NN	91.52%	[8]
Kernel Fisherface (G)	41x29	14	NN	93.94%	[8]
RLDA	106x81	*	NN	97.60%	[7]
Generalized LDA:To-R(S_w)	106x81	*	NN	89.70%	[7]
Generalized LDA:To-N(S_w)	106x81	*	NN	97.60%	[7]
Generalized LDA:To-NR(S_w)	106x81	*	NN	98.20%	[7]
LDA/GSVD	106x81	*	NN	**98.80%**	[7]
DCT	195x231	55	NN	80.00%	[10]
DCT (without 1^{st} 3 comp.)	195x231	55	NN	86.10%	[10]
DCT + FLD	195x231	15	NN	96.40%	[10]
DCT + FLD (without 1^{st} 3 comp.)	195x231	15	NN	97.00%	[10]
DCT + FLD (without 1^{st} 3 comp.)	195x231	15	RBF	**98.20%**	[10]
Isomap ($\varepsilon = 20$)	41x29	60	NN	72.73%	[8]
LLE (# of neighbor=70)	41x29	30	NN	73.94%	[8]
ICA	41x29	100	NN	71.52%	[8]
ICA-FX	30x21	14	NN	**96.36%**	[11]

(a) Result for YALE face database

Feature Extraction Methods	Image Scale	Feature	Classifier	Accuracy	Reference
CA-PCA	28x23	13	NN	**99.75%**	
CA-PCA	40x30	16	NN	**99.75%**	
Eigenface	28x23	40	NN	97.50%	[8]
Kernel Eigenface (d=3)	28x23	40	NN	98.00%	[8]
2DPCA	112x92	*	NN	**98.30%**	[9]
Fisherface	28x23	39	NN	98.50%	[8]
Kernel Fisherface (P, G)	28x23	39	NN	98.75%	[8]
RLDA	56x46	*	NN	98.00%	[7]
LDA/GSVD	56x46	*	NN	93.50%	[7]
Generalized LDA:To-R(S_w)	56x46	*	NN	98.00%	[7]
Generalized LDA:To-NR(S_w)	56x46	*	NN	98.80%	[7]
Generalized LDA:To-N(S_w)	56x46	*	NN	**99.00%**	[7]
Isomap ($\varepsilon = 10$)	28x23	30	NN	98.25%	[8]
LLE (# of neighbor=70)	28x23	70	NN	97.75%	[8]
ICA	28x23	80	NN	93.75%	[8]
ICA-FX	28x23	10	NN	**99.00%**	[11]

(b) Result for ORL face database

5 Conclusion

In this paper, a new supervised feature extraction scheme called CA-PCA (Class Augmented PCA) has been proposed. The issues for using class information with PCA are dealt with, and the scheme is proposed to consider these issues; the scheme is used for encoding class information and augmenting it to normalized data, and selecting the useful components from the principal components obtained by standard PCA.

The performance of the proposed scheme is evaluated by experiments using two well-known YALE and ORL face databases. For both databases, the CA-PCA can generate features which result in the 100.00% and 99.75% classification accuracies, respectively, and the number of necessary features for these results is also smaller in CA-PCA than those in other methods. This performance of CA-PCA is superior to the reported performance of other methods and therefore we can conclude that CA-PCA is an effective feature extraction method for classification.

References

1. M. Turk and A. Pentland, "Eigenfaces for recognition," Journal of Cognitive Neuroscience, vol.3, pp.71-86, 1991.
2. R. Duda and P. Hard, Pattern Classification and Scene Analysis, New York: Wiley, 1973.
3. R. A. Fisher, "The Use of Multiple Measures in Taxonomic Problems," Annual Eugenics, vol.7, pp.179-188, 1936.
4. R. Huang, Q. Liu, H. Lu, S. Ma, "Solving the Small Sample Size Problem of LDA", in Proceedings of IEEE International Conf. on Pattern Recognition, vol.3, 2002.
5. P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. FisherFaces: Recognition using class specific linear projection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.19, no.7, pp.711-720, 1997.
6. J. Yang, Y. Yu, and W. Kunz, "An Efficient LDA Algorithm for Face Recognition," in Proceedings of the 6th International Conference on Control, Automation, Robotics and Vision (ICARCV2000), 2000.
7. Cheong Hee Park, Haesun Park, Panos Pardalos, "A Comparative Study of Linear and Nonlinear Feature Extraction Methods," in Proceedings of 4th IEEE International Conference on Data Mining (ICDM'04), pp.495-498, 2004.
8. M. H. Yang, "Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition Using Kernel Methods," Proceedings of 5th IEEE Internationl Conference on Automatic Face and Gesture Recognition (RGR'02), pp.215-220, 2002.
9. Jian Yang, David Zhang, Alejandro F. Frangi, Jing-yu Yang, "Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.26, no.1, pp.131-137, 2005.
10. J. E. Meng, W. Chen and W. Shiqian, "Highspeed face recognition based on discrete cosine transform and RBF neural networks," IEEE Transactions on Neural Networks, vol.16, issue.3, pp.679-691, 2005.
11. Nojun Kwak, Chong-Ho Choi and Narendra Ahuja, "Face recognition using feature extraction based on independent component analysis," in Proceedings of International Conference on Image Processing 2002, Rochester, 2002.

A Comparative Study of the Objectionable Video Classification Approaches Using Single and Group Frame Features

Seungmin Lee, Hogyun Lee, and Taekyong Nam

Electronics and Telecommunications Research Institute (ETRI),
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-700, South Korea
{todtom, hglee, tynam}@etri.re.kr

Abstract. This paper deals with the methods for classifying whether a video is harmful or not and also evaluates their performance. The objectionable video classification can be performed using two methods. One can be practiced by judging whether each frame included in the video is harmful, and the other be obtained by using the features reflecting the entire characteristics of the video. The former is a single frame-based feature and the latter is a group frame-based feature. Experimental results show that the group frame-based feature outperforms the single frame-based feature and is robust to the objectionable video classification.

1 Introduction

The development of multimedia and Internet has led to the flood of contents, making it much easier for users to get access to tons of contents than ever before. This also made much easier for users including teenagers to be exposed to sexual contents; therefore, it is desperately needed to devise some measures to protect users, especially teenagers. The sexual contents are not only distributed properly but also produced by amateurs. Thus, we need to come up with a method, which can be practiced quickly, simply, and precisely at the moment of replay, rather than the prior censorship.

This paper provides algorithms which meet the requirements to classify whether videos are objectionable or not. For this, two methods are proposed and evaluated. The first method is to classify videos by using shape information of the skin color region[6]. In this method, we need to determine if the frames of the videos are harmful in order to judge whether the entire videos are harmful or not. That is, the method uses color information by extracting skin color region from the frame images, and then utilizes shape information by learning the shape of the skin color region with SVM(Support Vector Machine)[10,11]. By using this algorithm, it is possible to get the degree of harmfulness of individual images and to judge whether the entire video is harmful or not. The second method makes use of GoF(Group of Frame) information defined in MPEG-7[9]. By obtaining average SCD(Scalable Color Descriptor) histograms from several harmful and non-harmful videos, we get the GoF information and find criteria for classification using the SVM.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 616–623, 2006.

The paper is composed as follows. The 2nd section provides related works and explains why we chose the two methods. The 3rd section deals with the process of data collection and the characteristics of videos. The 4th section explains algorithms devised for the classification of harmful videos. The 5th section describes the experimental results. The 6th section is a conclusion.

2 Related Works

In the past, several researchers have worked on face and nude image detection for video content indexing, analysis and classification[1,2,3,4,5]. Most of them use aggregate features obtained from the binary skin region image which represent the percentage of skin within an image. Therefore, it is not sufficient for high accuracy harmful image detection in the objectionable video.

To increase the accuracy of the objectionable image detection, other information such as shape should be considered because shape patterns in the objectionable video are regular and repeated. In addition to the shape information, skin color model should be robust, accurate in lighting and race variations. Appearance-based nude image detection method[6] proposed the feature vector that contained shape and skin color information. Experimentally it shows that the method can achieve an excellent classification performance. And the previous works[7,8] discovered that under arbitrary conditions of illumination and race variation, the HSV was most discriminative color space of the RGB, HSV, YIO, YCbCr and CMY. Therefore, we used the shape and color information proposed by [6] and GoF descriptor computed from HSV color space defined in MPEG-7[9].

3 Collecting the Data

In order to get the reasonable experimental results, the testing data should be large enough and gathered from various genre. A total of 2,000 videos, whose replay length is an average of 53 minutes, were collected for the experiment. Half of them include harmful images, and the other half don't, consisting of five sub-categories such as documentary, movie, soap-opera, music, and sports. Out of all the collected videos, still images numbering in 630,733 were extracted at an interval of 10 seconds. Then, the still images were classified whether they are harmful by individual examiners. The classification work might have some errors according to the examiners; therefore, some detailed rules for classification were made in advance and learned by the examiners so as to minimize possible errors. Through this process, it turned out that 208,318 still images are harmful and the proportion of harmful images contained in the harmful videos is 66%. The file format was restricted to AVI and MPEG.

4 Methods for Classifying Objectionable Videos

As mentioned, objectionable video classification can be performed using two methods. One can be practiced by judging whether each frame included in the

video is harmful or not, and the other be obtained by using the features reflecting the entire characteristics of the video.

4.1 The First Method: Where and How Many Objectionable Frames Are Included?

To find out the proportion of the harmful frames contained and their emerging locations, it must be first determined whether each frame is harmful or not. Although there are several existing algorithms to determine whether images are harmful, the newly proposed classification method[6] employing shape information of the skin color region is used for this paper. First, we need to get skin color region using texture characteristics of the human skin, which then generates the skin likelihood image. The third picture in Figure 1 represents the high and low of the probability by means of shading. These images are standardized and used as input for Support Vector Machines (SVM). The standardizing is carried out by converting the size of the images into 40×40 and then into the vector of 1,600 in length. These vectors (1,600 in length) are defined as feature values. By extracting these values from the learning image set and analyzing them with SVM, the criteria for the classification are found.

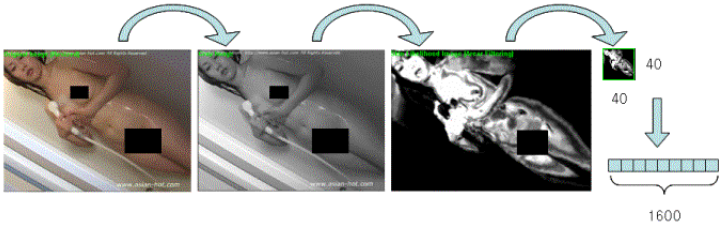


Fig. 1. The process of feature extraction for objectionable image classification using shape and skin color information

Figure 2 shows the SVM learning and X modeling procedure for generating the single frame based decision values. We first select training images manually from the extracted frames. Next, we extract feature values from the training images and conduct SVM learning process. After leaning process, we can get hyperplane value and decide each frame whether harmful or not. If the SVM learning machine returns positive value, the input frame is harmful. If we extract m frames from a test video I_n, we can get return values from SVM from $I_{n,1}$ to $I_{n,m}$.

We summarize these values with two functions, X_{avg} and X_{ratio}. Finally, we can decide whether the test video I_n is harmful or not based on the return value from one of two functions. The variable x_i stands for i^{th} input frame. The function f_{svm} represent the SVM classification function that extract feature vector from input frame and conduct SVM classification. The function B_i represents

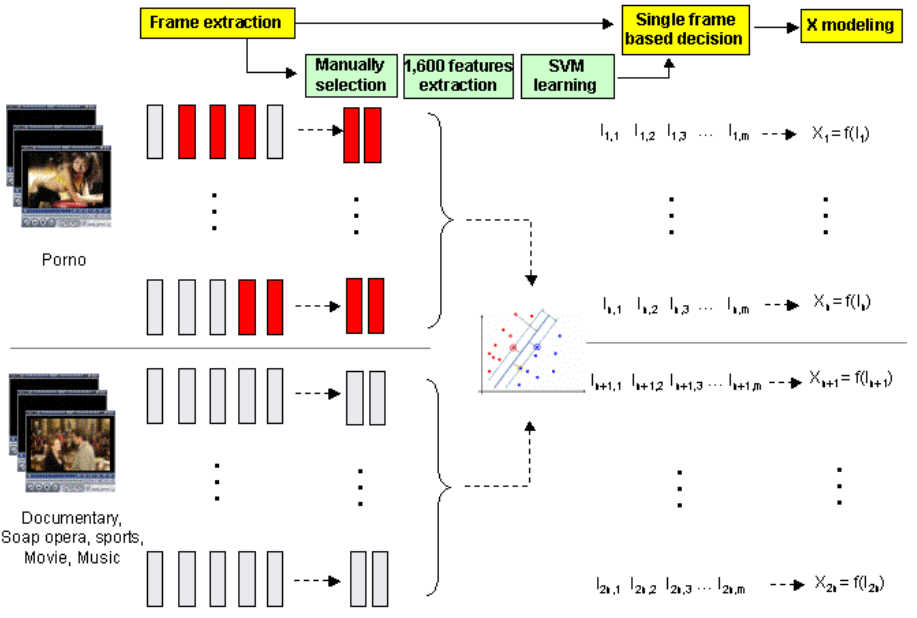


Fig. 2. X modeling procedure

the binary classification function that returns 1 or 0. The m is the number of the frames extracted.

$$X_{avg} = \frac{\sum_{i=1}^{m} I_i}{m} \quad \text{where} \quad I_i = f_{svm}(x_i) \tag{1}$$

$$X_{ratio} = \frac{\sum_{i=1}^{m} B_i}{m} \quad \text{where} \quad B_i = \begin{cases} 1 \text{ if } I_i \geq 0 \\ 0 \text{ if } I_i < 0 \end{cases} \tag{2}$$

4.2 The Second Method: Is the Entire Characteristics of the Video Objectionable Or Not?

If it is needed to know whether or not a video are objectionable, it might be an efficient way to use only some features reflecting the entire characteristics of the video without examining every frame. Figure 3 shows the SVM learning and Y modeling procedure for generating the group frame based decision values.

Objectionable videos generally contain more skin color information than the others. We use HSV color space for skin color detection because color in HSV space is robust to illumination, lights and noise and is most discriminative in face detection. HSV color space consists of hue(H), saturation(S), value(V). Hue shows the attribute of a visual sensation. Saturation measures the lack of white in the color and value is a linear combination of RGB components. The HSV color space is the color space associated with the group of frames histogram descriptor. For this descriptor, the HSV space is uniformly quantized into 256 bins - 16 levels

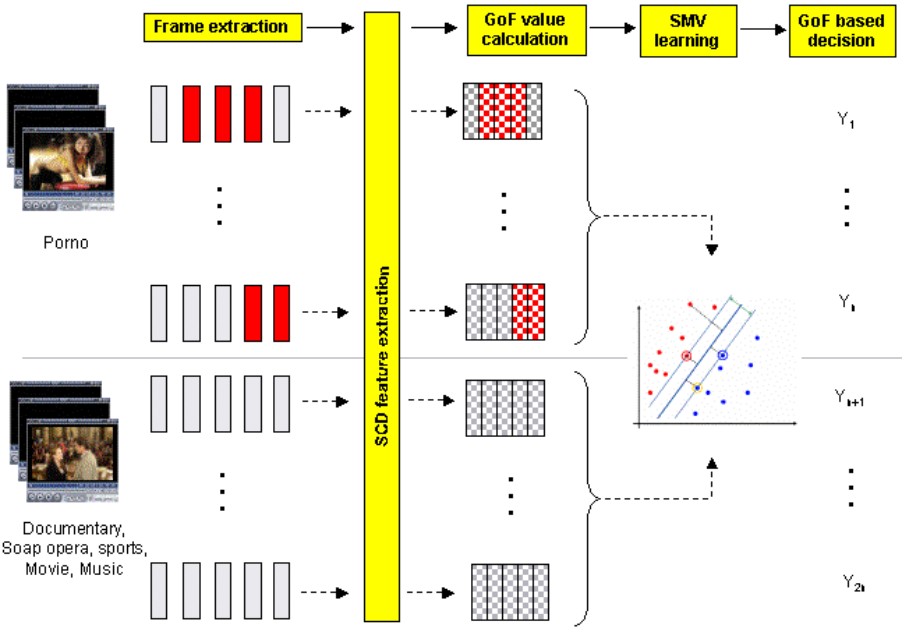


Fig. 3. Y modeling procedure

in H, 4 levels in S and 4 levels in V. The group of frame (GoF) color descriptor is used for the joint representation of color-based features for multiple frames in a video segment. The GoF color descriptor is obtained by aggregating the 256 bins of multiple video frames and by averaging the aggregated bin values. The average histogram is computed by accumulating the frame histograms in the group and subsequently normalizing each accumulated bin value by m, where m is the number of frames in the GoF. We obtain GoF values of 256 features from the training set and perform SVM learning based on Radial Basis Function (RBF) kernel using the values, as illustrated in Figure 3. We define Y as a classification result that is a distance between GoF and support vectors.

5 Experimental Results

The process of the experiment runs as follows. The number of videos used for experiment was 1,186 out of total 2,000 videos as shown in table 1. The single frame-based learning models denoted as the variable X were generated using 1,000 frames from 598 videos. The GoF-based learning models denoted as the variable Y were learned by extracting frames from 598 videos at an interval of 60 seconds. The Table 2 stands for the result of classification for 47,465 images using shape and skin color information. These images are produced from test video set. This method shows lower performance (74.4%) than it is expected because we omit pre-processing modules(face detection and skin filter) to improve system

Table 1. Video data set for learning and test

Category	Sub-category	Learning set	Test set
Non-objectionable	Documentary	60	60
	Soap opera	60	60
	Sports	60	60
	Movie	60	60
	Music	60	51
Objectionable	Porno	298	297

Table 2. Image classification performance using shape and skin color information

Precision	Recall	Accuracy
0.722	0.721	0.744

Table 3. Video classification performance using X, Y models with cross point threshold

Method	Precision	Recall	Accuracy
X_{avg}	0.842	0.893	0.861
X_{ratio}	0.892	0.883	0.886
Y	0.950	0.889	0.920

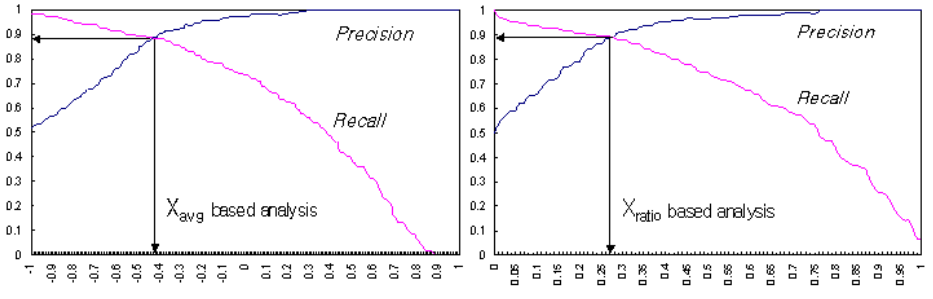


Fig. 4. Finding thesholds

speed and test set contains low quality videos that are difficult to extract shape and skin color information.

To decide whether input video is harmful or not, we need a threshold for each function. To find the optimal threshold, we repeat the experiment changing the threshold. Figure 4 shows the optimal threshold of two X functions for learning data set where the x axis shows threshold and the y axis shows performance. The cross point of the lines(precision, recall) can be a optimal threshold. The Table 3 stands for the result of classification for test data set by X and Y modeling with the optimal threshold. The optimal threshold is obtained from the experiment with learning data. Y shows best performance and X_{ratio} shows better performance in precision and accuracy than X_{avg}.

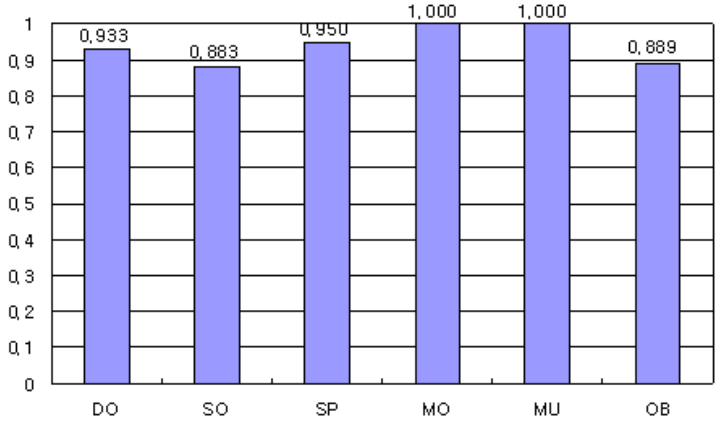


Fig. 5. Performance of sub categories using Y model

Figure 5 shows hit ratio using Y model for six sub categories where DO is documentary, SO is soap opera, SP is sports, MO is movie, MU is music, OB is objectionable, x axis shows sub categories and y axis shows performance. Y model shows even performance for all sub categories.

6 Conclusion

This paper compares two methods to judge whether videos are harmful or not: One is using shape and skin color information for the classification and the other is based on the GoF feature. As the classification method for the objectionable image considers shape information as well as skin color, it was expected to show higher performance in the classification of a single image. When applying this to the binary classification, a single frame-based method shows the lower performance than the GoF-based method. Although the single frame-based method shows high performance in judging a single image, it does not reflect the entire characteristics of the videos. That is why it has lower performance than GoF reflecting the traits of the group frames. And the performance of GoF-based method is robust to the objectionable video classification. Based on the fact that it is important to reflect the whole features of videos, we would like to achieve better classification performance by considering motion and audio information and all the properties of videos.

References

1. M. Fleck, D. Forsyth, and C. Bregler, "Finding Naked People," In European Conf. on Computer Vision, vol.2, p.592-602, 1996.
2. M. J. Jones and J. M. Regh, "Statistical Color Model with Application to Skin Detection," In Technical Report CRL, 1998.

3. J. Z. Wang, G. Wiederhold and O. Firschein, "System for Screening Objectionable Imagers," Computer Communications, vol.21, p.1355-1600, 1998.
4. A.Bosson, G. C. Cawley, Y. Chan and R. Harvey, "Non-Retrieval: Blocking Pornographic Images," In International Conf. on Image and Video Retrieval, 2002.
5. Jae-Ho Lee, "Automatic Video Management System Using Face Recognition and MPEG-7 Visual Descriptors," ETRI Journal, vol. 27, p.806-809, 2005.
6. Chi-Yoon Jeong, Jong-Sung Kim, and Ki-Sang Hong, "Appearance-Based Nude Image Detection," ICPR, p.467-470, 2004.
7. Dzmitry Tsishkou, Mohamed Hammami, and Liming Chen, "Face Detection in Video Using Combined Data-mining and Histogram based Skin-color Model," Proceedings of 3rd International Symposium on Image and Signal Processing and Analysis, p. 500-503, 2003.
8. O. Ikeda, "Segmentation of Face in Video Footage Using HSV Color for Face Detection and Image Retrieval," Image Processing, 2003. ICIP 2003, Proceedings, vol. 3, p.14-17, 2003.
9. B.S. Manjunath, Philippe Salembier, and Thomas Sikora, Introduction to MPEG-7, John Wiley & Sons, LTD, 2002.
10. V. Vapnik, "The Nature of Statistical Learning Theory," Springer-Verlag, New York, 1995.
11. Thorsten Joachims, "SVMlight Support Vector Machine," http://svmlight.joachims.org/

Human Facial Expression Recognition Using Hybrid Network of PCA and RBFN

Daw-Tung Lin

Department of Computer Science and Information Engineering
National Taipei University
Sanshia, Taipei County, Taiwan

Abstract. In this paper, we propose a hybrid architecture combining radial basis function network (RBFN) and Principal Component Analysis (PCA) re-constructure model to perform facial expression recognition from static images. The resultant framework is a two stages coarse to fine discrimination model based on local features extracted from eyes and face images by applying PCA technique . It decomposes the acquired data into a small set of characteristic features. The objective of this research is to develop a more efficient approach to classify between seven prototypic facial expressions, such as neutral, joy, anger, surprise, fear, disgust, and sadness. A constructive procedure is detailed and the system performance is evaluated on a public database "Japanese Females Facial Expression (JAFFE)". As anticipated, the experimental results demonstrate the potential capabilities of the proposed approach.

1 Introduction

Facial data analysis is one of the essential medium of perceptual processing and emotion modeling [1,2]. Facial expression recognition methods can be generally divided into two categories: static images vs. video sequences, based on differ use of data and feature extraction methods. Typical techniques include optical flow estimation, spatial feature analysis, and local filter analysis. Yacoob and Davis [3] utilized optical flow method to track the dynamic movement of facial features from video sequences and classified the representation of facial feature movement into six expressions (i.e., joy, surprise, anger, fear, sadness, and disgust). Barlett et al. [4] combined optical flow and principal component analysis (PCA) for facial expression recognition. The Facial Action Coding System (FACS) derived by Ekman and Friesen has been widely used to describe the facial expression by movement of action units (AUs) [5]. The FACS is often incorporated with the above mentioned techniques to delineate the details of human expression for video sequences. Donato et al. [6] provided a more detail review of the recent techniques for facial expression recognition based on video sequences and FACS encoding. Anderson and McOwan presented a fully automated and multistage system for real-time recognition of facial expression utilizing SVM to distinguish the motion signatures [7]. Devillers et al. addressed how the emotion express is perceived from spoken dialogs based on machine learning approach [8].

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 624–633, 2006.

Fig. 1. Examples of seven principal facial expressions in JAFFE [20]: joy, disgust, anger, surprise, fear, neutral, and sadness (from left to right)

Facial expression recognition from still images is a more difficult problem than from video sequences due to the fact that less information during expression actions is available [9]. Cottrell and Metcalfe [10] applied PCA and backpropagation neural network to recognize facial expression, gender, and identity from static images. Chen and Huang [9] modified linear discriminate analysis (LDA) algorithm and presented a new clustering based feature extraction method for facial expression recognition. A constructive feed-forward neural network was further proposed for facial expression recognition with pruning technique by Ma and Khorasani [11]. Other approaches such as multiple discriminate analysis, ICA, Adaboost, Fisher Weight Maps, Appearance model etc. can also been found in the recent literatures [12,13,14,15,16,17,18,19].

In this paper, we are concerned with automatic classification of facial expressions from still images. The psychologists have indicated that as least six emotions are universally associated with distinct facial expressions. Examples of Japanese Females Facial Expression (JAFFE) databases are shown in Fig. 1. The rest of this paper is arranged as follows. Section 2 illustraes the main components of the proposed system. We utilized PCA in the pre-processing stage to extract features from face imagery. We further proposed a hybrid model of radial basis function network and PCA re-construction network to fulfill the facial expression differentiation task. In section 3, the experimental results are presented. Finally, conclusion remarks are drawn in section 4.

2 Hybrid Network Model

2.1 Radial Basis Function Network

The radial basis function neural network (RBFN) theoretically provides a sufficient large network structure such that any continuous function can be approximated to within an arbitrary degree of accuracy by appropriately choosing radial basis function centers [21]. The RBFN is trained using sample data to approximate a function in multidimensional space. The RBFN is a three-layered

network. The first layer constitutes input layer in which the number of nodes is equal to the dimension of input vector. In the hidden layer, the input vector is transformed by radial basis function as activation function: $\varphi(\boldsymbol{x}; \boldsymbol{c_j}) = exp(-\frac{1}{2\sigma^2}\|\boldsymbol{x} - \boldsymbol{c_j}\|^2)$, where $\|\cdot\|$ denotes a norm (usually Euclidean distance) of the input data sample vector $\boldsymbol{x}$ and the center $\boldsymbol{c_j}$ of radial basis function. The kth output is computed by equation

$$F_k(x) = \Sigma_{j=1}^{m} w_{kj} \cdot \varphi(\boldsymbol{x}; \boldsymbol{c_j}), \tag{1}$$

where w_{kj} represents a weight synapse associates with the jth hidden unit and the kth output unit with m hidden units. Given a set of N different points $\{\boldsymbol{x_i} \in R^p | i = 1, 2, \cdots, N\}$ as input pattern and the corresponding set of desired target values $\{\boldsymbol{d_i} \in R^k | i = 1, 2, \cdots, N\}$, the goal is to find a function $F : R^p \rightarrow R^k$ that satisfies the condition: $F(\boldsymbol{x_i}) = \boldsymbol{d_i}, \quad i = 1, 2, \cdots, N$. Thus, we obtain the following result derived from Equation (1):

$$\begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1m} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2m} \\ \vdots & & & \\ \varphi_{N1} & \varphi_{N2} & \cdots & \varphi_{Nm} \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1k} \\ w_{21} & w_{22} & \cdots & w_{2k} \\ \vdots & & & \\ w_{m1} & w_{m2} & \cdots & w_{mk} \end{bmatrix} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1k} \\ d_{21} & d_{22} & \cdots & d_{2k} \\ \vdots & & & \\ d_{N1} & d_{N2} & \cdots & d_{Nk} \end{bmatrix} \tag{2}$$

where $\varphi_{ij} = \varphi(\boldsymbol{x_i}; \boldsymbol{c_j})$, and $i = 1, 2, ..., N$, $j = 1, 2, ..., m$. We can then rewrite Equation (2) to the form $\varphi \cdot W = D$. Thus, the weight matrix can be obtained by the least square approximation algorithm $W = (\varphi^T \varphi)^{-1} \varphi^T D$.

We employed the RBFN to classify the facial expressions images in the Eigenspace domain extracted via PCA as described in the next section. The major advantages of RBFN are its fast training speed and local feature convergence [21].

2.2 Classification with PCA Reconstruction

Principal Component Analysis (PCA) has been commonly used to faces recognition problems. Typical PCA algorithm (Eigenface/Fisherface) is one of the main streams of research on face feature processing [22]. PCA has advantage over other face recognition schemes in its speed and simplicity. We utilize PCA in the pre-processing stage to extract features from face imagery. The basis of the ordinary image space is composed of all single pixel vectors. However, the image space is not a optimal space for face representation and categorization. The aim of applying PCA is to build a face space which better describes the face images. The basis vectors of this face space are called the principal components. These components will be uncorrelated and will maximize the variance accounted in the original basis. It can also reduce the dimension of the feature space. The details of PCA derivation can be found in literatures [22].

To illustrate the feasibility of using eigen feature to fulfill expression classification task, we modify the PCA reconstruction method for preliminary evaluation. Notice that if the input image is much similar to some expression training set, the reconstructed image will has less distortion than the image reconstructed

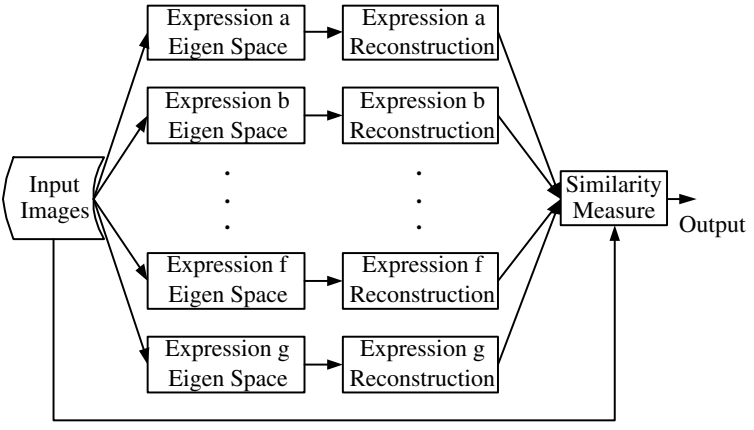


Fig. 2. Classification procedure of PCA reconstruction

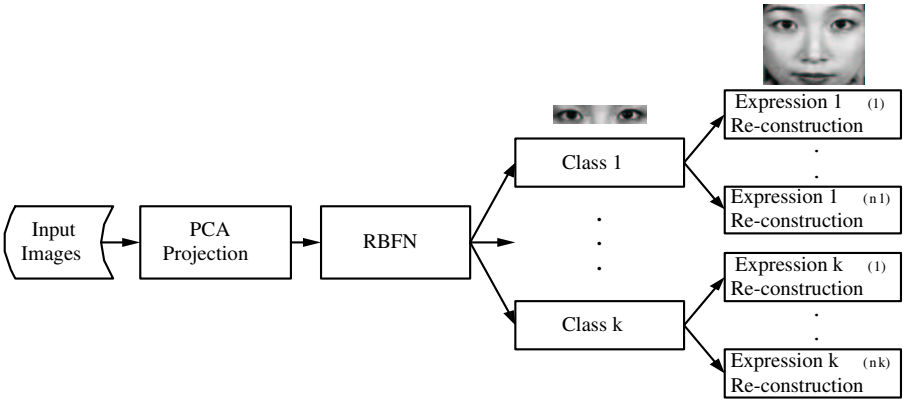


Fig. 3. Schematic block diagram of the proposed hybrid model

from other eigen vectors of training expressions. Based on this episode, we divide the training set into seven classes according to different expression and compute the eigen space of each class. For a test face image, we first project it onto the eigen space of each class independently and then derive reconstructed image from each eigen space. By measuring the similarity (mean-square error) between input image and the reconstructed image of each class, we can identify the class of input image whose reconstructed image is most similar to the input one. The procedure of the developed PCA reconstruction method is delineate in Fig. 2.

2.3 Hybrid Architecture

The cognitive and emotional states of a person can be correlated with visual features derived from images of the mouth and eye regions [23]. Many researches

on facial expression representation has focused on the specific feature motion of upper face (ie., eyes and brows) and lower face (lips) [24]. To improve the performance of recognition, we further proposed a hybrid model combining RBFN and PCA re-construction network. The schematic diagram of the proposed hybrid architecture is presented in Fig. 3. We divide the classification process into two stages. In the first stage, we intend to categorize the expressions into k ($2 \leq k \leq 6$) coarse classifiers according to Eigen-features of eyes. Each classifier is aimed to differentiate candidates of a subset of expressions. In the second stage, each classifier is further discriminated by using PCA re-construction method according to their corresponding face features. The number of expressions to be recognition is denoted as n_k for classifier k. The notation is shown in Fig. 3.

3 Experimental Results

In this section, we demonstrate the performance of the proposed hybrid approach in classifying seven facial expressions. The JAFFE database used in the experiments consists of of 213 frontal pose images of ten Japanese females. Each person posed some examples of each of the seven fundamental facial expressions, such as neutral, joy, sadness, surprise, fear, anger, and disgust. We noticed that there are two original pictures may have been labeled incorrectly. To avoid confusion, we removed them from the database. Then, he database are partitioned into ten folds without overlapping. Images of different expressions are randomly selected from ten persons and divided into ten sets with roughly equal same sizes. Nine sets are used for training and the remaining images are used for testing. This process is repeated until all of the images are tested. The recognition rate are measured based on the overall database. To investigate the local effect of the source images, two types of images are acquired from the database for the experiments (examples are illustrated in Fig. 4): Type A – face images without hair and shoulders, image size is 80×80; Type B – images of eyes and mouth region with size of 80×20 and 45×30, respectively. Experiments were first performed according to the procedure introduced in the previous sections with PCA reconstruction and single stage RBFN approaches. Table 1 and Table 2 list the corresponding simulation results. As we can observe from Table 1 and Table 2, the classification rate of using PCA reconstruction method based on face images is up to 93.81% (top three matches). The correct classification rates based on local lips and eyes features with single stage RBFN are 91.90% and 93.33%, respectively, by counting the top three matched candidates. Thus, it is possible to re-organize the training mechanism and construct a hybrid network for further investigation and improvement.

Notice that the recognition performance of RBFN for eyes images could be as high as 92.86% when we count the rank of top three matches. This phenomenon indicates that it is possible to improve the performance by grouping several classes of expressions together and constitute a pre-classification filter. To obtain the design guideline of the multiple layers delineation structure, we want to observe whether local eyes image eigen-feature can help to do the

(a) Type A – face images.

(b) Type B – eyes and lips images.

Fig. 4. Sample images extracted from JAFFE database: (a) Type A – face images, and (b) Type B – eyes and lips images

Table 1. Classification rate (%) of PCA reconstruction method

Data type	top one match	top two matches	top three matches
Type A - face	84.83	90.95	93.81
Type B - lips	78.57	88.10	91.90
Type B - eyes	83.33	90.95	93.33

Table 2. Classification rate (%) of single stage RBFN with PCA pre-processing method

Data type	top one match	top two matches	top three matches
Type A - face	73.81	87.14	93.33
Type B - lips	64.29	84.29	90.95
Type B - eyes	76.19	89.52	92.86

preliminary clustering task. The confusion matrices of the single stage RBFN top one match test results are recorded in Table 3. As we can see from the confusion matrices, not all expressions were equally well recognized by the system. Expressions "fear" and "neutral" were often mis-classified to other classes. Besides, expressions "sadness", "disgust", "anger" were often mis-classified with each other. "Joy" is seldom mis-classified by other expressions. Thus, we intend to divide the first stage classification into various categories by grouping several

Table 3. Confusion matrix of facial expression classification measured by single stage RBFN method on type B eyes test images from JAFFE database

I \ O	Sadness	Joy	Disgust	Neutral	Surprise	Fear	Anger
Sadness	22	0	3	0	0	1	4
Joy	1	22	0	3	4	1	0
Disgust	4	1	18	1	1	0	4
Neutral	0	1	0	18	1	2	8
Surprise	1	0	0	1	19	3	5
Fear	6	1	1	0	3	17	3
Anger	0	0	5	3	0	1	21

Table 4. Comparison of the recognition rate (%) of different arrangements

scenario	stage	Sadness	Joy	Disgust	Neutral	Surprise	Fear	Anger	average
A	eyes	100	100	96.55	93.33	89.66	90.32	93.33	94.74
	face	86.67	84.21	82.76	83.33	89.66	83.87	83.33	84.78
B	eyes	100	80.65	93.10	100	100	96.77	100	95.79
	face	86.67	82.76	82.76	83.33	90.00	87.10	90.00	87.18
C	eyes	100	90.32	100	86.67	82.76	93.55	100	93.33
	face	90.00	77.42	86.21	86.67	82.70	90.32	90.00	85.72

Table 5. Comparison of the recognition rate of different approaches tested on JAFFE data set

method	recognition rate
HLAC+Fisher Weight Maps [14]	69%
LNNF [13]	70% $\sim$ 80%
Gabor wavelet [20]	82%
Boosted ICA [25]	86%
LBP+Linear Programming [16]	93.8%
our method	95.79% (eyes), 87.18% (face)

mis-classified categories. Later, train the face image individually for each classifier to perform less intensive identification task in the second stage. Various scenarios can be verified. Three of the well trained cases are: scenario A separates "surprise" from the reset of expressions; scenario B uses "joy" as one set and the rest becomes one set; scenario C contains two subsets ("surprise, neutral") and the rest of expressions. The recognition results of the proposed hybrid model are illustrated in Table 4. The best recognition of the first stage, eyes phase, is 95.79%. The performance of the second stage recognition is 87.18%. For the purpose of comparison, Table 5 shows the performance reported in previous literatures tested on the same JAFFE database. Recently, Wang *et al.* reported 92.4% recognition rate of facial expression, but they tested on different dataset, although trained on JAFFE database [12]. Besides, Feng *et al.* proposed a Linear

Programming method and achieved 93.8% recognition rate based on 21 two-class classifiers [16]. The evaluation scheme is different than ours. As we can observe from Table 4, our method is very competitive and outperforms the previous methods.

4 Conclusion

In this paper, we proposed a hybrid model combining PCA feature re-construction and RBFN method to tackle facial expressions recognition problems. From our experimental results, this cascade strategy works properly. As we can see from the records, our method is very competitive. We learn that it is possible to re-organize the training scenarios and construct a multi-layer hybrid network for further investigation and improvement. Thus, we arrange the first stage classification by dividing into several groups of categories using local features of eyes images. Later, train the face image individually for each classifier to perform less intensive identification task in the second stage. Various arrangements have been verified. From these results, we conclude the local face features is useful for differentiating expressions when a more appropriate classifier is arranged. In the future, we can adopt a more sophisticate learning algorithm of neural network and fuzzy theory to induce the best combination of sub-category in the first stage. Leverage of the recognition performance can be expected. Furthermore, more work will be conducted to validate the technique on other databases such as CMU AMP Face Expression Database [26], POFA image set of Ekman and Friesen [27], and other emotion research databases [28].

Acknowledgement

This work was supported in part by National Science Council. Taiwan, R.O.C. under grants NSC 88-2213-E216-010 and NSC 89-2213-E216-016. The author would like to thank Mr. De-Cheng Pan for his help in simulations and experiments.

References

1. S. Ioannou, A. Raouzaiou, V. Tzouvaras, T. Mailis, K. Karpouzis, and S. Kollias. Emotion recognition through facial expression analysis based on a neurofuzzy network. *Neural Networks*, 18:423–435, 2005.
2. N. Fragopanagos and J. Taylor. Emotion recognition in human-computer interaction. *Neural Networks*, 18:389–405, 2005.
3. Y. Yacoob, H.M. Lam, and L.S. Davis. Recognizing faces showing expressions. In *International Workshop on Automatic Face and Gesture Recognition*, Zurich, 1995.
4. M. Bartlett, P. Viola, T. Sejnowski, L. Larsen, J Hager, and P. Ekman. Classifying facial action. In M. Mozer D.S. Touretzky and M. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8. The MIP Press, 1996.
5. P. Ekman and W. Friesen. The facial action coding system. *Consulting Psychologists Press*, 1965.

6. G. Donato, M. Bartlett, J. Hager, P. Ekman, and T. Sejnowski. Classifying facial actions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(10):974–989, October 1999.
7. K. Anderson and P. McOwan. A real-time automated system for the recognition of human facial expressions. *IEEE Transactions on Systems, Mand, and CyberneticsXPart B*, 36(1):96–105, 2006.
8. L. Devillers, L. Vidrascu, and L. Lamel. Challenges in real-life emotion annotation and machine learning based detection. *Neural Networks*, 18:407–422, 2005.
9. X.W. Chen and T. Huang. Facial expression recognition: A clustering-based approach. *Pattern Recognition Letters*, 24:1295–1302, 2003.
10. G. Cottrell and J. Metcalfe. EMPATH: Face, gender, and emotion recognition using holons. In J. Moddy R. P. Lippman and D.S. Touretzky, editors, *Advances in Neural Information Processing Systems*, volume 3, pages 564–571, 1991.
11. L. Ma and K. Khorasani. Facial expression recognition using constructive feedforward neural network. *IEEE Trans. on Systems, Man, and Cybernetics – Part B: Cybernetics*, 34(3):1588–1559, June 2004.
12. Y. Wang, H. Ai, B. Wu, and C. Huang. Real time facial expression recognition with adaboost. In *Proceedings of the 17th International Confeence on Pattern Recognition*, volume 3, pages 926–929, Aug. 2004.
13. I. Buciu and I. Pitas. Application of non-negative and local non negative matrix factorization to facial expression recognition. In *Proceedings of the 17th International Confeence on Pattern Recognition*, volume 1, pages 288– 291, Aug. 2004.
14. Y. Shinohara and N. Otsu. Facial expression recognition using fisher weight maps. In *Proceedings of the Sixth IEEE International Confeence on Automatic Face And Gesture Recognition*, pages 499 – 504, 2004.
15. B. Abboud, F. Davoine, and M. Dang. Facial expression recognition and synthesis based on an appearance model. *Signal Processing: Image Communication*, 19(8):723–740, 2004.
16. X. Feng, M. Pietikainen, and A. Hadid. Facial expression recognition with local binary patterns and linear programming. *Pattern Recognition and Image Analysis*, 15(2):546–548, 2005.
17. M. Nakayama and T. Kumakura. Face identification performance using facial expressions as perturbation. In *Proceedings of the International Confeence on Artificial Neural Networks*, volume 1, pages 557–562, 2005.
18. M. Amin, N. Afzulpurkar, M. Dailey, and V. Esichaikul. Fuzzy-C-mean determines the principle component pairs to estimate the degree of emotion from facial. In *Proceedings of the 2nd International Confeence on Fuzzy Systems and Knowledge Discovery*, volume 1, page 484, 2005.
19. D. Datcu and L. Rothkrantz. Facial expression recognition with relevance vector machines. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, July 2005.
20. M. Lyons, J. Budynek, and S. Akamatsu. Automatic classification of single facial images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(12):1357–1362, December 1999.
21. S. Haykin. *Neural Networks: A Comprehensive Foundation*. Macmillan College Publishing Company, New York, 1994.
22. M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

23. Z. Duric, W. Gray, R. Heishman, F. Li, A. Rosenfeld, M. Schoelles, C. Schunn, and H. Wechsler. Integrating perceptual and cognitive modeling for adaptive and intelligent human - computer interaction. *Proceedings of the IEEE*, 90(7):1272–1289, 2002.
24. Y.-L. Tian, T. Kanade, and J. Cohn. Recognitizing action units for facial expression analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(2):97–115, February 2001.
25. L. He, J. Zhou, D. Hu, C. Zou, and L. Zhao. Boosted independent features for face expression recognition. In *Proceedings of the Second International Symposium on Neural Networks*, pages 137–145, May 2005.
26. Advanced Multimedia Processing Lab. Carnegie Mellon University. http://amp.ece.cmu.edu/projects/faceauthentication/download.htm.
27. P. Ekman and W.V. Friesen. Pictures of facial affect. Human Interaction Laboratory, Univ. of California Medical Center.
28. R. Cowie, E. Douglas-Cowie, and C. Cox. Beyong emotion archetypes: Databases for emotion modelling using neural networks. *Neural Networks*, 18:371–388, 2005.

Extracting Motion Primitives from Natural Handwriting Data

Ben H. Williams, Marc Toussaint, and Amos J. Storkey

Institute of Adaptive and Neural Computation
University of Edinburgh
School of Informatics
5 Forrest Hill, EH1 2QL Edinburgh, UK
ben.williams@ed.ac.uk

Abstract. For the past 10 years it has become clear that biological movement is made up of sub-routine type blocks, or *motor primitives*, with a central controller timing the activation of these blocks, creating synergies of muscle activation. This paper shows that it is possible to use a factorial hidden Markov model to infer primitives in handwriting data. These primitives are not predefined in terms of location of occurrence within the handwriting, and they are not limited or defined by a particular character set. Also, the variation in the data can to a large extent be explained by timing variation in the triggering of the primitives. Once an appropriate set of primitives has been inferred, the characters can be represented as a set of timings of primitive activations, along with variances, giving a very compact representation of the character. Separating the motor system into a motor primitive part, and a timing control gives us a possible insight into how we might create scribbles on paper.

1 Introduction

As with all planning tasks, there is a debate as to the degree of pre-planning in movement control. Humans and animals find solutions to movement tasks that are both repeatable to some extent across trials and circumstances, and subjects [17, 16, 12]. Despite this repeatability, we are also very adaptable to new tasks, and can quickly adjust learnt movements to cope with new environments [5]. There must therefore be some compromise between preprogrammed movement, and instantaneous movement planning in biological organisms.

Evidence suggests that once a particular movement has commenced, it cannot be unexpectedly switched off. Rather, to quickly modify a movement trajectory, the movements are superimposed [11]. This suggests that there is a subroutine type of movement activation, where the subroutines are not quickly adaptable, but their individual activation can be a fast and globally relevant process. This modularisation of movement control could provide a good compromise between movement pre-planning and online error correction. These subroutines of motion will be referred to as motor primitives. Strong biological evidence exists to suggest that these primitives exist, with motor primitives first being conclusively found in frogs [2, 3, 4] where stimulation of a single spinal motor afferent triggered a complete sweeping movement of the frog's leg. For a review of modularisation of motor control in the spine, see [1].

There have been many studies on recording the dynamics of all aspects of natural human movement. People have tried to infer motor primitive type sub-blocks from the sequences of movement [18, 12, 8, 15, 10]. Most of these attempts have pre-partitioned

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 634–643, 2006.

the sequences into movement sub-blocks, then extracted principal components. This means either considering the entire movement as a primitive, in the domain of handwriting, the entire character, or finding segmentation points, such as points of highest curvature, or maximum torque. The disadvantage of this method is that strong assumptions must be made about the partitioning of the data, and the duration of the primitives. Rather than pre-partitioning the data into sub-blocks, it would be better to allow this partitioning to be inferred.

We have used a probabilistic framework to define a generative model for primitive activation. The parameters for the generative model can be learnt using Expectation-Maximisation. This method provides a way to infer primitives without any pre-partitioning of the data. We hypothesise that motor primitives make up the movement commands being sent to the hand and arm muscles during handwriting. We record a vector describing the position, pressure, and tilt of the pen over time. The dynamics of this vector will reflect the hand motion and therefore contain projections of motor primitives.

In Section 2, we describe the model from motivation to modelling details. Sections 3 and 4 present the data, and the results obtained so far, showing typical primitives obtained, the separation of primitive from primitive timing, using primitives inferred from one dataset to model a different set of data, and some examples of the generative model running without specific timing information. Section 5 discusses the results, the partitioning of the model into primitive and timing parts, and the possible biological parallels with structures in the mammalian brain.

2 The Model

Unlike many previous studies which analysed motor primitives directly from given data, we base our approach on a generative model of motion. With probabilistic inference methods, we infer the primitives inherent in given data. We assume that the activation of motor primitives can be overlapping, and that they do not have uniform fixed length. The primitives are therefore the output vocabulary, which may either be inherited, or learnt over long time-scales. We assume that the primitives have little or no dependence on each other. The independence of one primitive from another, and post-activation persistence of the motor primitives, give rise to a model that is similar in nature to that of a piano. (See Figure 1)

2.1 A Simple Generative Model: The Piano Model

To formalise the model in a generative way, the output of the system Y at time t is defined as

$$Y(t) = \sum_{m,n} \alpha_{mn} W_m(t - \tau_{mn}) \,, \tag{1}$$

where $W_m(t)$ are the primitives, and τ_{mn} represents the time of the n^{th} activation of primitive m, and α_{mn} defines the activation strengths of the primitives.

In this definition, m enumerates the primitives, whilst n enumerates the occurrence of the primitive within the sample window, at time τ_{mn}. We have called this model the Piano Model because of its similarities to the operation of a piano being played, where the timing controller (the pianist) presses each key at the appropriate time in the piece of music. The keys on the piano produce time extended clips of sound, which are superimposed to create the music that is heard by the listener. The crucial point is that

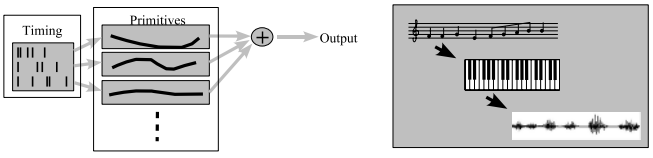


Fig. 1. The Piano Model. The model is segmented into a timing part and a movement subroutine, or primitive part. The timing information is encoded in spike positions, possibly in a similar way to how biological neurons may encode the movement. The movements are encoded as multidimensional muscle activation synergies in biology. Here they have been simplified to one-dimensional signals. The analogy with written music being translated into auditory music is also shown.

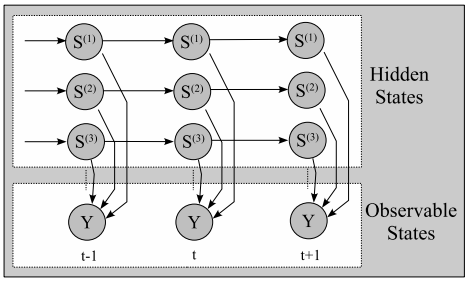


Fig. 2. Graphical representation of a Factorial Hidden Markov Model, showing the independence of the separate Markov chains. Although the observable output is dependent upon the state of the entire system, the internal states evolve with no interdependencies. S_t^m denotes the hidden state vector at time t, in factor m.

the only dependence that the music has on the pianist is the timing and choice of keys he presses, with associated pressure[1]. In the same way, in our motor primitive model, a central controller does not control the precise movements but rather the timings of particular temporal movement sequences. Figure 1 shows a diagram to illustrate the Piano Model.

The Piano Model neglects noise effects and learning the parameters of the model is better realised in a probabilistic framework. Assuming discrete time steps, an appropriate modelling framework is that of Factorial Hidden Markov Models (fHMMs). These are the same as standard HMMs, but with multiple, parallel and independent hidden state chains, as seen in Figure 2.

2.2 The Factorial Hidden Markov Model

A graphical model of the fHMM can be seen in Figure 2. At each time step, the observable output Y_t, a vector of dimension D, is dependent on M hidden variables $S_t^{(1)}, .., S_t^m$. The output is a multivariate Gaussian, such that

[1] This is debatable, as pianos may respond to the speed of key press, the duration of the key press, and what other keys are being pressed at the time. Pianos also have pedals to create different effects. The analogy is not intended to be exact.

$$Y_t \sim \mathcal{N}(\mu_t, C) \,, \tag{2}$$

where C is a $D \times D$ parameter matrix of output covariance, and

$$\mu_t = \sum_{m=1}^{M} W^m S_t^m \tag{3}$$

is the D-dimensional output mean at time t. W^m is a $D \times K$ parameter matrix giving the output means for each factor m, such that the output mean μ_t is a linear combination of its columns weighted with the hidden state activations.

Each of the M hidden variables can be in K different states. In equation (3) this is encoded in the K-dimensional state vector S_t^m using a 1-in-K code, i.e., $S_{t,i}^m = 1$ if the m-th factor is in state i and zero otherwise. This allows us to write expectations of the hidden states as $\langle S_t^m \rangle$, which is also the probability distribution over the individual states S_t^m. Each latent factor is a Markov chain defined by the state transition probabilities and the initial state distribution as

$$P(S_1^m = i) = \pi_i^m \,, \qquad P(S_t^m = i \,|\, S_{t-1}^m = j) = P_{i,j}^m \,, \tag{4}$$

where π^m is a K-dimensional parameter vector giving the initial hidden state distribution, and P^m is a $K \times K$ parameter matrix denoting the state transition probabilities. As can be seen in Figure 2, each factor is independent. This means that the joint probability distribution can be factorised as

$$P(\{Y_t, S_t\}) = P(S_1)P(Y_1|S_1) \prod_{t=2}^{T} P(S_t|S_{t-1})P(Y_t|S_t) \tag{5}$$

$$= \prod_{m=1}^{M} \pi^m P(Y_1|S_1) \prod_{t=2}^{T} \prod_{m=1}^{M} P^m P(Y_t|S_t) \,. \tag{6}$$

The fHMM model was based upon that described in [9], which provides arguments for using a distributed state representation as is used here, and discusses the model in further detail.

To use the fHMM framework as a probabilistic implementation of the Piano Model, we must attribute each hidden Markov chain, or factor, to one primitive. The observables, being real-valued output vectors describing pen position derivatives are modelled by the multivariate output Gaussian distribution, dependent upon the hidden state values at time t. The model has M primitives, with each primitive having K states. This is similar to the Piano Model, given some extra constraints.

2.3 Constraints

The Piano Model differs from the fHMM model by allowing the primitives to be inactive, giving a zero output contribution. In fact, it is a tacit assumption that the primitive activation is fairly sparse, making the periods of inactivity important to the model. In the fHMM model, the output is always a linear combination of all the factors, thus a major constraint imposed was that state 0 for all Markov chains should contribute towards a zero output mean (i.e., to not contribute, as the data mean is zero).

In the Piano Model, each primitive should keep its shape. This means that the possible hidden state transitions in the fHMM model needed to be constrained. When a

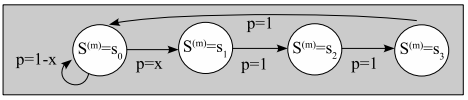


Fig. 3. State change probabilities. This shows how the state change probabilities were constrained so that the individual Markov chains can correspond to a time extended primitive.

particular primitive was triggered, the hidden state vector for that factor changes from state 0 to state 1, then is constrained to progress monotonically through the states until the last state is reached,and returns to state 0. These state change restrictions can be seen graphically in Figure 3.

2.4 Learning the Model

Given the fully parameterised modelling framework, learning of the parameters can be done using an Expectation-Maximisation (EM) method. The structured variational approximation was chosen for the E-step inference. For more details on the various arguments for and against this choice, refer to [9]. With the structured variational approximation, the inference in the fHMM is split up into M separate Hidden Markov Models, with single hidden state chains, with each HMM contributing a learnt proportion towards the output. With separate HMMs, the normal Baum-Welch Forward-Backward algorithm can be used to infer the hidden state expectations [6]. The only addition necessary is a responsibility factor h_t^m, which models the amount that the m^{th} HMM contributes towards the output. h_t^m takes the place of the observation likelihood in a standard HMM. See (7) for the details of calculating h_t. The M-step updates the parameters W^m, π^m, P^m, and C. The update equations are in Appendix A.

3 Implementation

Handwriting data were gathered using an INTUOS 3 WACOM digitisation tablet `http://www.wacom.com/productinfo/9x12.cfm`. This provided 5 dimensional data at 200Hz. The dimensions of the data were x-position, y-position, pen tip pressure, pen tilt angle, and pen orientation (0-360°). The normalised first differential of the data was used, so that the data mean was close to zero, providing the requirements for the zero state assumption in the model constraints (see section 2.1). The data collected were separated into samples, or characters, for processing purposes, and then the parameters were fitted to the data using our algorithm. Once the parameters were learnt, the hidden state expectations $\langle S \rangle$ were finalised, and the pen space reconstruction of the data could be calculated, along with the primitive timing statistics.

To clarify the operation of our algorithm, and the iterative nature of the EM inference, here are the pseudo-code and parameter settings.

Constants. T=times of each sample, N=number of samples, K=max primitive length, M=number of primitives, D=dimension of data.
Initialisations. The primitives W^m are initialized with a zero mean Gaussian distribution of the same variance as the data. The transition probabilities π^m and P^m are set as defined in Section 2.3 with primitive onset probability N/T, giving a prior expectation

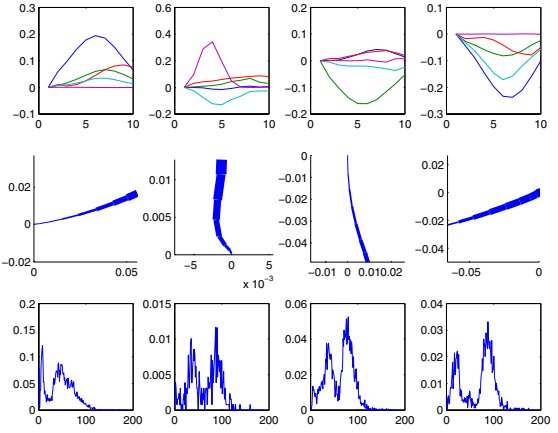


Fig. 4. A sample of 4 typical primitives taken from a set of 36, inferred from the *'g'* dataset. At the top, the 5 dimensional velocity space primitives are shown. The maximum length of these primitives was 10 samples. The centre row shows a pen-space reconstruction of each primitive, with the thickness of the line representing the pressure of the pen tip. The starting point of the reconstruction is at $(0, 0)$. The bottom row shows the distribution of the onset of the primitive over all the character samples.

of each primitive to be used once in each character. The output covariance C is taken directly from the covariance of the data.

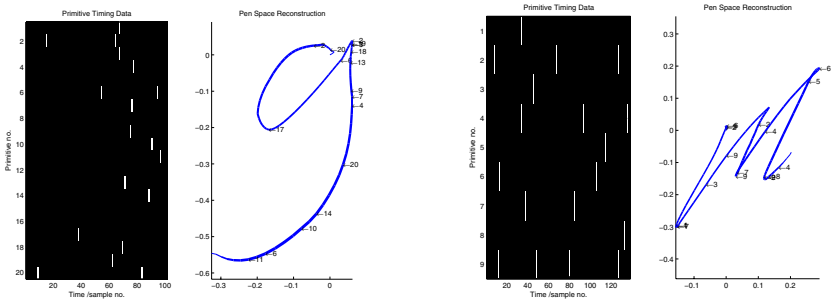
loop (EM loop)
E-step: initialize $\langle S_t^m \rangle$ to $P(S_t^m = 0) = 1$ for all t, m.
  loop
    compute h_t^m from equation (7).
    $\forall m$: forward-backward algorithm gives $\langle S_t^m \rangle$ using h_t^m as obs. likelihood.
  until expectations not changing significantly, or max 20 iterations
M-step: update the parameters as in (10)
until primitives don't changing significantly, or max 50 iterations.

A large dataset of over 1000 samples of the character *'g'* was created, and a smaller dataset of over 100 samples of the character *'m'* was also used, for speed issues. To examine whether primitives can be disassociated from any particular character set, a dataset of over 100 samples of scribbling was also created.

4 Results

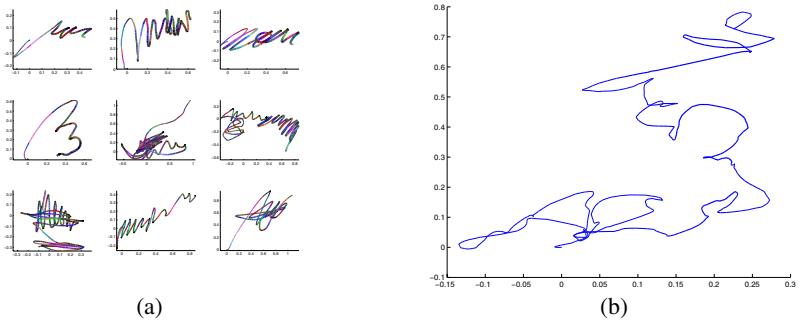
Typical primitives. From the *'g'* samples, it was found that the primitives tend to model similar parts of the character across most, but not all samples. In Figure 4, we can see a sample of primitives in velocity and their pen-space reconstructions along with their timing distributions, from a set used to model the *'g'* samples dataset.

Compact encoding with onset timing. $P(S_t^m)$ for state 1 represents the onset probability at time t for the m^{th} primitive, and are inferred during the E-step. Figure 5 shows this timing information, which efficiently encodes the reproduction of the sample shown.



(a) (b)

Fig. 5. Two examples of primitive timing codes. In both (a) and (b), the timing information is shown on the left, and the reproduction of the sample on the right, with the onset of each primitive marked with an arrow. In (a), 20 primitives of length 40 time steps were used to model the *'g'* dataset, of over 1000 characters. In (b), 9 primitives of length 30 were used to model the *'m'* dataset, of over 100 characters.



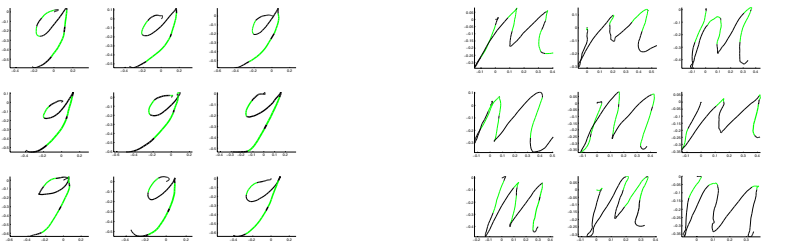
(a) (b)

Fig. 6. A reconstruction of some samples from the scribble dataset is shown in (a). This was a set of unconstrained scribbles. The primitives are coloured differently, to show different activation areas. (b) shows a sample generated using primitives from this dataset. Starting point of reconstruction is $(0, 0)$.

The distributions of activation of a primitive over the different samples tends to be bell shaped, either uni- or bimodal, meaning that the primitives have a preference for a particular part of the character, with the variation in timing accounting for the variations across characters, at least to a certain extent.

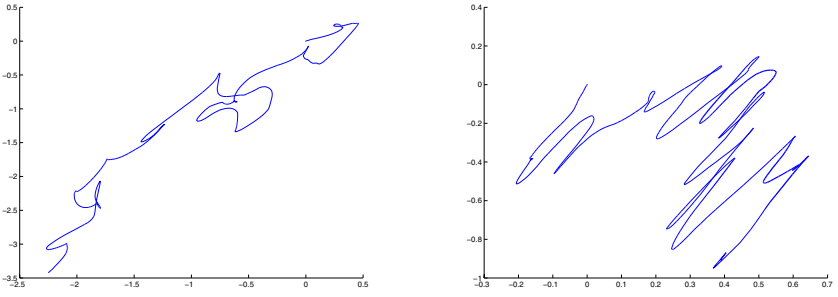
Primitives from scribbling to generate characters. To explore whether the algorithm was picking up features of a particular character set, or more generalised motor primitives, a dataset of scribbles were used. These consisted of unconstrained scribbling, without any character set goal objectives. Our algorithm was run on this dataset, and in Figure 6(a), we can see a reconstruction of scribble samples.

Using the primitives from this dataset, it was possible to represent the other dataset. In Figure 7, we can see examples of the character *'g'*, and *'m'* being drawn with scribble primitives.



(a) (b)

Fig. 7. A reconstruction of some samples, using primitives from the scribble set. One of the 36 primitives is highlighted. (a) shows characters from the *'g'* dataset, (b) shows characters from the *'m'* dataset.



(a) (b)

Fig. 8. Two samples generated using primitives without specific timing information. (a) was generated using primitives inferred from the *'g'* dataset, (b) was generated using primitives from the *'m'* dataset. Starting point of both reconstructions is $(0, 0)$.

Random expression of primitives with different timing statistics. To further explore what aspects of the character set are captured by the primitives, it is possible to simply run the model generatively, using the inferred parameters. In Figure 8 we see two samples generated using the sets of primitives and other associated parameters inferred from the *'g'* and *'m'* datasets. In both samples, we see aspects of the character, but no clear examples of a character drawn perfectly. This is partly because the primitives are assumed to be generatively independent, meaning the primitive that models the start of the character cannot convey information about its state to the primitive modelling the subsequent part. The primitives, although capturing an 'aspect' of the character, lack the precise timing information that dictates how the character is drawn. This timing information can be seen in Figure 5.

The generative scribbling could be likened to absent-minded doodles, where we control a pen, and produce writing output, but without any constraints dictating what character we should draw. Indeed, it is impossible to tell whether or not the sample shown in Figure 6(b) is from a scribbled dataset drawn by a human, or a generative sample 'drawn' by the algorithm.

5 Discussion and Conclusions

It is possible to model characters with primitives using a Factorial Hidden Markov Model that does not pre-specify the timing of the primitives. The primitives, although not constrained to be active for the whole character, or to be active between pre-defined segmentation points in the character, tend to model particular areas of the character. The variation of the character within the dataset is modelled partly by variations in primitive timing, and partly by different primitives being active.

The primitives do capture an aspect of the character set they were trained upon, although, they do not contain the precise timing information required for the reproduction of a character accurately. This timing information can be learnt by looking at the primitive activation statistics taken from a particular dataset, (work in progress), and forms a compact representation of the character, as it simply encodes the onset timing of each primitive. Without this timing information, the generative output of the model acquires the aspect of scribbling. It is possible that when humans doodle absent-mindedly, it is a lack of timing information that is causing the scribble type output from the motor system.

This model lends itself towards breaking up of the motor system into 2 main modules. Firstly, the Primitive Module, encompassing parameters such as likelihood of a particular motion, likely amplitude of a particular motion, with implementation capabilities (control of muscles) of segmented, independent motions. Secondly, the Timing Module, encoding the overall motor 'strategy', and allowing compensation through feedback, and online error correction to the timing of the motions, rather than the actual motions themselves. In the brain, this segmentation could be paralleled by the motor cortex and lower motor systems encoding the actual motor commands, while the cerebellum encodes the motor command timing. There is strong evidence to suggest that the cerebellum is involved not only with motor timing, but perception of timed events [13, 7, 14]. As the lack of a cerebellum does not completely impair movement, this implies that muscle control is located elsewhere, such as in the motor cortex.

References

[1] E. Bizzi, A. d'Avella, P. Saltiel, and M. Trensch. Modular organization of spinal motor systems. *The Neuroscientist*, 8(5):437–442, 2002.

[2] E. Bizzi, S.F. Giszter, E. Loeb, F.A. Mussa-Ivaldi, and P. Saltiel. Modular organization of motor behavior in the frog's spinal cord. *Trends in Neurosciences*, 18(10):442–446, 1995.

[3] A. d'Avella and E. Bizzi. Shared and specific muscle synergies in natural motor behaviors. *PNAS*, 102(8):3076–3081, 2005.

[4] A. d'Avella, P. Saltiel, and E. Bizzi. Combinations of muscle synergies in the construction of a natural motor behavior. *Nature Neuroscience*, 6(3):300–308, 2003.

[5] P. R. Davidson and D. M. Wolpert. Motor learning and prediction in a variable environment. *Curr. Opinion in Neurobiology*, 13:1–6, 2003.

[6] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm (with discussion). *J. R. Statist. Soc. B*, 39:1–38, 1977.

[7] M. Dennis, K. Edelstein, R. Hetherington, K. Copeland, J. Frederick, S. E. Blaser, L.A. Kramer, J.M. Drake, M. Brandt, and J. M. Fletcher. Neurobiology of perceptual and motor timing in children with spina bifida in relation to cerebellar volume. *Brain*, 2004.

[8] A. Fod, M.J. Mataric, and O.C. Jenkins. Automated derivation of primitives for movement classification. *Autonomous robots*, 12(1):39–54, 2002.

[9] Z. Ghahramani and M.I. Jordan. Factorial hidden Markov models. *Machine Learning*, 29:245–275, 1997.
[10] A. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor lanscapes for motor primitives. *MIT Press*, 15, 2003.
[11] W.J. Kargo and S.F. Giszter. Rapid corrections of aimed movements by combination of force-field primitives. *J. Neurosci.*, 20:409–426, 2000.
[12] M.J. Matarić. Primitives-based humanoid control and imitation. Technical report, DARPA MARS project, 2004.
[13] D.V. Meegan, R.N. Aslin, and R. A. Jacobs. Motor timing learned without motor training. *Nature Neuroscience*, 3(9):860–862, 2000.
[14] V.B. Penhume, R.J. Zatorre, and A.C. Evans. Cerebellar contributions to motor timing: A pet study of auditory and visual rythm reproduction. *Journal of Cognative Neuroscience*, 10(6):752–765, 1998.
[15] S.P. Schaal, J. Nakanishi, and A. Ijspeert. Learning movement primitives. In *ISRR2003*, 2004.
[16] E. Todorov and M.I. Jordan. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5(11):1226–1235, 2002.
[17] D. M. Wolpert, Z. Ghahramani, and J. R. Flanagan. Perspectives and problems in motor learning. *TRENDS in Cog. Sci.*, 5(11):487–494, 2001.
[18] D. M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11:1317–1329, 1998.

A Appendix

In the E-step, the responsibility factor, h_t was calculated using a residual error,

$$h_t^{(m)new} = \exp\{W^{(m)T} C^{-1} \tilde{Y}_t^{(m)} - \frac{1}{2} \triangle^{(m)}\} \tag{7}$$

$$\triangle^{(m)} \equiv \text{diag}(W^{(m)T} C^{-1} W^{(m)}) \tag{8}$$

$$\tilde{Y}_t^{(m)} \equiv Y_t - \sum_{l \neq m}^{M} W^{(l)} \langle S_t^{(l)} \rangle \tag{9}$$

where $\tilde{Y}_t$ is the residual error.

In the M-step, the parameter update equations used were

$$W \leftarrow (\sum_{t=1}^{T} Y_t \langle S_t^T \rangle)(\sum_{t=1}^{T} \langle S_t S_t^T \rangle)^{\dagger} \tag{10}$$

$$\pi^{(m)} \leftarrow \langle S_1^{(m)} \rangle \qquad P_{i,j}^{(m)} \leftarrow \frac{\sum_{t=2}^{T} \langle S_{t,i}^{(m)} S_{t-1,j}^{(m)} \rangle}{\sum_{t=2}^{T} \langle S_{t-1,j}^{(m)} \rangle} \tag{11}$$

$$C \leftarrow \frac{1}{T} \sum_{t=1}^{T} Y_t Y_t^T - \frac{1}{T} \sum_{t=1}^{T} \sum_{m=1}^{M} W^{(m)} \langle S_t^{(m)} \rangle Y_t^T \tag{12}$$

$\langle S_t \rangle$ is the expected value of the hidden states at time t. $\dagger$ denotes *pseudo-inverse*.

Including Metric Space Topology in Neural Networks Training by Ordering Patterns

Cezary Dendek and Jacek Mańdziuk

Faculty of Mathematics and Information Science,
Warsaw University of Technology,
Plac Politechniki 1, 00-661 Warsaw, Poland
dendekc@student.mini.pw.edu.pl, mandziuk@mini.pw.edu.pl
http://mini.pw.edu.pl/~mandziuk/

Abstract. In this paper a new approach to the problem of ordering data in neural network training is presented. According to conducted research, generalization error visibly depends on the order of the training examples. Construction of an order gives some possibility to incorporate knowledge about structure of input and output space into the training process. Simulation results conducted for the isolated handwritten digit recognition problem confirmed the above claims.

1 Introduction

The problem of optimal ordering of the training data has a great meaning in sequential supervised learning. It has been shown ([1],[2]), that improper order of elements in the training process can lead to catastrophic interference. This mechanism can also occur during each training epoch and disturb neural network training process. Random order of elements prevents from interference but can lead to non-optimal generalization. Consequently, for example, most of efficient algorithms for training RBF networks arbitrarily choose initial patterns ([3]).

In this paper a new approach to patterns ordering is proposed and experimentally evaluated in the context of supervised training with feed-forward neural networks. The idea relies on *interleaving two training sequences: one of particular order and the other one chosen at random.*

In order to show the feasibility of this approach four models of an order are defined in the next section together with a sample test problem - isolated handwritten digit recognition. Numerical results of proposed *interleaved* training are presented in Sect. 2. Conclusions and directions for future research are placed in the last section.

Input and output spaces of a network can be considered as metric spaces. It is always possible to introduce a metrics since each of them can be immersed in $\mathbb{R}^n$ (where n is a space dimension) with natural metrics

$$M : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+ \cup \{0\}, \qquad M : ((x)_{k=1}^n, (y)_{k=1}^n) \mapsto \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}.$$

Moreover, if some other knowledge about the data is possessed - e.g. if input data consists of $p, (p > 1)$ values of different scales - metrics with normalization or non-euclidean metrics may be used, which would model the space considerably better.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 644–653, 2006.

In such a case it is possible to divide a space into p subspaces and calculate the following metrics:

$$M : (x, y) \mapsto \sqrt{\sum_{i=1}^{p} \left(\frac{d_i(x, y)}{\overline{d_i}} \right)^2}, \tag{1}$$

where $d_i(x, y)$ denotes the distance between elements x and y according to the i-th metric and $\overline{d_i}$ represents the average pairwise distance between elements belonging to the i-th scale data.

After choosing and normalizing metrics on input and output spaces it is possible to introduce metrics on pattern space as it was done on space divided into subspaces.

1.1 Four Models of an Order

In this section four schemes of ordering training patterns together with their characteristics are introduced.

Let input and output spaces be denoted by I and O, resp., and let $\{T_k\}$ be the set of training patterns. The models presented below rely on the fact that given a metrical space of patterns it is possible to determine a pattern that is the nearest to the center of the average probability of occurrence - analogously to the *mass center point*.

Model I. Let us denote by S_k^I a sum of distances from a given element T_k to the rest of elements of the set:

$$S_k^I = \sum_{l=1}^{n} M(T_k, T_l)$$

A sequence of q training patterns $(T_l)_{l=1}^{q}$ that fulfils the following set of inequalities:

$$\forall_{1 \le l \le q-1} \qquad S_l^I \ge S_{l+1}^I \tag{2}$$

is called *ordered set of model I*. A sequence created with rule (2) will begin with outlying patterns and end with close-to-average ones. Since the ending of the sequence finally tunes weights of the network (and if not randomly chosen can have a biased impact on the final network's weights) it is important to characterize these average elements. In the space of patterns an ending of the sequence is expected to concentrate on neighborhoods that are chosen combining two following tendencies of concentration:

1. *Concentration on global maxima of probability density*. In such a neighborhood an average distance should be minimized.
2. *Concentration on geometrical centers*. These points minimize the sum of distances to all other points. If probability of patterns is uniformly distributed the sequence ending would be concentrated on geometrical centers.

In case of multicluster data it is expected that the training sequence ending would be dominated by elements of one of the clusters (except for the cases of symmetrical distributions of clusters). In such a case the sequence ordered in the above way will generalize an approximated function better than a randomly ordered sequence only on elements of preferred cluster.

Since the construction of an ordered set according to *model I* is straightforward its description is omitted.

Model II. Given a metrics M defined on pattern space and a set $\{T_k\}$ an average pairwise distance S_n^{II} between the first n elements of the sequence can be expressed as:

$$S_n^{II} = \frac{2}{(n-1)n} \sum_{k=1}^{n} \sum_{l=k+1}^{n} M(T_k, T_l).$$

A sequence of q training patterns $(T_l)_{l=1}^{q}$ that fulfils the set of inequalities:

$$\forall_{1 \leq l \leq q-1} \qquad S_l^{II} \geq S_{l+1}^{II} \tag{3}$$

is called *ordered set of model II*. Similarly to the previous model a sequence created with rule (3) is expected to prefer outlying patterns at the beginning of the sequence and place the average ones at the sequence ending. Rule (3) is more sensitive to geometrical centers than probability centers compared to rule (2). A reason for such statement is an observation that elements in the sequence ordered using rule (3) that occur after given element do not have an influence on its position (as if they had been removed from the set). What is more, a selection of an element according to presented algorithm implies that the difference in the average distance after selection is minimal - the change of geometrical center of a set should also be small. Removal of an element changes local density of probability.

Algorithm for ordering a set in Model II. Given set $\{T_k\}$ can be ordered to sufficiently approximate *ordered set of model II* with the use of the following algorithm:

1. Put all q elements in any sequence $(T_l)_{l=1}^{q}$.
2. Create an empty sequence O.
3. Create distance array $D[1..q]$:

$$\forall_{1 \leq l \leq q} \qquad D_l := \sum_{k=1}^{q} M(T_l, T_k)$$

4. Choose a minimal value of element of D:

$$v := min_{1 \leq l \leq q} \qquad D_l.$$

5. Pick one element k from the set $\{1 \leq l \leq q \,|\, D_l = v\}$.
6. Update distance matrix:

$$\forall_{1 \leq l \leq q} \qquad D_l := D_l - M(T_k, T_l)$$

7. Take element T_k out of sequence T and place it at the beginning of sequence O.
8. Remove element D_k from distance array.
9. Put $q := q - 1$.
10. Repeat steps 4-10 until $q = 0$.

Model III. *Ordered set of model III* is obtained by reverting *ordered set of model I.*
Model IV. *Ordered set of model IV* is obtained by reverting *ordered set of model II.*

1.2 Test Problem

In order to test an influence of training data ordering on the learning process, a sample problem consisting in isolated handwritten digits recognition was chosen. The pattern set consisted of 6000 elements, randomly divided into training set T, $|T| = 5500$ and test set V, $|V| = 500$. Binary $\{0, 1\}$ input vectors of size 19×17 represented bitmaps of patterns, and the 10-element binary $\{0, 1\}$ output vector represented the classification of the input digit (i.e. exactly one of its elements was equal to 1). All patterns were centered. It should be noted that no other preprocessing took place. In particular digits were not scaled, rotated or skewed appropriately. A detailed description of this data set and results achieved by other recognition approaches can be found in [4]. A general overview of methods applied to handwritten text recognition can be found in [5].

An ensemble of neural networks with 1 hidden layer composed of 30 neurons was trained using backpropagation method. Both hidden and output neurons were sigmoidal.

Input subspace became metrical with the use of the following metrics:

$$I(v, w) = min_{x,y \in \{-2,-1,0,1,2\}} H(v, R(x, y, w)) + |x| + |y|$$

where $H(\cdot, \cdot)$ denotes Hamming distance, and $R(x, y, w)$ denotes translation of vector w by x rows and y columns. In the output subspace a discrete metrics $O(v, w)$ was used. Based on metrics defined on subspaces a metrics on pattern space was defined according to (1) as follows:

$$M : (x, y) \mapsto \sqrt{\left(\frac{I(x, y)}{\overline{I}}\right)^2 + \left(\frac{O(x, y)}{\overline{O}}\right)^2}.$$

For the training set it was obtained $\overline{I} = 62.55$, $\overline{O} = 0.9$.

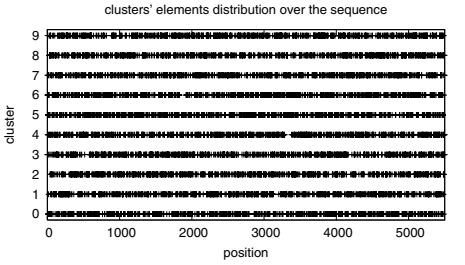
2 Results

All numerical results concerning RMSE and STDEV are presented as the averages over 100 networks, each with randomly selected initial weights. Unless otherwise stated each training period was composed of 600 epochs. For comparison purposes all figures representing one pass of training/testing for different orders of the training data are presented for the same, randomly selected network (i.e. with the same initial choice of weights).

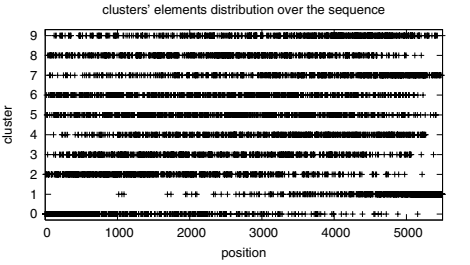
According to previously formulated hypothesis in case of ordered sequences elements of particular clusters were not uniformly distributed over the sequence, which is illustrated in Fig. 1. For example, elements representing digits 1 and 7 are concentrated at the endings of both ordered sequences (Fig. 1(b) and Fig. 1(c)) and elements representing 0 and 2 are located mainly at the beginnings, whereas distributions of all digits in case of random order (Fig. 1(a)) are uniform. Distributions of ordered sequences are similar to each other, but they remarkably differ on digits 8 and 9.

2.1 Initial Results for Pure Random and Ordered Training Data

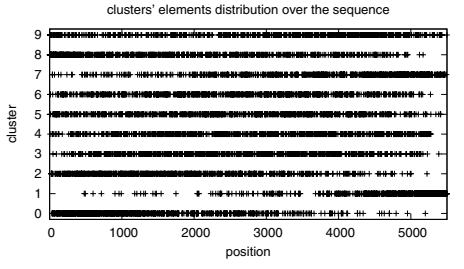
The case of randomly ordered training data (henceforth referred to as *pure random* case) proves that the considered problem can be solved using assumed network architecture



(a) randomly ordered

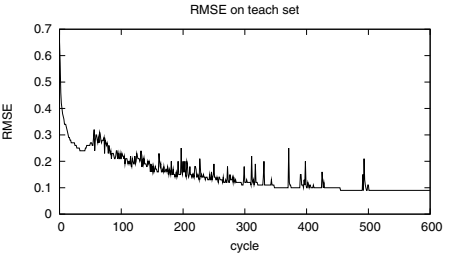


(b) ordered of *model I*



(c) ordered of *model II*

Fig. 1. Clusters' elements distribution over sequences



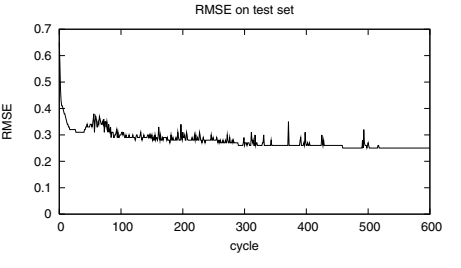


Fig. 2. RMSE obtained in each cycle using randomly ordered training data

and learning algorithm. This case also provides a possibility of comparison between ordered training models and the pure random one. The plot of RMSE of the training and test data in pure random order case are presented in Fig. 2.

The plots of RMSE of the network trained with sequence ordered according to *model II* are presented in Fig. 3. **It can be concluded from the figure that convergence of training is worse compared to random order**.

In hope to improve the convergence of the training process switching of training sequences with a random one was tried. Fig. 4 presents changes of RMSE in case the first 300 training epochs was performed with the sequence defined according to *model IV*, which was then replaced with a randomly ordered sequence for the remaining 300 epochs. **Please note the high decrease of the error in the middle of the plot - i.e. when the *model IV* training sequence was replaced by the random one.**

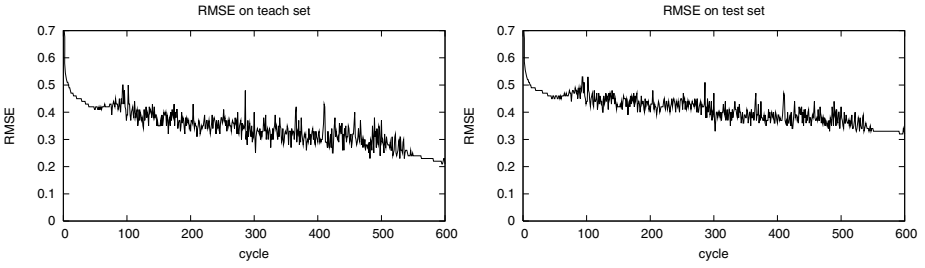


Fig. 3. RMSE when using training data ordered according to *Model II*

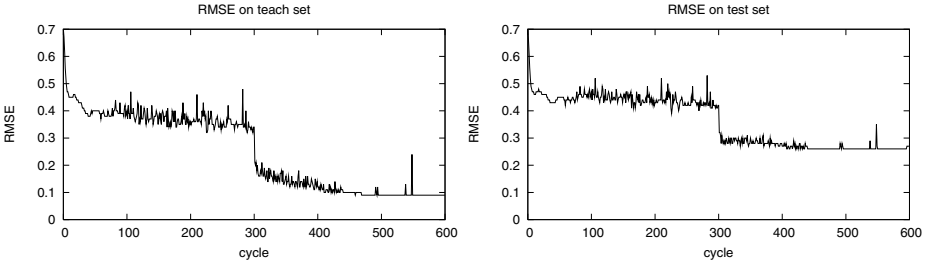


Fig. 4. RMSE in case the training data ordered according to *Model IV* is used in the first 300 epochs followed by training with the random sequence in the remaining 300 epochs

Following the idea of switchings sequences applied in the previous experiment a simulation of the training process with more frequent sequence exchange was performed. In this case the sequence ordered according to *model II* was exchanged with the random sequence after every 20 training epochs. The results in terms of RMSE plot are presented in Fig. 5. A comparison of RMSE in the above case with a pure random case is presented in Fig. 6. It is remarkable that **after each alteration of *model II* sequence with a random one RMSE becomes lower than in pure random case**. The possible explanation is that non-uniformity of elements' distribution has the effect in local changes of weights' change direction during presentation of training sequences which consequently allows the network to escape from local shallow minima.

2.2 Proposition of Training Sequence Switching

Due to observed activity of ordered sequences it should be considered to interleave them with random ones in the training process. It is therefore proposed to apply a model with decreasing probability of using ordered sequences in the training process. Let

$$P(t) = pe^{-\eta t} \tag{4}$$

be the probability of presenting ordered sequence, where t is the number of the training epoch, p - the initial probability, η - positive coefficient of probability decrease. Having two training sequences - one ordered according to any of the above described four

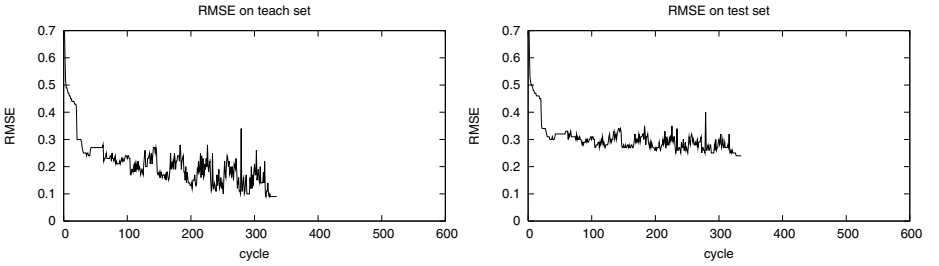


Fig. 5. RMSE in case when training data ordered according to *Model II* is periodically (after every 20 cycles) exchanged with a random sequence

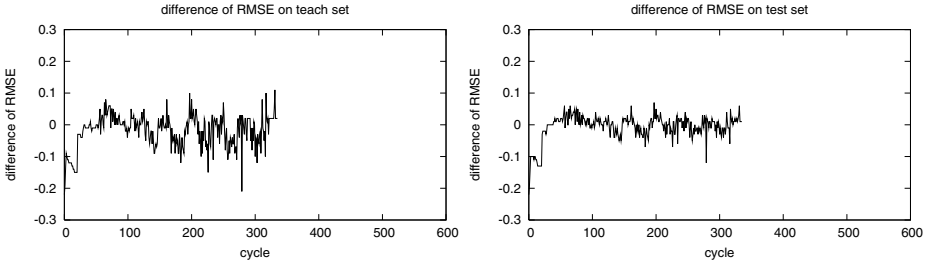


Fig. 6. Difference between RMSEs calculated in Fig. 2 (pure random case) and Fig. 5 (*model II* interleaved with random sequence)

models and the other one being a purely random) in each epoch the ordered training sequence is chosen according to the above probability. Since the remainder of the paper will be devoted to the proposed algorithm, henceforth, *model I*, *model II*, *model III* and *model IV* will refer to the above training method in which the respectively ordered sequence is interleaved with the random one. As a special case also two randomly chosen (fixed) sequences are considered as the two interleaved sequences. This case will be denoted by *switched random*.

3 Performance of Proposed Algorithm

In each case training process consisted of 600 epochs, initial probability p was equal to 1.0 and η was chosen so that $P(600) = 0.03$.

3.1 Independent Training

Statistics (mean RMSEs and Standard Deviations) of populations of neural networks obtained by training with given model of an order are presented in Table 1. Sequences are ordered according to RMSE values on the test set. Visualization of the RMSE values is presented Figure 4, in which all populations are presented. Each dot represents one neural network. Initial weights of these networks were independently chosen at random.

Table 1. Statistics of RMSE

model	*mean RMSE on train set*	*SD RMSE on train set*	*mean RMSE on test set*	*SD RMSE on test set*
model III	0.0798	0.0140	0.2591	0.0109
model I	0.0844	0.0115	0.2602	0.0116
model IV	0.0818	0.0138	0.2621	0.0098
model II	0.0841	0.0206	0.2640	0.0128
switched random	0.0882	0.0244	0.2640	0.0165
pure random	0.0939	0.0209	0.2668	0.0118

It is remarkable that **among populations obtained with use of randomly ordered sequences and ones obtained using sequences ordered according to proposed models exists a statistically significant difference**. P-values for hypothesis about significant difference are presented in Table 2.

Table 2. P-value of hypothesis that distributions of RMSE on the training set are different

	model III	*switched random*	*model IV*	*model II*	*model I*	*pure random*
model III	1					
switched random	0.002	1				
model IV	0.288	0.0170	1			
model II	0.069	0.1688	0.3256	1		
model I	0.009	0.1449	0.130	0.874	1	
pure random	0.000	0.0613	0.000	0.000	0.000	1

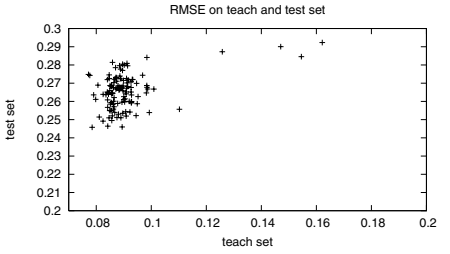
It can be concluded that an improvement of average RMSE in the best case of randomly ordered sequence and the best case of the ordered one (*model III* vs *switched random*) is equal to 9.52% and 1.84%, resp. on the training and tests sets.

The average pattern classification result of the best model (*model III*) on the test set was equal to 92.55%.

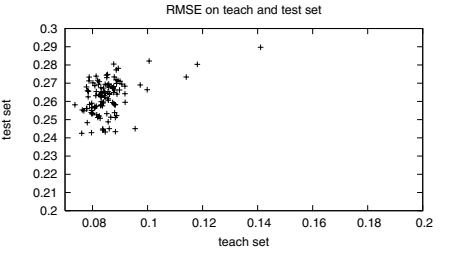
3.2 Training Represented as Dependent Variables

In order to eliminate randomness of neural network initial weights (which can be considered as a noise in case of independent samples) the research of dependent samples was performed. Population consisted of 64 neural networks and each of them has been trained 6 times (once for each training model) - each time with the same set of initial weights.

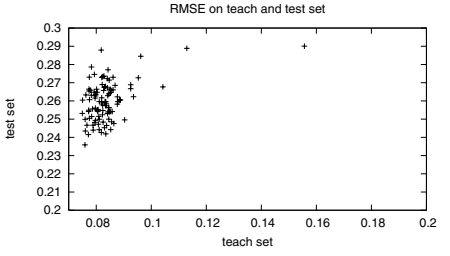
In order to analyze the influence of ordering on possibility of obtaining a network with good generalization abilities the top 20 recognition results on the test set were selected. The average and the maximum pattern classification results on the test set of these networks were equal to 93.93% and 94.49%, resp. Fractions of networks trained according to particular models' orders, which belonged to this group are presented in Table 3. Note, that **reverted models are dominating (**70%**)** and also **no neural network trained exclusively with random sequences has qualified to the set**.



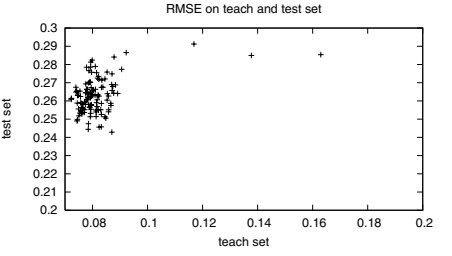
(a) pure random sequence (no switching)



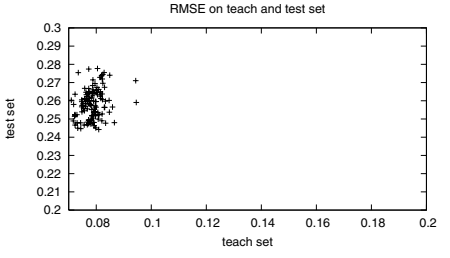
(b) switched random



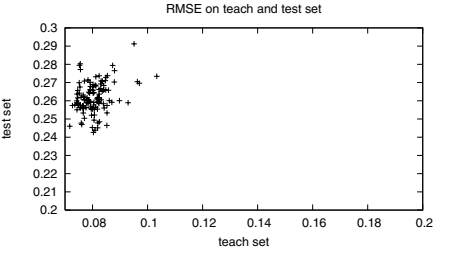
(c) data ordered according to *Model I*



(d) data ordered according to *Model II*



(e) data ordered according to *Model III*



(f) data ordered according to *Model IV*

Fig. 7. RMSE on training and test sets in last epoch in case of using ordered sequence switched by random one according to formula (4)

Table 3. Percentage of sequences among the top 20 networks on the test set

	model III	*switched random*	*model IV*	*model II*	*model I*	*pure random*
percentage	25%	0%	45%	10%	20%	0%

4 Conclusions

The problem of ordering training patterns is essential in supervised learning with neural networks. In the paper a new method of ordering training patterns is proposed and experimentally evaluated. It was shown that proposed approach produces in average

better results than training without its use in the sample problem representing clustered pattern space. Some theoretical considerations supporting this result has been provided.

Tests in other problem domains are under research. Other possible uses of ordered sequences (e.g. as a measure of generalization ability of network architecture) are considered as future research plans.

Acknowledgment

This work was supported by the Warsaw University of Technology under grant no. 504G 1120 0008 000. Computations were performed using grid provided by Enabling Grids for E-sciencE (EGEE) project.

References

1. Ratcliff, R.: Connectionist Models of Recognition Memory: Constraints Imposed by Learning and Forgetting Functions. Psychological Review **97**(2) (1990) 285–308
2. French, R. M.: Catastrophic Forgetting in Connectionist Networks. Trends in Cognitive Sciences **3**(4) (1999) 128–135
3. de Carvalho, A., Brizzotti M. M. : Combining RBF Networks Trained by Different Clustering Techniques. Neural Processing Letters **14**(3) (2001) 227–240
4. Mańdziuk, J., Shastri, L.: Incremental Class Learning approach and its application to Handwritten Digit Recognition. Information Sciences **141**(3-4) (2002) 193–217
5. Bunke, H.: Recognition of Cursive Roman Handwriting - Past, Present and Future. In: International Conference on Document Analysis and Recognition (ICDAR'03), (2003) 448–459

A Technical Trading Indicator Based on Dynamical Consistent Neural Networks

Hans Georg Zimmermann[1], Lorenzo Bertolini[2,3], Ralph Grothmann[1], Anton Maximilian Schäfer[1,4], and Christoph Tietz[1]

[1] Information & Communications, Learning Systems
Siemens AG, Corporate Technology, 81739 Munich, Germany
Hans_Georg.Zimmermann@siemens.com
[2] JPMorgan, London, UK
[3] DekaBank, Frankfurt, Germany
[4] University of Ulm, Optimisation and Operations Research, Germany

Abstract. In econometrics, the behaviour of financial markets is described by quantitative variables. Mathematical and statistical methods are used to explore economic relationships and to forecast the future market development. However, econometric modeling is often limited to a single financial market. In the age of globalisation financial markets are highly interrelated and thus, single market analyses are misleading. In this paper we present a new way to model the dynamics of coherent financial markets. Our approach is based on so-called dynamical consistent neural networks (DCNN), which are able to map multiple scales and different sub-dynamics of the coherent market movement. Unlikely to standard econometric methods, small market movements are not treated as noise but as valuable market information. We apply the DCNN to forecast monthly movements of major foreign exchange (FX) rates. Based on the DCNN forecasts we develop a technical trading indicator to support investment decisions.

1 Introduction

Recurrent neural networks (RNNs) allow the identification of any open dynamical systems in form of high dimensional, nonlinear state space models [1,2]. However, the question often arises if the standard RNNs are a sufficient framework for the modeling of complex nonlinear and high dimensional dynamical systems like financial markets, which can only be understood by analysing the interrelationship of different subdynamics. Consider the following economic example: The dynamics of the USDEUR foreign exchange (FX) market is clearly influenced by the development of other major FX, stock or commodity markets [3]. In other words, movements of the USDEUR FX rate can only be comprehended by a combined analysis of the behavior of other markets. This means that a model of the USDEUR FX market must also learn the dynamics of related markets and intermarket dependencies.

In this paper we present a new way to model the dynamics of coherent financial markets. Our approach is based on so-called dynamical consistent neural networks (DCNN), which are able to map multiple scales and different sub-dynamics of the coherent market movement. In addition, unlikely to standard econometric methods, small market movements are not treated as noise but as valuable market information.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 654–663, 2006.

We successfully apply the DCNN to forecast monthly movements of major FX rates. Based on the DCNN forecasts we develop a technical trading indicator to support investment decisions.

2 Modeling the Dynamics of Observables

Standard recurrent neural networks (RNNs) [4] are able to forecast single time series. Still, the difficulty with RNN, especially in high-dimensions, is the training with backpropagation through time [5], because we do not have the same learning behavior for the weight matrices in the different time steps [6]. Besides that, inputs and outputs are considered independently. This distinction between externals u_τ and the network output y_τ is arbitrary and mainly depends on the application or the view of the model builder instead of the real underlying dynamical system.

Therefore, the following model merges inputs and targets into one group of variables, which we call observables y_τ, and incorporates besides a bias c only one connector type, a single transition matrix A. In doing so, we now look at the network as a high dimensional dynamical system where the dynamics is modelled by the single transition matrix A and input and output represent the observable variables of the environment. We arrive at an integrated view of the dynamical system:

$$\begin{aligned} \tau \leq t: \quad & s_\tau = \tanh(As_{\tau-1} + c + \begin{bmatrix} 0 \\ 0 \\ Id \end{bmatrix} y_\tau^d) \\ \tau > t: \quad & s_\tau = \tanh(As_{\tau-1} + c) \\ & y_\tau = [Id\ 0\ 0]s_\tau \\ \sum_t \sum_\tau & (y_\tau - y_\tau^d)^2 \rightarrow \min_{A,c} \end{aligned} \tag{1}$$

The corresponding model architecture is shown in figure 1:

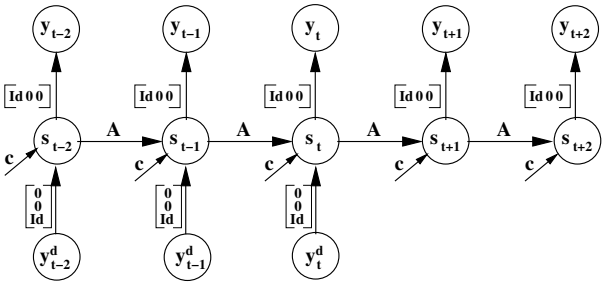


Fig. 1. Normalised recurrent neural network, unfolded in time and with overshooting [4,6], modeling the dynamics of observables y_τ^d

Note, that because of the one step time delay between input and output, y_τ^d and y_τ are not directly connected. Furthermore it is important to understand, that we now take a totally different view on the dynamical system. In contrast to standard RNN, this network (eq. 1) not only generates forecasts for the dynamics of interest but for all external observables y_τ^d. Consequently, the first r state neurons are used for the identification of the network outputs. They are followed by q computational hidden neurons and r state neurons which read in the external inputs.

3 Dynamical Consistent Neural Networks (DCNN)

When the dynamics of an open dynamical system is iterated into the future, the development of the system environment is unknown. In this context, one of the standard statistical paradigms is to assume, that the external influences are not significantly changing in the future. This means, that the expected value of a shift in an external input y_τ^d with $\tau > t$ is zero per definition. For that reason, we have so far neglected the external inputs y_τ^d at all future time steps, $\tau > t$, of the unfolding (eq. 1).

Especially when we consider fast changing external variables with a high impact on the dynamics of interest, the latter assumption is very questionable. Considering equation 1 it even poses a contradiction, as the observables are assumed to be constant on the input and changing on the output side. The model is therefore not consistent from a dynamical point of view. The longer the forecast horizon is, the more the statistical assumption is violated. Even in case of a slowly changing environment, long-term forecasts become doubtful. For a dynamical consistent approach, one has to integrate assumptions about the future development of the environment into the modeling.

We propose a network that uses its own predictions as replacements for the unknown future observables. This is modeled with an additional fixed matrix in the state equation. The resulting dynamical consistent neural network (DCNN) is shown in equation 2:

$$\begin{aligned}
\tau \le t: \quad s_\tau &= C_{\le} \quad \tanh(As_{\tau-1} + c) + \begin{bmatrix} 0 \\ 0 \\ Id \end{bmatrix} y_\tau^d \\
\tau > t: \quad s_\tau &= C_{>} \quad \tanh(As_{\tau-1} + c) \\
y_\tau &= [Id\ 0\ 0] \quad s_\tau \\
\sum_t \sum_\tau (y_\tau - y_\tau^d)^2 &\to \min_{A,c}
\end{aligned} \tag{2}$$

The recursion of the state equations (eq. 2) acts in the past ($\tau \le t$) and future ($\tau > t$) always on the same partitioning of the inner state vector s_τ which can, for all τ, be described as:

$$s_\tau = \begin{bmatrix} y_\tau \\ h_\tau \\ \left\{ \begin{array}{l} \tau \le t: \ y_\tau^d \\ \tau > t: \ y_\tau \end{array} \right\} \end{bmatrix} = \begin{bmatrix} \text{expectations} \\ \text{hidden states} \\ \left\{ \begin{array}{l} \tau \le t: \ \text{observations} \\ \tau > t: \ \text{expectations} \end{array} \right\} \end{bmatrix} \tag{3}$$

That means, that in the first r components of the state vector we have the expectations y_τ, i.e. the predictions of the model. The q components in the middle of the vector represent the hidden units h_τ. They are actually responsible for the description of the dynamics. In the last r components of the vector we find in the past ($\tau \leq t$) the observables y_τ^d, which the model receives as external input. In the future ($\tau > t$) the model replaces the unknown observables by its own expectations y_τ. This replacement is modelled with two consistency matrices:

$$C_{\leq} = \begin{bmatrix} Id & 0 & 0 \\ 0 & Id & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } C_{>} = \begin{bmatrix} Id & 0 & 0 \\ 0 & Id & 0 \\ Id & 0 & 0 \end{bmatrix} \tag{4}$$

Let us explain one recursion of the state equation (eq. 2) step by step: In the past ($\tau \leq t$) we start with a state vector $s_{\tau-1}$, which has the structure as described in equation 3. This vector is first multiplied with the transition matrix A. After adding the bias c, the vector is sent through the nonlinearity $\tanh$. The consistency matrix then keeps the first $r + q$ components (expectations and hidden states) of the state vector but deletes (multiplication with zero) the last r ones. These are finally replaced by the observables y_τ^d, such that s_τ has the partitioning of equation 3. Note, that in contrast to the normalised recurrent neural network (eq. 1) the observables are added to the state vector after the nonlinearity. This is important for the consistency structure of the model.

In the future part of the unfolding ($\tau > t$) we replace the missing external inputs by an additional identity-block in the future consistency matrix $C_>$ which maps the first r components of the state vector, the expectations y_τ, to its last r components. So we get the desired partitioning of s_τ (eq. 3) and the model becomes dynamical consistent.

Figure 2 illustrates the corresponding network architecture. Note, that the nonlinearity and the final calculation of the state vector are separated and hence modelled in two different layers. This follows from the dynamical consistent state equation (eq. 2), in which the observables are added separately from the nonlinear component.

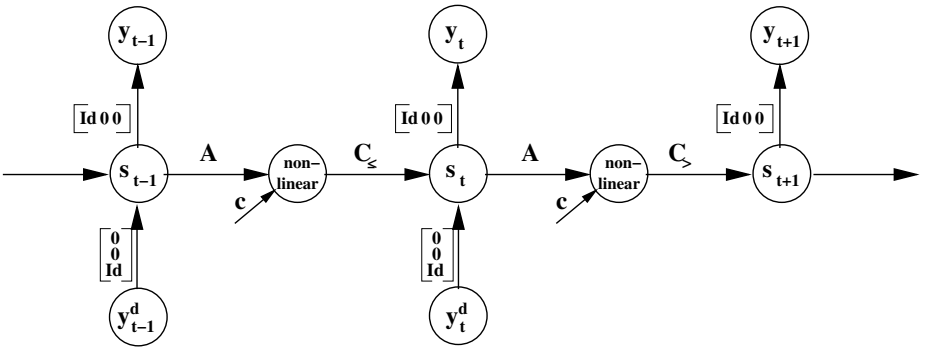


Fig. 2. Dynamical Consistent Neural Network (DCNN)

Note, that, in order to provide the network with sufficient memory and computational power, A has to be not only large but also sparse. Besides that, sparseness of the high-dimensional transition matrix is an essential condition for the performance of the backpropagation algorithm [6].

4 Monthly Forecasting and Trading of FX Rates with DCNN

In the following we present an application of DCNN to forecast the monthly development of the G4 FX rates (EURUSD, USDJPY, GBPUSD, EURJPY, EURGBP and GBPJPY) and the corresponding currency crosses.

Data and preprocessing. The data used for our model is depicted in table 1:

Table 1. List of observables (New York closing data)

Category	Input description
Foreign exchange rates	EURUSD foreign exchange rate USDJPY foreign exchange rate GBPUSD foreign exchange rate
10-year yield differences	10-year yield difference between U.S.A. and Germany 10-year yield difference between U.S.A. and Japan 10-year yield difference between G.Britain and U.S.A.
Relative strength of the main stock indices	Relative value of the Euro Stoxx 50 vs. the S&P 500 Relative value of the S&P 500 vs. the Nikkei 225 Relative value of the FTSE 100 vs. the S&P 500
Commodities	Brent Oil price in USD Gold price in USD

Our data selection is guided by the idea, that a major currency is driven by at least four influences [3], which are in particular the development of other major currencies, the 10-year bond yields, main stock indices in the respective countries and commodities. As a preprocessing we compute the logarithmic returns and, to make the data stationary, present the first differences as inputs to the neural network.

We divide the dataset, which ranges from 01.01.1999 to 12.08.2005, into a training set (01.01.1999 to 02.09.2004, 275 complete observations) and a generalisation set (03.09.2004 to 12.08.2005, 44 observations).

Model architecture and parameters. Our model uses an underlying weekly time grid. It is unfolded 16 weeks into the past and has an overshooting length of 6 weeks. To receive monthly forecasts we always add up four of the weekly outputs. The resulting sequence of forecasts is used for decision support. The internal state of the DCNN consists of 400 hidden neurons. By initialising the transition matrix A with a sparsity level of 12.5%, we enable the network to build up a memory of eight weeks [6].

Since all G4 currency crosses can be derived from the three main G4 FX rates, EURUSD, USDJPY and GBPUSD, we extend the DCNN architecture by an additional

output layer which is used to compute the FX-rates of all six currencies from the respective currency crosses. The output layer containing the price shifts is connected to this so-called interaction layer by a fixed handmade matrix.

Training procedure. We apply the LnCosh error function on the output and the interaction layer. Thus the network is trained to learn both the correct values of all the observables and the correct relation of the main currency crosses to each other, representing the remaining G4 FX rates EURJPY, GBPJPY and EURGBP. The applied training algorithm is pattern-by-pattern learning [7] with a learning rate of $\eta = 0.002$.

To desensitise the network from the unknown initial state we add adaptive noise, which we derive from the model's residuals, to the first hidden neurons in the unfolded network [6]. The uncertainty concerning the optimal sparsity structure is handled by averaging the forecasts of several DCNNs with randomly initialised weight matrices. As a result we obtain so-called monte carlo (MC) DCNN forecasts.

Measuring forecast performance. We use the error ratio as defined in equation 5 as a performance measure of our networks. Note, if the FX markets follow a random walk it should not be possible to systematically achieve values of the error ratio under one [8].

$$error\_ratio = \frac{\sum_{t=1}^{N} |y_t - y_t^d|}{\sum_{t=1}^{N} |y_t^d|} \tag{5}$$

In addition we compute the root mean squared error RMSE and the hitrate which counts how often the sign of the relative change of the FX market is correctly predicted. Our MC DCNN forecasts are benchmarked against a 3-layer-MLP with 10 hidden neurons which uses the monthly first and second differences of the variables in table 1 as inputs. The forecast performance of both network architectures are given in table 2.

Table 2. Root mean squared error, error ratio and hitrate for the 3-layer-MLP and the MC DCNN forecasts on the out of sample period

	3-layer-MLP			MC DCNN		
	RMSE	Error Ratio	Hitrate in %	RMSE	Error Ratio	Hitrate in %
EURGBP	0.0240	1.3725	52.57	0.0167	0.9593	54.55
EURJPY	0.0368	1.5984	61.36	0.0211	0.9193	65.91
EURUSD	0.0379	1.1562	50.00	0.0273	0.9928	52.27
GBPJPY	0.0298	1.3658	61.36	0.0214	0.9522	61.36
GBPUSD	0.0314	1.1580	47.73	0.0244	0.9273	52.27
USDJPY	0.0288	1.1889	52.27	0.0227	0.9673	63.64

As depicted in table 2, for all currency pairs the error ratio of our MC DCNN approach becomes smaller than one, which means that the MC DCNN outperforms a random walk. Also the hitrate is for all FX rates higher than 50%. In three cases the MC DCNN even reaches a hitrate of more than 60%. The results of the MLP are worse than the MC DCNN forecasts for all three performance ratios although the hitrates are still above 50% for five of the six currency pairs. The good performances in terms of

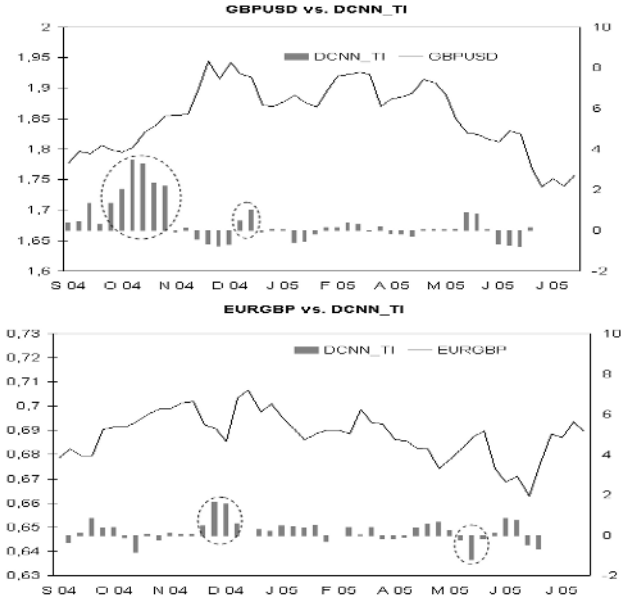


Fig. 3. GBPUSD and EURGBP FX rates vs. $DCNN\_TI$: As it can been seen in the marked areas, the $DCNN\_TI$ clearly indicates the right development of the respective market

forecast accuracy and hitrates underline that our MC DCNN forecasts are of high value in financial market modeling.

Introduction of a technical trading indicator based on DCNN. Based on our MC DCNN forecasts we want to support trading decisions. For such a decision support we have to combine our forecasts with intelligent financial evaluation criterions like return on investment, standard deviation of returns and the sharpe ratio as a risk adjusted measure of return.

In order to measure the confidence of our forecasts we compute a DCNN technical indicator, which uses, besides the average forecast $\bar{y}$, the ratio of the models agreeing with the direction of the mean forecast in the numerator and the standard deviation in the denominator:

$$DCNN\_TI = \frac{\frac{1}{N^2}\sum_{i=1}^{N} y_t \cdot 1_{\{sign(y_t)=sign(\bar{y}_i)\}}}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y}_i^d)^2}} \tag{6}$$

The $DCNN\_TI$ filters out the reliable from the less reliable forecasts. We expect the forecast reliability to rise when it takes on high absolute values, because in these cases the forecasts of more models agree with the direction of the mean forecast. The assumption is tested by looking at the development of the hitrate of the DCNN forecasts while changing the threshold value for the $DCNN\_TI$. The results and analysis are given in figure 3.

FX trading strategy based on the DCNN_TI. Now we use the $DCNN\_TI$ for the development of a trading strategy. Assuming a starting capital of 100.000 Euro, we specify the following trading rules based upon DCNN forecasts and the $DCNN\_TI$. Every week we take new long or short positions in the currencies as indicated by the $DCNN\_TI$. Each position is automatically closed after 4 weeks. The corresponding trading strategy is given by the following values:

$DCNN\_TI > 0$:	Take a long position in the respective FX rate.
$DCNN\_TI < 0$:	Take a short position in the respective FX rate.
Position Exits:	Automatic closure of this position after one month.

An essential issue when making trading decisions is position sizing, i.e., answering the question about how much capital to invest in each trade [9]. In conjunction with our market entries and exits we test three different position sizing techniques. Apart from the well established fixed units and fixed fraction techniques we propose a dynamic fraction position sizing technique controlled by the $DCNN\_TI$:

Fixed units:	Risk 1000 Euro in each trade.
Fixed fraction:	Risk 1 percent of capital in each trade.
$DCNN\_TI$ fraction:	Risk $DCNN\_TI$ percent of capital in each trade.

Thereby we calculate the risk of a currency by taking the Euro value of two historical standard deviations over the last twenty months of the relevant currency pair. The results for the three strategies are given in table 3.

Table 3. Annualised financial mean return μ, standard deviation σ, resulting sharpe ratio SR and investment degree ID for the three strategies on each FX rate and for the portfolio with all currencies. The strategies include transaction costs and slippage of 10 basis points (pips) per unit and interest rates in the respective countries. Since FX-spot trading allows for high leverage, the investment degrees of over 100% of the portfolios are realistic.

	Position sizing techniques								
	Fixed units			Fixed fraction			$DCNN\_TI$-fraction		
	$\frac{\mu}{\sigma}$ in %	SR	ID in %	$\frac{\mu}{\sigma}$ in %	SR	ID in %	$\frac{\mu}{\sigma}$ in %	SR	ID in %
EURGBP	0.55 / 2.03	0.27	22.56	0.65 / 2.16	0.30	23.84	1.14 / 1.86	0.61	14.14
EURJPY	4.36 / 2.48	1.76	26.15	4.50 / 2.65	1.70	27.66	3.27 / 2.82	1.16	27.96
EURUSD	0.18 / 1.76	0.10	14.29	0.03 / 1.86	0.02	14.97	2.01 / 2.00	1.00	16.81
GBPJPY	1.75 / 1.29	1.36	16.11	1.83 / 1.34	1.36	17.09	2.77 / 1.77	1.57	18.30
GBPUSD	0.22 / 0.82	0.27	9.34	0.17 / 0.89	0.19	9.92	2.30 / 0.85	2.69	8.88
USDJPY	4.18 / 2.46	1.70	27.92	4.49 / 2.61	1.72	29.70	3.51 / 2.65	1.32	26.76
PORTFOLIO	10.82 / 4.9	2.20	116.36	11.24 / 5.17	2.17	123.19	14.28 / 5.78	2.47	112.85

The $DCNN\_TI$-fraction strategy outperforms the fixed units and the fixed fractional strategies for several reasons. Apart from generating the highest sharpe ratio as a portfolio, the $DCNN\_TI$-fraction strategy yields stable results also when traded on single currencies since all individual sharpe ratios are above 0.60. The $DCNN\_TI$-fraction portfolio is also the one with the lowest average investment degree. This

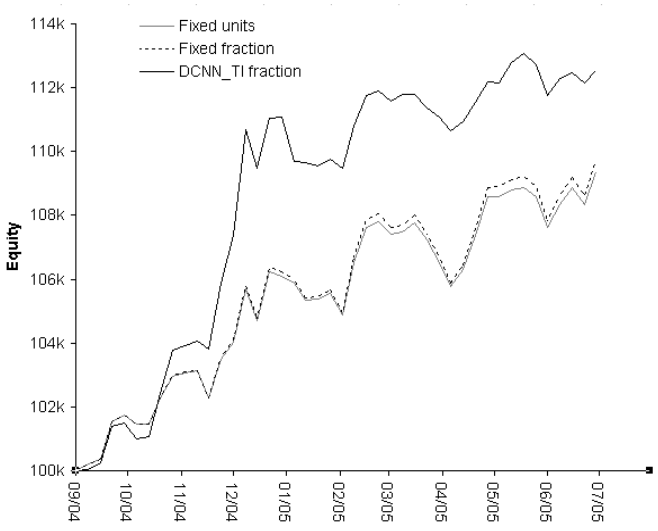


Fig. 4. Return on investment of the three trading strategies

theoretically allows to invest more capital in the strategy and generate higher returns. Due to the relatively short trading period the fixed fractional strategy can not produce strong exponential capital growth and therefore generates similar results as the fixed units strategy. Figure 4 shows the return on investment of the three trading strategies including transaction costs, slippage and interest rates.

5 Conclusion

In this paper we applied dynamical consistent neural networks for FX rate forecasting. We achieved very good out of sample results concerning forecasting accuracy and financial criterions. Furthermore we showed that our technical indicator based on the DCNN forecasts provides valuable decision support in currency trading and the development of trading strategies. Our results also generally underline the high performance quality of DCNN and the advantages of a coherent market modeling approach.

Further research is done on the analysis of optimal inputs, the modeling of a bigger universe of currencies and in depth investigation of the implementation of DCNN forecasts in trading.

Acknowledgment

Our computations were performed on our Neural Network modeling software SENN (*Simulation Environment for Neural Networks*), which is a product of Siemens AG.

References

1. Haykin, S.: Neural Networks: A Comprehensive Foundation. Macmillan, New York (1994)
2. Schaefer, A.M., Zimmermann, H.G.: Recurrent neural networks are universal approximators. In: Proceedings of the International Conference on Artificial Neural Networks (ICANN-06), Athens (2006)
3. Murphy, J.J.: Intermarket Technical Analysis. New York Institute of Finance, New York (1999)
4. Zimmermann, H.G., Neuneier, R.: Neural network architectures for the modeling of dynamical systems. In Kolen, J.F., Kremer, S., eds.: A Field Guide to Dynamical Recurrent Networks. IEEE Press (2001) 311–350
5. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In Rumelhart, D.E., et al., J.L.M., eds.: Parallel Distributed Processing: Explorations in The Microstructure of Cognition. Volume 1. MIT Press, Cambridge (1986) 318–362
6. Zimmermann, H.G., Grothmann, R., Schaefer, A.M., Tietz, C.: Identification and forecasting of large dynamical systems by dynamical consistent neural networks. In Haykin, S., J. Principe, T.S., McWhirter, J., eds.: New Directions in Statistical Signal Processing: From Systems to Brain. MIT Press (2006)
7. Neuneier, R., Zimmermann, H.G.: How to train neural networks. In Orr, G.B., Mueller, K.R., eds.: Neural Networks: Tricks of the Trade. Springer Verlag, Berlin (1998) 373–423
8. Jorion, P.: Financial Risk Manager Handbook. John Wiley & Sons (2003)
9. Wetzer, R.: Quantitative Trading Models - Quantitative Handelsmodelle. Herbert Utz Verlag, Munich (2004)

Testing the Random Walk Hypothesis with Neural Networks

Achilleas Zapranis

University of Macedonia, Accounting and Finance Department, 156 Egnatia St, 540 06 Thessaloniki, Greece
zapranis@uom.gr

Abstract. Although, there is an ongoing belief in the investment community that technical analysis can be used to infer the direction of future prices, the academic community always treated it (at best) with skepticism. However, if there is a degree of effectiveness in technical analysis, that necessarily lies in direct contrast with the efficient market hypothesis. In this paper, we use neural network estimators to infer from technical trading rules how to extrapolate future price movements. To the extend that the total return of a technical trading strategy can be regarded as a measure of predictability, technical analysis can be seen as a test of the independent increments version of random walk.

Keywords: neural networks, market efficiency, random walk, technical analysis, stock index, trading strategies.

1 Introduction

The efficient market hypothesis states that the current market price reflects the assimilation of all the information available. As a consequence, given the information, no prediction of the future price changes can be made. On the other hand, technical analysis, which is essentially the search for recurrent and predictable patterns in asset prices, attempts to forecast future price changes. Because it is based on public information, it should not generate excess profits if markets are operating efficiently. In particular, technical analysis can be used as a kind of "economic" test of the random walk version of independent but not identically distributed increments [2].

Thus, it is not surprising that contrary to the traditional dismissive attitude of the financial academics towards technical analysis, today there is a growing interest and a fast expanding empirical literature. For relevant studies see Neftci [4], Brock *et al* [1], Sullivan *et al* [6], Lo *et al* [2] among others. In this paper we use neural network estimators to infer from technical trading rules how to extrapolate future 5-day logarithmic returns. In section 2 we describe the data and in section 3, we discuss in detail the technical trading strategies considered in this analysis. In section 4, we present our results and finally in section 5, we conclude.

2 Index Data and Empirical Statistics

We considered the two most widely used US market stock indices (Dow Jones S&P 500), which are probably the most observed financial indicators in the world. For each

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 664 – 671, 2006.

index, from the available daily price data we generated 5-day non-overlapping logarithmic returns covering the period from 1 March 1990 to 2 November 2005. The full sample for each index contained 3,955 price observations and it was split into a training sample comprising 2,739 price observations and a test sample comprising 1,216 price observations. Each training sample covered the period from 1 March 1990 to 29 December 2000. Each test sample covered the period from 2 January 2001 to 2 November 2005.

The sample statistics of the 5-day continuously compounded returns for Dow Jones and S&P 500 are presented in Table 1. The means of the 5-day returns for the two indices are 0.17% and 0.16%, or 8.84% and 8.32% per year, correspondingly. The means of the 5-day returns in the training samples are considerably higher, i.e., 0.26% and 0.25%, or 13.52% and 13% per year, correspondingly. The means of the 5-day returns in the test samples are all negative, i.e., -0.01% band -0.03%, or -0.52% and -1.56% per year, correspondingly. The standard deviations of Dow Jones and S&P 500 are identical for the full sample, i.e., 2.20% or 15.52% per year. Both distributions show signs of skewness and heavy tails.

In Panel 1B, we can see that the estimated autocorrelations $\rho(i)$ at lag i in most cases are significantly different from zero. This implies that we can reject the hypothesis of uncorrelated increments (5-day returns), i.e., the weakest and the most often tested version of random walk in the empirical literature, for both samples. The other two versions of the random walk hypothesis, i.e., identically and independently distributed increments and independent increments are special cases of this version. To the extend that the total return of a technical trading strategy can be regarded as a measure of predictability, technical analysis can be seen as a test of the independent increments version of random walk hypothesis.

As we can see in Panel 1C, the correlation between the 5-day returns of the two indices is very high.

3 A Neural Network Based Technical Trading Strategy

3.1 A General Technical Forecasting Framework

Let I_t be the index closing price at day t. Define $r_t = \ln(I_t/I_{t-h})$ as the realized h-day logarithmic return at the end of day t. Let $\mathbf{x}_t = (x_{1,t}, x_{2,t}, \ldots, x_{m,t})$ be a vector of m technical indicators evaluated at the end of day t given the index close I_t. Also, suppose that r_t is generated by the function

$$r_{t+h} = \varphi\left(\mathbf{x}_t \mid I_t, I_{t-1}, ..., I_{t-k}\right) + \varepsilon_t \tag{1}$$

where, the term ε_t denotes random disturbances. If the function $\varphi(\circ)$ is sufficiently behaved then nonparametric neural network regression can be used to estimate $\varphi(\circ)$ consistently.

Let now $g_\lambda(\circ)$ be the output of a fully-connected one-hidden-layer neural network with λ hidden units and a bias term. For a given vector of technical indicators $\mathbf{x}$ we can estimate the h-day index return non-parametrically as follows:

$$\hat{r}_{t+h} = \ln\left(\hat{I}_{t+h}\right) - \ln\left(I_t\right) = g_\lambda\left(\mathbf{x}_t; \hat{\mathbf{w}}_n\right) \tag{2}$$

Table 1. Sample statistics for non-overlapping 5-day returns of Dow Jones and S&P 500

	DOW JONES			S&P 500		
Panel A	total	train	test	total	train	test
Mean	0.0017	0.0026	-0.0001	0.0016	0.0025	-0.0003
Std.	0.022	0.020	0.025	0.022	0.020	0.024
Skew.	-0.422	-0.358	-0.422	-0.313	-0.342	-0.197
	(0.039)	(0.047)	(0.070)	(0.039)	(0.047)	(0.070)
Kurt.	3.505	1.967	4.617	2.687	2.090	2.956
	(0.078)	(0.094)	(0.140)	(0.078)	(0.094)	(0.140)
Panel B						
$\rho(1)$	0.787*	0.791*	0.780*	0.780*	0.780*	0.779*
$\rho(2)$	0.573*	0.566*	0.578*	0.559*	0.550*	0.567*
$\rho(3)$	0.356*	0.342*	0.372*	0.330*	0.311*	0.353*
$\rho(4)$	0.146*	0.119*	0.180*	0.112*	0.081*	0.152*
$\rho(5)$	-0.067*	-0.093*	-0.034*	-0.107*	-0.148*	-0.052*
	(0.016)	(0.019)	(0.029)	(0.016)	(0.019)	(0.029)
Panel C						
0.93[a]/0.92[b]/0.96[c]						

In Panel A, "Total" refers to the full sample, "train" and "test" to the non-overlapping, consecutive training and test periods correspondingly. In Panel B, $\rho(i)$ is the estimated autocorrelation at lag i for each series and figures in parentheses are standard errors for the autocorrelation, $1 / \sqrt{N}$. In Panel C, ([a]) refers to the full sample, ([b]) refers to the training sample and ([c]) refers to the test sample.

where $\hat{\mathbf{w}}_n$ denotes the network parameter vector which was estimated from the training dataset D of size n. Note that the hat over I_{t+h} indicates that an estimate of the index level at $t + h$ can be extracted from the return forecast and not that we are attempting to forecast index levels.

Given the forecast of the h-day future return, $\hat{r}_{t+h}$, a signal, S_t, is generated at the end of day t (Buy, Sell, Neutral) according to the following rule:

if $\hat{r}_{t+h} > c$ then S_t = Buy

else if $\hat{r}_{t+h} < -c$ then S_t = Sell

else if $0 < \hat{r}_{t+h} < c$ or $-c < \hat{r}_{t+h} < 0$ then S_t = Neutral

where, c represents a threshold return corresponding to the breakeven transaction costs.

The composition of the invested portfolio at any point in time (denoted as C_t) can be either the index portfolio or cash equivalents (short-selling is not considered). Based on the composition of the portfolio C_t and the signal S_t at the end of day t, one of three actions At is performed: create index portfolio, liquidate index portfolio, no action. In Table 2 we can see the allowed actions (denoted as A_t) and the resulting new portfolio compositions, C_t.

The number of hidden units λ was estimated on the basis of "the minimum prediction risk principle". Prediction risk is the expected out-of-sample mean squared error and it is computed algebraically as in Refenes and Zapranis [5]. The model which minimized prediction risk was a single-hidden-layer network with $\lambda = 5$ hidden units. Throughout this paper the forecasting horizon is fixed to $h = 5$ days.

Table 2. Resulting portfolio compositions based on the combination of C_{t-h} and S_t.

C_{t-h}	Index Portfolio (I. P.)		T-Bills	
S_t	A_t	C_t	A_t	C_t
Buy	no action	$C_t = C_{t-h}$ (I. P.)	buy index portfolio	$C_t \neq C_{t-h}$ (I. P.)
Sell	liquidate index portfolio	$C_t \neq C_{t-h}$ (T-Bills)	no action	$C_t = C_{t-h}$ (T-Bills)
Neutral	no action	$C_t = C_{t-h}$ (I. P.)	no action	$C_t = C_{t-h}$ (T-Bills)

In the context of the simple framework we have just described, we are faced now with two problems. First, we must define the vector of technical indicators, $\mathbf{x}_t$, and second we must define the deterministic function $\varphi(\mathbf{x}_t|I_t)$. We tested out-of-sample two simple trading strategies. In the first one (TS_1) we set the return threshold to $c = 0$ and in the second one (TS_2) to $c = 40$ basis points. The benchmark strategy ($TS_{B\&H}$) was a buy-and-hold portfolio. For TS_2 there are three possible signals (Buy, Sell, Neutral) and the resulting portfolio compositions at the end of day t are given in Table 2. Note, that each time the portfolio composition changes from cash to stocks or from stocks to cash, the strategy incurs transaction costs 0.04% computed on the current portfolio valuation. That is, TS_2 returns are inclusive of transaction costs. For TS_1 there are only two possible signals (Buy, Sell) which are generated from the following rule:

```
if  r̂_{t+h} > 0  then S_t = Buy

else if  r̂_{t+h} < 0  then S_t = Sell
```

The resulting portfolio compositions at the end of day t are given in Table 2, for signals Buy and Sell. TS_1 returns do not account for transaction costs.

3.2 Determining the Relevant Indicators

Since, there are literally hundreds of technical indicators that can be used in the context of the aforementioned framework, we are faced with the problem of selecting the more *relevant*. However, relevance can only be determined in the context of a model. In order to select relevant technical indicators, we have used stepwise variable selection in the context of a multivariate linear regression model. The model parameters were estimated from the training samples. Then the statistical significant technical indicators at the 95% level were selected as the most relevant for the DJIA and S&P 500 indices. From a universe of 54 technical indicators (simple moving average, exponential moving average, projection oscillator, MACD, Qstick, TRIX, etc) 13 were found to be statistical significant for predicting the 5-day return of the DJIA, and 17 for the S&P 500. Only 4 technical indicators were found to be relevant to both indices.

4 Results

In Table 3 we can see the net returns for the out-of-sample period from 2 January 2001 to 2 November 2005.

Table 3. Overall returns for the out-of-sample period for DJIA and S&P 500

Strategy	TS_1	TS_2	$TS_{B\&H}$
DJIA	65.34 %	60.41 %	-1.62 %
S&P 500	43.80 %	36.71 %	-5.33 %

For the particular period the returns of the two indices are negative, with S&P 500 being the worst performer. In particular, the buy-and-hold strategy returns ($TS_{B\&H}$) are -1.62 percent for the DJIA and -5.33 percent for the S&P 500. At the same time, the returns of the strategies TS_1 and TS_2 are on average over 62 percent for the DJIA and over 40 percent for S&P 500. When trading costs of 40 basis points are taken into account (strategy TS_2) then the performance deteriorates slightly. However, the overall return for the out-of-sample period is still very satisfactory, i.e., 60.41 percent for the DJIA and 36.71 percent for the S&P 500, when at the same time the corresponding returns of the buy-and-hold strategies are both negative. As we will see in the following analysis the inclusion of a threshold return of $c = 40$ in TS_2 enhances the performance of the trading strategy, mainly because less signals are generated, but overall the trading costs offset any advantage.

In Table 4 we summarize standard test results for the out-of-sample period.Cumulative returns are reported for fixed 5-day periods after signals. “N(Buy)” and “N(Sell)” (columns 2 and 3) are the number of buy and sell signals generated from the trading strategy. Numbers in parentheses are standard t-statistics testing the difference between of the mean buy and sell from the unconditional 5-day mean, and

Table 4. Standard test results for the out-of-sample period for Dow Jones and S&P 500

Test	N(Buy)	N(Sell)	Buy	Sell	Buy > 0	Sell > 0	Buy – Sell
Panel A: Dow Jones							
c = 0	10	9	0.056	-0.051	0.700	0.444	0.108
			(15.509)	(-13.537)			(20.575)
c = 0.004	7	6	0.082	-0.080	0.714	0.166	0.162
			(19.025)	(-17.164)			(25.589)
Panel B: S&P 500							
c = 0	13	12	0.028	-0.029	0.692	0.333	0.057
			(8.858)	(-8.615)			(12.413)
c =0.004	12	11	0.039	-0.038	0.583	0.272	0.077
			(11.726)	(-10.924)			(16.076)

"N(Buy)" and "N(Sell)" are the number of buy and sell signals generated from the trading strategy. Numbers in parentheses are standard t-ratios testing the difference between of the mean buy and sell from the unconditional 5-day mean, and buy-sell from zero. "Buy > 0" and "Sell > 0" are the fraction of buy and sell returns greater than zero.

buy-sell from zero. "Buy > 0" and "Sell > 0" (columns 6 and 7) are the fraction of buy and sell returns greater than zero. "Buy-Sell" (column 8) is the difference between the mean "Buy" and "Sell" returns (columns 4 and 5). The *t*-statistics for the buys (sells) are as in Brock *et al*,

$$\frac{\mu_r - \mu}{\sqrt{\sigma^2/N + \sigma^2/N_r}} \tag{3}$$

where μ_r and μ_r are the mean return and number of signals for the buy and sell, and μ and N are the unconditional mean and number of observations. σ^2 is the estimated variance for the entire sample. For the buy-sell the t-statistic is,

$$\frac{\mu_b - \mu_s}{\sqrt{\sigma^2/N_b + \sigma^2/N_s}} \tag{4}$$

where μ_b and μ_b are the mean return and number of signals for the buys and μ_s and μ_s are the mean return and number of signals for the sells.

The first thing that we observe in Table 4 is that on average in more than 67 percent of the cases a buy signal corresponded to a positive return and in more than 30 percent of the cases a sell signal corresponded to a positive return.

Obviously we expect from a trading strategy that the generated buy signals correspond to positive returns more than 50 percent of the time, and the sell signals less than 50 percent of the time. In that respect both TS_1 and TS_2 performed very satisfactorily.

Furthermore, the number of generated buy and sell signals is on average for the DJIA 3.9 per year for $c = 0$ and 2.7 per year for $c = 40$ basis points. For the S&P 500 the corresponding figures are 5.2 per year for $c = 0$ and 4.8 per year for $c = 40$ basis points. These figures indicate that the strategies are not performing noise-trading. This is important, since the frequency of transactions can eliminate the profitability of most trading strategies. We also see that all the buy-sell differences are positive and the t-statistics for those differences are highly significant, rejecting the null hypothesis of equality with zero. Setting the threshold return to $c = 40$ basis points increased the spread between buy and sell returns.

We also observe that, the mean buy returns are all positive and statistically significant, i.e., we reject the null hypothesis that the returns equal the unconditional 5-day returns at the 5 percent significance level using a two-tailed test. In particular, for the DJIA the annualized returns are 2.912 percent for $c = 0$ (0.056 × 52) and 4.264 percent for $c = 40$ basis points. From Table 1, the unconditional 5-day return for the DJIA is -0.01 percent for the test period or -0.52 percent annualized. For the S&P 500 the annualized returns are 1.456 percent for $c = 0$ and 2.028 percent for $c = 40$ basis points. From Table 1, the unconditional 5-day return for the S&P 500 is -0.003 percent for the test period or -1.56 percent annualized. Overall, compared with the unconditional returns for the same period, the mean returns of both strategies are quite substantial.

Finally, the mean sell returns are also noteworthy. They are all negative, statistically significant and they are based on about 40% of all trading days.

5 Conclusions

In this paper we have used neural networks as a non-parametric platform for implementing simple technical trading strategies. Incorporating trading costs, a simple

trading strategy based on the 5-day return forecasts of a neural network returned a profit of 61.41% when applied to the Dow Jones, while in the same out-of-sample period the index dropped by -6.43%. A similar picture immerged for S&P 500. The neural network returned a profit of 36.71 percent, while the index dropped by -5.33 percent. Moreover, buy signals consistently generated higher returns than sell signals and the returns following sell signals were negative, which is not easily explained by any of the currently existing equilibrium models. Overall, the results of our study indicate that the nonparametric approach reveals the existence of a more complicated return generating mechanism than those suggested from studies using linear models.

Acknowledgments. The work presented in this paper was carried out as a part of the MAESTRO project (Configuration of Personalised Products for the Financial Sector Using Web-based and Optimisation Technologies. Integrated System Development). The MAESTRO project (SP NM-12) is partially funded by the General Secretariat for Research and Technology under the Community Support Framework III and the authors would like to acknowledge this support. The consortium partners are: Gnomon Informatics S.A.; Broker Systems S.A.; CERTH/ITI, University of Macedonia - Department of Accounting and Finance; Egnatia Securities S.A.; Eurolink Securities S.A.; Diolkos AEEX. I am also thankful to PhD candidate Efstratios Livanis for his valuable help in carrying out the simulations presented in this paper.

References

1. Brock, W., Lakonishok, J., and LeBaron, B: Simple technical rules and the stochastic properties of stock returns, The Journal of Finance, 47 (1992) 1731-1764
2. Campbell, J., Lo, A., MacKinlay, A.: The Econometrics of Financial Markets, Princeton, New Jersey, (1997)
3. Lo, A., Mamaysky, H., and Wang, J.: Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation, The Journal of Finance, 4 (2000) 1705-1765
4. Neftci, S., Policano, A.: Can chartists outperform the market? Market efficiency tests for "technical analysis", The Journal of Futures Markets, 4 (1984) 465-478
5. Refenes, A., Zapranis, A.: Neural model identification, variable selection and model adequacy, Journal of Forecasting, 18 (1999) 299-332
6. Sullivan, R., Timmermann, A., and White, H.: Data-snooping, technical trading rule performance, and the bootstrap, The Journal of Finance, 54 (1999) 1647-1691

Financial Application of Neural Networks: Two Case Studies in Greece

S. Kotsiantis[1], E. Koumanakos[2], D. Tzelepis[1], and V. Tampakas[1]

[1] Department of Accounting, Technological Educational Institute of Patras, Greece
[2] National Bank of Greece, Credit Division
sotos@math.upatras.gr, koumanak@upatras.gr, tzelepis@upatras.gr, tampakas@teipat.gr

Abstract. In the past few years, many researchers have used Artificial Neural Networks (ANNs) to analyze traditional classification and prediction problems in accounting and finance. This paper explores the efficacy of ANNs in detecting firms that issue fraudulent financial statements (FFS) and in predicting corporate bankruptcy. To this end, two experiments have been conducted using representative ANNs algorithms. During the first experiment, ANNs algorithms were trained using a data set of 164 fraud and non-fraud Greek firms in the recent period 2001-2002. During the second experiment, ANNs algorithms were trained using a data set of 150 failed and solvent Greek firms in the recent period 2003-2004. It was found that ANNs could enable experts to predict bankruptcies and fraudulent financial statements with satisfying accuracy.

Keywords: fraud detection, bankruptcy prediction.

1 Introduction

Neural networks are one of the most innovative analytical tools to surface in the financial arena. The availability of vast amounts of historical data in recent years, coupled with the enormous processing power of desktop computers, has enabled the use of automated systems to assist in complex decision making environments. The automated system examines financial ratios as predictors of performance, and assesses posterior probabilities of financial health. Neural Network Applications in the financial world include [6], [16]: currency prediction, futures prediction, bond ratings, debt risk assessment, credit approval and bank theft. The articles [9] and [16] review the literature on artificial neural networks (ANNs) applied to accounting and finance problems. Moreover, Vellido et al. [25] surveyed 123 articles from 1992 through 1998. They included 8 articles in accounting and auditing, and 44 articles in finance (23 on bankruptcy prediction, 11 on credit evaluation, and 10 in other areas).

Researchers have used various techniques and models to detect accounting fraud and predict corporate bankruptcy in circumstances in which, a priori, is likely to exist. However few studies have tested the predictive ability of different ANNs used by means of a common data set. In this study, we carry out an in-depth examination of publicly available data from the financial statements of various firms in order to (a) detect FFS and (b) predict corporate bankruptcy by using ANN learning methods.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 672 – 681, 2006.

The following section attempts a brief literature review of the techniques for detecting firms that issue fraudulent financial statements and describes the data set of our study. Section 3 attempts a brief literature review of the techniques for predicting corporate bankruptcy and describes the data set of our study. Section 4 presents the experimental results for the representative compared algorithms in our data sets. Finally, section 5 discusses the conclusions and some future research directions.

2 Literature Review for the Issue of FFS

Although it is not a new phenomenon, the number of corporate earnings restatements due to aggressive accounting practices, accounting irregularities, or accounting fraud has increased significantly during the past few years, and it has drawn much attention from investors, analysts, and regulators [23].

The financial statement audit is a monitoring mechanism that helps reduce information asymmetry and protect the interests of the principals, specifically, stockholders and potential stockholders, by providing reasonable assurance that management's financial statements are free from material misstatements. However, in real life, detecting management fraud is a difficult task when using normal audit procedures [2] since there is a shortage of knowledge concerning the characteristics of management fraud. Additionally, given its infrequency, most auditors lack the experience necessary to detect it. Moreover, managers deliberately try to deceive auditors [4].

Nieschwietz et al. [19] provide a comprehensive review of empirical studies related to external auditors' detection of fraudulent financial reporting while Albrecht et al. [2] review the fraud detection aspects of current auditing standards and the empirical research conducted on fraud detection. Ansah et al. [4] investigate the relative influence of the size of audit firms, auditor's position tenure and auditor's year of experience in auditing on the likelihood of detecting fraud in the stock and warehouse cycle. Green and Choi [15] developed a Neural Network fraud classification model. The model used five ratios and three accounts as input. The results showed that Neural Networks have significant capabilities when used as a fraud detection tool. Fanning and Cogger [13] also used a Neural Network to develop a fraud detection model. The input vector consisted of financial ratios and qualitative variables. They compared the performance of their model with linear and quadratic discriminant analysis, as well as logistic regression, and claimed that their model is more effective at 174 detecting fraud than standard statistical methods.

For Greek data, Spathis [24] constructed a model to detect falsified financial statements. He employed the statistical method of logistic regression. The reported accuracy rate exceeded 84%. Kirkos et al [17] investigate the usefulness of Decision Trees, Neural Networks and Bayesian Belief Networks in the identification of fraudulent financial statements. In terms of performance, the Bayesian Belief Network model achieved the best performance managing to correctly classify 90.3% of the validation sample in a 10-fold cross validation procedure. For both studies [17] and [24], 38 FFS firms were matched with 38 non-FFS firms.

The application of ANN techniques for financial classification is a fertile research area [9], [13], [15]. As a consequence, a main objective for this study is to evaluate the predictive ability of ANN techniques by conducting a number of experiments using representative learning algorithms.

Table 1. Research Variables description

Variables	Variable Description
RLTC/RCR02	Return on Long -term capital / Return on Capital and Reserves 2002
AR/TA 01	Accounts Receivable/Total Assets 2001
TL/TA02	Total liabilities/Total assets 2002
AR/TA02	Accounts Receivable/Total Assets 2002
WC/TA 02	Working capital/total assets 2002
DC/CA02	Deposits and cash/current assets 2002
NFA/TA	Net Fixed Assets/Total Assets
NDAP02	Number of days accounts payable 2002
LTD/TCR02	Long term debt/total capital and reserves 2002
S/TA02	Sales/total assets 2002
RCF/TA02	Results carried forward/total assets 2002
NDAR02	Number of days accounts receivable 2002
CAR/TA	Change Accounts Receivable/Total Assets
WCL02	Working capital leveraged 2002
ITURN02	Inventory turnover 2002
TA/CR02	Total Assets/Capital and Reserves 2002
EBIT/TA02	Earnings before interest and tax/total assets 2002
CFO02	Cash flows from operations 2002
CFO01	Cash flows from operations 2001
CR02	Current assets to current liabilities 2002
GOCF	Growth of Operational Cash Flow
CAR/NS	Change Accounts Receivable/Net Sales
EBT02/EBIT02	Earnings before tax 2002/Earnings before interest and tax 2002
Z-SCORE02	Altman z-score 2002
CR/TL02	Capital and Reserves/total liabilities 2002

2.1 Data Description

Our sample contained data from 164 Greek listed on the Athens Stock Exchange manufacturing firms (no financial companies were included). Auditors checked all the firms in the sample. For 41 of these firms, there was published indication or proof of involvement in issuing FFS. The classification of a financial statement as false was based on the following parameters: inclusion in the auditors' report of serious doubts as to the accuracy of the accounts, observations by the tax authorities regarding serious taxation intransigencies which significantly altered the company's annual balance sheet and income statement, the application of Greek legislation regarding negative net worth, the inclusion of the company in the Athens Stock Exchange categories of "under observation and "negotiation suspended" for reasons associated with the falsification of the company's financial data and, the existence of court proceedings pending with respect to FFS or serious taxation contraventions. The 41 FFS firms were matched with 123 non-FFS firms. All the variables used in the sample were extracted from formal financial statements, such as balance sheets and income statements. This implies that the usefulness of this study is not restricted by the fact that only Greek company data was used. The selection of variables to be used as candidates for participation in the input vector was based upon prior research work, linked to the topic of FFS [13], [15], [24]. Additional variables were added in an attempt to catch as

many as possible predictors not previously identified. Table 1 provides a brief description of the financial variables used in the present study.

3 Literature Review for Bankruptcy Prediction

The problem of Bankruptcy prediction is a classical one in the financial literature (see e.g. [1] for a review). The main impact of Bankruptcy prediction is in bank lending. Banks need to predict the possibility of default of a potential counterparty before they extend a loan. This can lead to sounder lending decisions, and therefore result in significant savings. O'Leary [20] analyzed 15 articles that applied ANNs to predict corporate failure or bankruptcy. For each study, he provided information about the data, the ANN model and software (means of development), the structure of the ANN (input, hidden and output layers) training and testing, and the alternative parametric methods used as a benchmark. He then analyzed the overall ability of the ANN models to perform the prediction task. The primary objectives of [8] were to develop failure prediction models for UK public industrial firms using a recent company sample, via logit analysis and the ANN methodology, and also to explore the incremental information content of operating cash flows in predicting the probability of business failure. NNs achieved the highest overall classification rates for all three years prior to insolvency, with an average classification rate of 78%. Zhang et al. [28] include in their paper a nice review of existing work on NN bankruptcy prediction. The majority of the NN approaches to default prediction use multilayer networks. Many other studies have been conducted for bankruptcy prediction using neural networks [3], [7], [27] and Support Vector Machines (SVM) [22].

Recently the performance of alternative non-parametric approaches has been explored in the Greek context to overcome the aforementioned shortcomings of the statistical and econometric techniques such as rough sets [11] and multicriteria discrimination method [12]. As we have already mentioned, in this paper we analyzed the performance of several ANN models on the problem of Bankruptcy prediction in the Greek context.

3.1 Data Description

Bankruptcy filings in the years 2003 and 2004 were provided directly from the National Bank of Greece directories and the business database of the financial information services company called ICAP, in Greece. Financial statement data for the fiscal years prior to bankruptcy were obtained from ICAP financial directories. The financial statements of these firms were collected for a period of three years. The critical year of failure denoted as year 0, three years before as year –3 and year –1 is the final year prior to bankruptcy filing. As the control sample, each selected bankrupt firm was matched with two non-bankrupt (healthy) firms of exactly the same industry, by carefully comparing the year of the reported data (year –1) assets size and the number of employees. The selected non-bankrupt corporations were within 20% of the selection criteria. Following the prior literature, we examine the probability of a firm's initial filing for bankruptcy and eliminate any observations for a firm after it has filed for bankruptcy during our sample period.

Table 2. Research Variables description

Category	Independent variables	Variable Description
Profitability Variables	OPIMAR	Operating income divided by net sales
	NIMAR	Net income divided by sales
	GIMAR	Gross income divided by sales
	ROE	Net income pre tax divided by Shareholder's equity capital
	ROCE	Net income pre tax divided by capital employed
Liquidity-Leverage Variables	EQ/CE	Shareholder's equity to capital employed
	CE/NFA	Capital employed to net fixed assets
	TD/EQ	Total debt to shareholder's equity capital
	CA/CL	Current assets to current liabilities
	QA/CL	Quick assets to current liabilities
	WC/TA	Working capital divided by total assets
Efficiency Variables	COLPER	Average collection period for receivables
	INVTURN	Average turnover period for inventories
	PAYPER	Average payment period to creditors
	S/EQ	Sales divided by Shareholder's equity capital
	S/CE	Sales divided by capital employed
	S/TA	Sales divided by Total Assets
Growth variables	GRTA	Growth rate of total assets $(TA_t - TA_{t-1})/(ABS(TA_t)+ ABS(TA_{t-1})$
	GRNI	Growth rate of net income
	GRNS	Growth rate of net sales
Size variable	SIZE	Size of firm is the ln(Total Assets/GDP price index)

Our final bankruptcy sample consists of 50 initial bankruptcies in the year period 2003-2004 and is similar in size but more complete and recent compared to previous studies. The final pooled sample of failed and solvent firms is composed of 150 individual firms with financial data for a three-year period, which attributes 450 firm-year observations. Table 2 provides a brief description of the financial variables used in the present study classified in 5 groups.

4 Experimental Results

WINNOW is the representative of perceptron-based algorithms in our study [18]. It classifies a new instance x into the second-class if $\sum_i x_i w_i > \theta$ and into the first class otherwise. It initializes its weights w_i and θ to 1 and then it accepts a new instance (x, y) applying the threshold rule to compute the predicted class y'. If $y' = 0$ and $y = 1$, then the weights are too low; so, for each feature such that $x_i = 1$, $w_i = w_i \cdot \alpha$, where α is a number greater than 1, called the *promotion parameter*. If $y' = 1$ and $y = 0$, then

the weights were too high; so, for each feature $x_i = 1$, it decreases the corresponding weight by setting $w_i = w_i \cdot \beta$, where $0<\beta<1$, called the *demotion parameter*. The vector, which is correct on all examples of the training set, is then used for predicting the labels on the test set.

Voted-perceptron [14] stores more information during training and then use this elaborate information to generate better predictions on the test data. The information it maintains during training is the list of all prediction vectors that were generated after each and every mistake. For each such vector, the algorithm counts the number of iterations the vector "survives" until the next mistake is made; they refer to this count as the "weight" of the prediction vector. To calculate a prediction it computes the binary prediction of each one of the prediction vectors and combines all these predictions by a weighted majority vote. The weights used are the survival times described above. This makes intuitive sense, as "good" prediction vectors tend to survive for a long time and thus have larger weight in the majority vote.

ANN depends upon three fundamental aspects, the input and activation functions of the unit, the network architecture and the weight on each of the input connections. Given that the first two aspects are fixed, the behavior of the ANN is defined by the current values of the weights. The weights of the net to be trained are initially set to random values, and then instances of the training set are repeatedly exposed to the net. The values for the input of an instance are placed on the input units and the output of the net is compared with the desired output for this instance. Then all the weights in the net are adjusted slightly in the direction that would bring the output values of the net closer to the values for the desired output. The most well-known and widely used learning algorithm to estimate the values of the weights is the Back Propagation (BP) algorithm [5].

RBF network [5] uses the k-means clustering algorithm to provide the basis functions and learns a logistic regression on top of that. Symmetric multivariate Gaussians are fit to the data from each cluster. It uses the given number of clusters per class. It standardizes all numeric attributes to zero mean and unit variance.

The SVM technique revolves around the notion of a 'margin' that separates two data classes. Maximizing the margin, and thereby creating the largest possible distance between the separating hyperplanes can reduce the upper bound on the expected generalization error [21]. However, most real-world problems involve non-separable data for which no hyperplane exists that successfully separates the positive from negative instances in the training set. The solution is then to map the data into a higher-dimensional space and define a separating hyperplane there. Sequential Minimal Optimization (or SMO) algorithm was the representative of the SVMs as one of the fastest methods to train SVMs [21].

All accuracy estimates were obtained by averaging the results from stratified 10-fold cross-validation in our datasets. It must be mentioned that we used the free available source code for our experiments in order to find the best parameters for each algorithm by the book [26]. The results for the first case study (fraud detection) are presented in Table 3. In Table 3, we also present the accuracy of Logistic Regression (LR) as benchmark algorithm.

Table 3. Accuracy of models in fraud detection (discretized)

	Winnow	BP	Voted Perceptron	RBF	SMO	LR
Total Acc.	75.32	82.91	81.01	77.85	79.11	75.3
Fraud (F)	56.1	56.1	39.0	34.1	39.0	36.6
Non-Fraud (NF)	82.1	92.3	95.7	93.2	93.2	88.9

The Winnow algorithm (with alpha: 2, beta: 0.5) correctly classifies 75.32% of the total sample, 56.1% of the fraud cases and 82.1% of the non-fraud cases. The RBF algorithm (with minimum standard deviation for the clusters: 2, number of clusters for K-Means to generate: 2) manages to correctly classify 77.85% of the total validation sample, 34.1% of the fraud cases and 93.2% of the non-fraud cases. Moreover, BP algorithm (with 1 hidden layer, learning rate: 0.3, momentum: 0.2) succeeds in correctly classifying 56.1% of the fraud cases, 92.3% of the non-fraud cases and 82.91% of the total validation sets. Furthermore, Voted Perceptron algorithm (with maximum number of alterations to the perceptron: 10.000) succeeds in correctly classifying 39% of the fraud cases, 95.7% of the non-fraud cases and 81.01% of the total validation sets. SMO algorithm (with exponent for the polynomial kernel: 1) correctly classifies 79.11% of the total sample, 39% of the fraud cases and 93.2% of the non-fraud cases.

Recently in the area of Machine Learning the concept of combining classifiers is proposed as a new direction for the improvement of the performance of individual classifiers [10]. For this reason, we combined the previous algorithms using the simple voting methodology [10]. Let us consider the voting step as a separate classification problem, whose input is the vector of the responses of the base classifiers. Simple voting uses a predetermined algorithm for this, namely to count the number of predictions for each class in the input and to predict the most frequently predicted class. The intuition is that the models generated using different learning biases are more likely to make errors in different ways.

The proposed voting ensemble of Winnow, BP, Voted Perceptron, SMO and RBF correctly classifies 91.2% of the total sample, 85.2% of the fraud cases and 93.3% of the non-fraud cases. In a comparative assessment of the models' performance we can conclude that the ensemble outperforms the simple models and achieve outstanding classification accuracy.

To facilitate the presentation and discussion of the results for the second case study (bankruptcy prediction), each year prior to financial distress is denoted as year –1, year –2, year –3, Year –1 refers to the first year prior to financial distress (e.g., for the firms that faced financial distress in 2004, year –1 refers to 2003); year –2 refers to the second year prior to financial distress (e.g., for the firms that faced financial distress in 2004, year –2 refers to 2002), etc.

In Table 4, there is the classification accuracy for each representative learning algorithm (with the previous referred parameters) for each examined year. We also present the accuracy of Logistic Regression (LR) as benchmark algorithm.

Table 4. Accuracy of the algorithms in bankruptcy prediction

		Winnow	BP	Voted Perceptron	SMO	RBF	LR
Year (-3)	Total Acc.	36.55	64.14	39.31	68.28	67.59	66.21
	Bankrupt	89.8	28.6	81.6	10.2	22.4	22.4
	Non- Bankrupt	9.4	82.3	17.7	97.9	90.6	88.5
Year (-2)	Total Acc.	42.07	69.65	64.83	69.66	71.72	67.59
	Bankrupt	69.4	22.4	2.0	10.2	26.5	22.6
	Non- Bankrupt	28.1	93.8	96.9	100	94.8	90.5
Year (-1)	Total Acc.	62.75	71.03	68.28	72.41	72.41	68.28
	Bankrupt	57.1	61.2	44.9	49.0	24.5	12.2
	Non- Bankrupt	65.6	76.0	80.2	84.4	96.9	96.9

It was found that learning algorithms could enable users to predict bankruptcies with satisfying accuracy long before the final bankruptcy. The experts are in the position to known 3 years before, which of the industries will bankrupt or not with sufficient precision, which reaches the 68% in the initial forecasts (3 years before the examined year) and exceeds the 72% the last year.

The proposed voting ensemble reaches the 71.72% (28.6% of Bankrupt firms and 93.8% of Non-Bankrupt) in the initial forecasts (3 years before the examined year) and exceeds the 73.79% (67.3% of Bankrupt firms and 77.1% of Non-Bankrupt) the last year. In a comparative assessment of the models' performance we can conclude that the ensemble outperforms the simple tested algorithms and achieve outstanding classification accuracy.

5 Conclusion

ANN based financial forecasting has been explored for about a decade. Many research papers are published on various international journals and conferences proceedings [7]. Some research results of financial forecasting found in references.

The aim of this study has been to compare the performance of ANNs techniques in detecting fraudulent financial statements and predicting corporate bankruptcy by using published financial data. According to our experiments, the attributes that mostly influence the induction in bankruptcy prediction are: WC/TA, EQ/CE and GRNI, while, the attributes that mostly influence the induction in detecting fraudulent financial statements are: RLTC/RCR02, AR/TA01, TL/TA02, AR/TA02, WC/TA02, DC/CA02, NFA/TA02, NDAP02. Finally, all the experimental results indicate that published financial statement data contains falsification indicators. In terms of performance, a voting ensemble achieved the best performance. It must be mentioned that our input vector solely consists of financial ratios. Enriching the input vector with qualitative information, such as previous auditors' qualifications or the composition of the administrative board, could increase the accuracy rate. The other open issue is

to consider macroeconomic indicators as inputs to the ANN. The prevailing economic conditions (as well as the current interest rates) can have a significant effect on the probability of bankruptcy.

Of course, all the techniques employed in the problem of predicting bankruptcy and FFS can be straight forwardly used in other financial classification problems such as bond rating or credit scoring.

Acknowledgments. The Project is Co-Funded by the European Social Fund & National Resources - EPEAEK II.

References

1. Altman, E.L.: Corporate Financial Distress and Bankruptcy. John Wiley and Sons (1993).
2. Albrecht, C.C., Albrecht, W.S. and Dunn, J.G.: Can auditors detect fraud: a review of the research evidence, Journal of Forensic Accounting, Vol. 2 No. 1 (2001) 1-12.
3. Anandarajan, M., Lee, P. and Anandarajan, A.: Bankruptcy Prediction of Financially Stressed Firms: An Examination of the Predictive Accuracy of Artificial Neural Networks, International Journal of Intelligent Systems in Accounting, Finance & Management, 10 (2001) 69–81.
4. Ansah, S.O., Moyes, G.D., Oyelere, P.B. and Hay, D.: An empirical analysis of the likelihood of detecting fraud in New Zealand, Managerial Auditing Journal, Vol. 17 No. 4, (2002) 192-204.
5. Bishop, C.: Neural Networks in Pattern Recognition. Oxford: Oxford University Press, (1995).
6. Calderon T.G., and Cheh J.J.: A roadmap for future neural networks research in auditing and risk assessment, International Journal of Accounting Information Systems, Vol. 3, No. 4 (2002) 203-236.
7. Charalambous, C., Charitou, A., & Kaourou, F.: Comparative analysis of artificial neural network models: Application in bankruptcy prediction. Annals of Operations Research, 99(1-4) (2000) 403-425.
8. Charitou, A., Neophytou, E., Charalambous, C.: Predicting Corporate Failure: Empirical Evidence for the UK, European Accounting Review, Vol.13 (2004) 465–497.
9. Coderre G. D.: Fraud Detection. Using Data Analysis Techniques to Detect Fraud. Global Audit Publications (1999).
10. Dietterich, T.G.: Ensemble methods in machine learning. In Kittler, J., Roli, F., eds.: Multiple Classifier Systems. LNCS Vol. 1857, Springer (2001) 1–15
11. Dimitras, A. I.; Slowinski, R.; Susmaga, R.; and Zopounidis, C.: Business failure prediction using rough sets. European Journal of Operational Research 114 (1999) 263-280.
12. Doumpos, M., Zopounidis, C.: A Multicriteria Discrimination Method for the Prediction of Financial Distress: The Case of Greece, Multinational Finance Journal, vol. 3, no. 2, (1999) 71–101.
13. Fanning K. and Cogger K.: Neural Network Detection of Management Fraud Using Published Financial Data, International Journal of Intelligent Systems in Accounting, Finance & Management, Vol. 7, No. 1 (1998) 21-24.
14. Freund Y., Schapire R.: Large Margin Classification Using the Perceptron Algorithm, Machine Learning 37 (1999) 277–296, Kluwer Academic Publishers.
15. Green B.P. and Choi J.H.: Assessing the risk of management fraud through neural network technology, Auditing: A Journal of Practice and Theory, Vol. 16(1), (1997) 14-28.

16. Coakley, J., Brown, C.: Artificial Neural Networks in Accounting and Finance: Modeling Issues, International Journal of Intelligent Systems in Accounting, Finance & Management, 9, (2000) 119–144
17. Kirkos S., Spathis C., Manolopoulos Y.: Detection of Fraudulent Financial Statements through the Use of Data Mining Techniques, Proceedings of the 2nd International Conference on Enterprise Systems and Accounting, Thessaloniki, Greece, (2005) 310-325.
18. Littlestone, N. and Warmuth M.: The weighted majority algorithm, Information and Computation, Vol. 108(2), (1994) 212–261.
19. Nieschwietz, R.J., Schultz, J.J. Jr and Zimbelman, M.F.: Empirical research on external auditors' detection of financial statement fraud, Journal of Accounting Literature, Vol. 19 (2000) 190-246.
20. O'Leary DE.: Using neural networks to predict corporate failure. International Journal of Intelligent Systems in Accounting, Finance and Management 7 (1998) 187–197.
21. Platt, J.: Using sparseness and analytic QP to speed training of support vector machines. In M. S. Kearns, S. A. Solla, & D. A. Cohn (Eds.), Advances in neural information processing systems 11. MA: MIT Press (1999).
22. Shin, K., Lee, T., Kim, H.: An application of support vector machines in bankruptcy prediction model, Expert Systems with Applications, Volume 28 (2005) 127-135.
23. Spathis C.: Detecting false financial statements using published data: some evidence from Greece, Managerial Auditing Journal, Vol. 17, No. 4 (2002) 179-191.
24. Spathis C., Doumpos M. and Zopounidis C.: Detecting falsified financial statements: a comparative study using multicriteria analysis and multivariate statistical techniques, The European Accounting Review, Vol.11, No. 3, (2002) 509-535.
25. Vellido A, Lisboa PJG, Vanghan J.: Neural networks in business: a survey of applications (1992– 1998). Expert Systems with Applications 17 (1999) 51–70.
26. Witten I. & Frank E., Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, San Mateo, (2000).
27. Xie, J., Wang, J. and Qiu, Z.: Effectiveness of Neural Networks for Prediction of Corporate Financial Distress in China, LNCS 3174 (2004) 994–999.
28. Zhang, G., Hu, M. Y., Patuwo, B. E., Indro, D. C.: Artificial Neural Networks in Bankruptcy Prediction: General Framework and Cross-Validation Analysis. European Journal of Operational Research 116 (1999) 16-32.

Credit Risk Analysis Using a Reliability-Based Neural Network Ensemble Model

Kin Keung Lai[1,2], Lean Yu[2,3], Shouyang Wang[1,3], and Ligang Zhou[2]

[1] College of Business Administration, Hunan University, Changsha 410082, China
[2] Department of Management Sciences, City University of Hong Kong,
Tat Chee Avenue, Kowloon, Hong Kong
{mskklai, msyulean, mszhoulg}@cityu.edu.hk
[3] Institute of Systems Science, Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, Beijing 100080, China
{yulean, sywang}@cityu.edu.hk

Abstract. Credit risk analysis is an important topic in the financial risk management. Due to recent financial crises and regulatory concern of Basel II, credit risk analysis has been the major focus of financial and banking industry. An accurate estimation of credit risk could be transformed into a more efficient use of economic capital. In this study, we try to use a triple-phase neural network ensemble technique to design a credit risk evaluation system to discriminate good creditors from bad ones. In this model, many diverse neural network models are first created. Then an uncorrelation maximization algorithm is used to select the appropriate ensemble members. Finally, a reliability-based method is used for neural network ensemble. For further illustration, a publicly credit dataset is used to test the effectiveness of the proposed neural ensemble model.

1 Introduction

In the financial risk management field, the credit risk analysis is beyond doubt an important topic. Especially for any credit-granting institution, such as commercial banks and certain retailers, the ability to discriminate good customers from bad ones is crucial. The need for reliable models that predict defaults accurately is imperative so that the interested parties can take either preventive or corrective action [1].

Due to its importance of credit risk analysis, there is a growing research stream about credit risk analysis. Accordingly, many different approaches including individual models, such as linear discriminant analysis [2], logit analysis [3], probit analysis [4], linear programming [5], integer programming [6], *k*-nearest neighbor (KNN) [7], classification tree [8], artificial neural networks (ANN) [9-10], genetic algorithm (GA) [11-12] and support vector machine (SVM) [13-14], and some hybrid models, such as neuro-fuzzy system [15-16] and fuzzy SVM [1],were widely applied to credit risk analysis tasks. Two recent surveys on credit scoring and credit modeling are [17-18].

In the above individual models [2-14], it is difficult to say that the performance of one model is consistently better than that of another model in all circumstances. In most situations, the performance of these individual models is problem-dependent. In

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 682 – 690, 2006.

the hybrid models [1, 15-16], some researchers have revealed that these hybrid classifiers which hybridize two or more classification methods can show higher correctness of classification than that of individual models. Motivated by this finding, we try to integrate multiple classifiers into an aggregated output to obtain the further performance improvement. In this study, ANN is selected as the basic learner to construct an ensemble classifier. The main reason of selecting ANN reflects the following two aspects. First of all, a neural network is often viewed as a "universal approximator" [19]. Usually, a three-layer back propagation neural network (BPNN) with an identity transfer function in the output unit and logistic functions in the middle-layer units can approximate any continuous function arbitrarily well given a sufficient amount of middle-layer units [19-20]. That is, neural networks have the ability to provide flexible mapping between inputs and outputs. Secondly, neural networks are far from being optimal classifier [21]. Many experimental results have shown the generalization of individual networks is not unique. Even for some simple problems, different neural networks with different settings (e.g., different network architecture and different initial conditions) may result in different generalization results. This characteristic makes neural networks have large improvement space in performance.

The motivations of this study are to propose a triple-phase neural network ensemble model for credit risk analysis and meantime to compare the performance with individual and hybrid credit analysis models. The rest of this study is organized as follows. Section 2 describes the building process of the proposed triple-phase neural network ensemble model in detail. For further illustration, two real credit datasets are used for testing in Section 3. Finally, some conclusions are drawn in Section 4.

2 The Building Process of Neural Network Ensemble Model

In this section, a triple-phase neural network ensemble model is proposed for credit risk analysis. First of all, multiple individual neural classifiers are generated. Secondly, an uncorrelation maximization algorithm is used to select the appropriate ensemble members. Finally, a reliability-based method is used for neural network ensemble for classification purpose. Particularly, a three-layer back propagation neural network (BPNN) with an identity transfer function in the output unit and logistic functions in the hidden layer units are used in this study [19-20].

2.1 Generating Diverse Individual Neural Network Classifiers

According to the principle of bias-variance trade-off [22], an ensemble model consisting of diverse models with much disagreement is more likely to have a good generalization performance. Therefore, how to generate the diverse model is a crucial factor. For neural network model, there are three methods for generating diverse models.

(1) Initializing different starting weights for each neural network models.
(2) Training neural network with different training subsets.
(3) Varying the architecture of neural network, e.g., changing the different numbers of layers or different numbers of nodes in each layer.

Usually, a neural network can usually be trained by the in-sample dataset and applied to out-of-sample dataset to verification. The model parameters (connection weights and node biases) will be adjusted iteratively by a process of minimizing the error function. Basically, the final output of the BPNN model can be represented as

$$y = f(x) = a_0 + \sum_{j=1}^{q} w_j \varphi(a_j + \sum_{i=1}^{p} w_{ij} x_i) \tag{1}$$

where x_i (i = 1, 2, …, p) represents the input patterns, y is the output, a_j (j = 0, 1, 2, …, q) is a bias on the jth unit, and w_{ij} (i = 1, 2, …, p; j = 1, 2, …, q) is the connection weight between layers of the model, $\varphi(\bullet)$ is the transfer function of the hidden layer, p is the number of input nodes and q is the number of hidden nodes.

In the classification problem, the neural network classifier can be represented by

$$F(x) = sign\left(a_0 + \sum_{j=1}^{q} w_j \varphi(a_j + \sum_{i=1}^{p} w_{ij} x_i) \right) \tag{2}$$

If $f(x)$ is larger than zero, then $F(x)$ belongs to positive class, representing by "+1"; otherwise $F(x)$ belongs to negative class, representing by "-1". In our study, we use neural network output value $f(x)$ as its credit classification score, instead of the classification results $F(x)$ directly. For credit risk classification problem, a credit analyst can adjust the parameter a_0 to modify the cutoff to change the percent of accepted. Only when the applicant's score is larger than the cutoff, his application will be accepted.

In addition, the neural network output value $f(x)$ is a good indicator of the confidence or reliability of ensemble classifiers. The larger the $f(x)$, the higher the reliability of neural network classifier for positive class (denoted by $g_i^+(x)$) is. Therefore the neural network output value $f(x)$ as a reliability measure is used to integrate the ensemble members, as further illustrated in Section 2.3.

2.2 Selecting Appropriate Ensemble Members

After training, each individual neural classifier has generated its own result. However, if there are a great number of individual members, we need to select a subset of representatives in order to improve ensemble efficiency. Furthermore, in the neural network ensemble model, it does not follow the rule of "the more, the better", as proposed by [23]. In this study, an uncorrelation maximization method is used to select the appropriate number of neural network ensemble members.

The basic starting point of the uncorrelation maximization algorithm is the principle of model diversity. That is, the correlations between the selected classifiers should be as small as possible, i.e., uncorrelation maximization. Supposed that there are p neural classifiers (C_1, C_2, …, C_p) with n forecast values. Then the error matrix (e_1, e_2, …, e_p) of p predictors is as

$$E = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1p} \\ e_{21} & e_{22} & \cdots & e_{2p} \\ \vdots & \vdots & & \vdots \\ e_{n1} & e_{n2} & \cdots & e_{np} \end{bmatrix}_{n \times p} \tag{3}$$

From the matrix, the mean, variance and covariance of E can be calculated as

$$\text{Mean: } \bar{e}_i = \frac{1}{n}\sum_{k=1}^{n} e_{ki} \quad (i = 1, 2, \ldots, p) \tag{4}$$

$$\text{Variance: } V_{ii} = \frac{1}{n}\sum_{k=1}^{n} (e_{ki} - \bar{e}_i)^2 \quad (i = 1, 2, \ldots, p) \tag{5}$$

$$\text{Covariance: } V_{ij} = \frac{1}{n}\sum_{k=1}^{n} (e_{ki} - \bar{e}_i)(e_{kj} - \bar{e}_j) \quad (i,j = 1, 2, \ldots, p) \tag{6}$$

Considering Equations (5) and (6), we can obtain a variance-covariance matrix:

$$V_{p \times p} = (V_{ij}) \tag{7}$$

Based upon the variance-covariance matrix, correlation matrix R can be calculated using the following equations:

$$R = (r_{ij}) \tag{8}$$

$$r_{ij} = \frac{V_{ij}}{\sqrt{V_{ii} V_{jj}}} \tag{9}$$

where r_{ij} is correlation coefficient, representing the degree of correlation classifier C_i and classifier C_j.

Subsequently, the plural-correlation coefficient $\rho_{Ci(C1,C2,\ldots,Ci\text{-}1,Ci+1,\ldots,\ Cp)}$ between classifier C_i and other p-1 classifiers can be computed based on the results of Equations (8) and (9). For convenience, $\rho_{Ci(C1,C2,\ldots,Ci\text{-}1,Ci+1,\ldots,\ Cp)}$ is abbreviated as ρ_i, representing the degree of correlation between C_i and (C_1, C_2, …, $C_{i\text{-}1}$, C_{i+1}, …, C_p). In order to calculate the plural-correlation coefficient, the correlation matrix R can be represented with block matrix, i.e.,

$$R \xrightarrow{\text{after transformation}} \begin{bmatrix} R_{-i} & r_i \\ r_i^T & 1 \end{bmatrix} \tag{10}$$

where R_{-i} denotes the deleted correlation matrix. It should be noted that $r_{ii} = 1$ (i = 1, 2, …, p). Then the plural-correlation coefficient can be calculated by

$$\rho_i^2 = r_i^T R_{-i}^T r_i \quad (i = 1, 2, \ldots, p) \tag{11}$$

For a pre-specified threshold θ, if $\rho_i^2 > \theta$, then the classifier C_i should be taken out from the p classifiers. On the contrary, the classifier C_i should be retained. Generally, the uncorrelation maximization algorithm can be summarized into the following steps:

(1) Computing the variance-covariance matrix V_{ij} and correlation matrix R with Equations (7) and (8);

(2) For the ith classifier (i =1, 2, …, p), the plural-correlation coefficient ρ_i can be calculated with the Equation (11);

(3) For a pre-specified threshold θ, if $\rho_i < \theta$, then the ith classifier should be deleted from the p classifiers. Conversely, if $\rho_i > \theta$, then the ith classifier should be retained.

(4) For the retained classifiers, we can also perform the procedure (1)-(3) iteratively until satisfactory results are obtained.

2.3 Fusing the Selected Members into an Aggregated Output

Depended upon the work done in previous two phases, a set of appropriate number of ensemble members can be collected. The subsequent task is to combine these selected members into an aggregated classifier in an appropriate ensemble strategy. Usually, majority voting is the most widely used fusion strategy for classification problems due to its easy implementation. It takes over half the ensemble to agree a result for it to be accepted as the final output of the ensemble regardless of the diversity and accuracy of each network's generalization. The main disadvantage of the majority voting is that it ignores the fact some neural network that lie in a minority sometimes do produce the correct results. In addition, at the stage of integration, it ignores the existence of diversity that is the motivation for ensembles [21].

In such situations, this study proposes a reliability-based ensemble strategy to fuse these ensemble members. According to Section 2.1, the reliability of positive class is $f_i(x)$. But note that the reliability measure falls into the interval $(-\infty, +\infty)$. The main drawback of this reliability measure is that ensemble classifier with large absolute value often dominate the final decision of the ensemble model. In order to overcome this shortcoming, we scale this measure into the unit interval [0, 1] with logistic function, i.e., $g_i^+(x)=1/(1+e^{-fi(x)})$ in the classification problem. Accordingly, the reliability of negative class can be represented as $g_i^-(x) = 1- g_i^+(x)$. In terms of the reliability of positive and negative class, the following five rules can be used to fuse the m individual ensemble members into an aggregated output:

(1) Maximum fusion rule:

$$F(x) = \begin{cases} 1, & \text{if } \max\limits_{i=1,\cdots,m} g_i^+(x) \geq \max\limits_{i=1,\cdots,m} g_i^-(x), \\ -1, & \text{otherwise.} \end{cases} \tag{12}$$

(2) Minimum fusion rule:

$$F(x) = \begin{cases} 1, & \text{if } \min\limits_{i=1,\cdots,m} g_i^+(x) \geq \min\limits_{i=1,\cdots,m} g_i^-(x), \\ -1, & \text{otherwise.} \end{cases} \tag{13}$$

(3) Median fusion rule:

$$F(x) = \begin{cases} 1, & \text{if } \underset{i=1,\cdots,m}{median}\left(g_i^+(x)\right) \geq \underset{i=1,\cdots,m}{median}\left(g_i^-(x)\right), \\ -1, & \text{otherwise.} \end{cases} \tag{14}$$

(4) Mean fusion rule:

$$F(x) = \begin{cases} 1, & \text{if } \sum_{i=1}^{m} g_i^+(x) \geq \sum_{i=1}^{m} g_i^-(x), \\ -1, & \text{otherwise.} \end{cases} \tag{15}$$

(5) Product fusion rule:

$$F(x) = \begin{cases} 1, & \text{if } \prod_{i=1,\cdots,m} g_i^+(x) \geq \prod_{i=1,\cdots,m} g_i^-(x), \\ -1, & \text{otherwise.} \end{cases} \tag{16}$$

3 Experiment Study

In this section, a publicly credit dataset is used to test the performance of the proposed reliability-based neural network ensemble model. For comparison purposes, three individual classification models: logit regression (LogR) [3], artificial neural network (ANN) [9-10] and support vector machine (SVM) [13-14], two hybrid classification models: neuro-fuzzy system [15-16] and fuzzy SVM [1] are also conducted.

The experimental dataset in this study is about Japanese credit card application approval obtained from UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/databases/credit-screening). For confidentiality all attribute names and values have been changed to meaningless symbols. After deleting the data with missing attribute values, we obtain 653 data, with 357 cases were granted credit and 296 cases were refused. To delete the burden of resolving multi-category, we use the 13 attributes A1-A5, A8-A15. Because we generally should substitute *k*-class attribute with *k*-1 binary attribute, which will greatly increase the dimensions of input space, we don't use two attributes: A6 and A7.

In this empirical analysis, we randomly draw 400 data from the 653 data as the initial training set, 100 data as the validation set and the else as the testing set. In order to increase model accuracy for credit risk evaluation, forty different neural network models with different initial weights are generated. Using the uncorrelation maximization algorithm, 18 diverse neural network classifiers are selected. For individual neural network models, a three-layer back-propagation neural network with 25 TANSIG neurons in the hidden layer and one PURELIN neuron in the output layer is used. The network training function is the TRAINLM. Besides, the learning rate and momentum rate is set to 0.1 and 0.15. The accepted average squared error is 0.05 and the training epochs are 2000. The above parameters are obtained by trial and error. In the SVM, the kernel function is Gaussian function with regularization parameter C = 50 and σ^2=5. Similarly, the above parameters are obtained by trial and error.

The classification accuracy in testing set is used as performance evaluation criterion. Typically, three evaluation criteria are used to measure the classification results.

$$\text{Type I accuracy} = \frac{\text{number of both observed bad and classified as bad}}{\text{number of observed bad}} \tag{17}$$

$$\text{Type II accuracy} = \frac{\text{number of both observed good and classified as good}}{\text{number of observed good}} \tag{18}$$

$$\text{Total accuracy} = \frac{\text{number of correct classification}}{\text{the number of evaluation sample}} \tag{19}$$

To reflect model robustness, each class of experiment is repeated 10 times and the final Type I, Type II and total accuracy is the average of the results of the 10 individual tests. According to the experiment design, the final results are presented in Table 1. Note that the results of two hybrid classification models are from the original literature [1, 15]. Because the results of type I and type II in [15] are not reported, the result of neuro-fuzzy system is kept to be blank in Table 1. Based on the similar reason, the standard deviations of the fuzzy SVM model are not shown in Table 1.

Table 1. Credit risk evaluation results with different approaches*

Category	Model	Rule	Type I (%)	Type II (%)	Total (%)
Single	LogR		74.58 [6.47]	76.36 [5.81]	75.82 [6.14]
	ANN		80.08 [7.23]	82.26 [6.25]	80.77 [6.86]
	SVM		78.41 [5.71]	81.43 [6.13]	79.91 [5.87]
Hybrid	Neuro-fuzzy [15]				77.91 [5.10]
	Fuzzy SVM [1]		82.70	85.43	83.94 [4.75]
Ensemble	Voting-based	Majority	84.37 [5.73]	86.58 [6.11]	85.22 [6.01]
	Reliability-based	Maximum	88.43 [4.34]	86.54 [5.25]	**87.24 [4.89]**
		Minimum	88.86 [4.41]	87.44 [4.74]	**88.08 [4.63]**
		Median	86.52 [4.96]	85.63 [5.03]	86.03 [4.99]
		Mean	86.17 [5.28]	87.85 [5.43]	**86.89 [5.35]**
		Product	85.75 [5.11]	86.46 [6.08]	85.96 [5.73]

* Standard deviations appear in brackets.

As can be seen from Table 1, we can find the following conclusions.

(1) Of the three single models, neural network model performs the best, followed by single SVM and logit regression. Using two tailed *t*-test, we find that the difference between performance of ANN and SVM is insignificant at five percent level of significance, while the difference between logit regression and ANN is significant at ten percent level of significance.

(2) In the two listed hybrid models, the neuro-fuzzy system performs worse than that of two single AI models, i.e., ANN and SVM. The main reason reflects the following aspects. First of all, the neurofuzzy system used the approximations of both the inputs as well as the output, as it fuzzified the inputs and defuzzified the output. Comparatively, the ANN and SVM did not use any such approximations. Secondly, the classification accuracies using neurofuzzy system is also influenced by the overlap in the way the range of values of a given attribute is split into its various categories (e.g., range of values for small, medium, and large). Again, these are pitfalls associated

with the mechanisms used for both fuzzification and defuzzification of input and output data, respectively [15]. However, the fuzzy SVM obtain good performance relative to single classification models. The main reason is that the fuzzy SVM can reduce the effect of outliers and yield higher classification rate than single SVM and ANN do.

(3) In the ensemble model, five reliability-based neural network ensemble models consistently outperform the majority voting based ensemble model, implying that the proposed reliability-based neural network ensemble model is a class of promising approach to handle credit risk analysis. Among the five reliability-based neural network ensemble models, the neural network ensemble model with minimum fusion rule perform the best, followed by maximization fusion rule and mean fusion rule. Although there is no significant difference in performance of the five reliability-based neural network ensemble models, the main reason resulting in such a small difference is still unknown, which is worth further exploring in the future.

(4) Generally speaking, the proposed reliability-based neural network ensemble model perform the best in terms of Type I accuracy, Type II accuracy, and total accuracy, revealing that the proposed reliability-based neural network ensemble learning technique is a feasible solution to improve the accuracy of credit risk evaluation.

4 Conclusions

In this study, a triple-phase neural network ensemble model is proposed to evaluate the credit risk problem. First of all, multiple individual neural classifiers are generated. Secondly, an uncorrelation maximization algorithm is used to select the appropriate ensemble members. Finally, a reliability-based method is used for neural network ensemble for classification purpose. Through the practical data experiment, we have obtained good classification results and meantime demonstrated that the proposed neural network ensemble model outperforms all single and hybrid models listed in this study. These results obtained reveal that the proposed triple-phase neural network ensemble technique can provide a promising solution to credit risk analysis.

Acknowledgements

This work is partially supported by National Natural Science Foundation of China, Chinese Academy of Sciences and Strategic Research Grant of City University of Hong Kong (SRG No. 7001806).

References

1. Wang, Y.Q., Wang, S. Y., Lai, K.K.: A New Fuzzy Support Vector Machine to Evaluate Credit Risk. IEEE Transactions on Fuzzy Systems 13 (2005) 820-831
2. Fisher, R.A.: The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics 7 (1936) 179-188
3. Wiginton, J.C.: A Note on the Comparison of Logit and Discriminant Models of Consumer Credit Behaviour. Journal of Financial Quantitative Analysis 15 (1980) 757-770

4. Grablowsky, B.J., Talley, W.K.: Probit and Discriminant Functions for Classifying Credit Applicants: A Comparison. Journal of Economic Business 33 (1981) 254-261
5. Glover, F.: Improved Linear Programming Models for Discriminant Analysis. Decision Science 21 (1990) 771-785
6. Mangasarian, O.L.: Linear and Nonlinear Separation of Patterns by Linear Programming. Operations Research 13 (1965) 444-452
7. Henley, W.E., Hand, D.J.: A *k*-NN Classifier for Assessing Consumer Credit Risk. Statistician 45 (1996) 77-95
8. Makowski, P.: Credit Scoring Branches out. Credit World 75 (1985) 30-37
9. Malhotra, R., Malhotra, D.K.: Evaluating Consumer Loans Using Neural Networks. Omega 31 (2003) 83-96
10. Smalz, R., Conrad, M.: Combining Evolution with Credit Apportionment: A New Learning Algorithm for Neural Nets. Neural Networks 7 (1994) 341-351
11. Chen, M.C., Huang, S.H.: Credit Scoring and Rejected Instances Reassigning through Evolutionary Computation Techniques. Expert Systems with Applications 24 (2003) 433-441
12. Varetto, F.: Genetic Algorithms Applications in the Analysis of Insolvency Risk. Journal of Banking and Finance 22 (1998) 1421-1439
13. Van Gestel, T., Baesens, B., Garcia, J., Van Dijcke, P.: A Support Vector Machine Approach to Credit Scoring. Bank en Financiewezen 2 (2003) 73-82
14. Huang, Z., Chen, H.C., Hsu, C.J., Chen, W.H., Wu, S.S.: Credit Rating Analysis with Support Vector Machines and Neural Networks: A Market Comparative Study. Decision Support Systems 37 (2004) 543-558
15. Piramuthu, S. Financial Credit-Risk Evaluation with Neural and Neurofuzzy Systems. European Journal of Operational Research 112 (1999) 310-321
16. Malhotra R., Malhotra, D.K.: Differentiating between Good Credits and Bad Credits Using Neuro-Fuzzy Systems. European Journal of Operational Research 136 (2002) 190-211
17. Thomas, L.C.: A Survey of Credit and Behavioral Scoring: Forecasting Financial Risk of Lending to Consumers. International Journal of Forecasting 16 (2002) 149-172
18. Thomas, L.C., Oliver, R.W., Hand D.J.: A Survey of the Issues in Consumer Credit Modelling Research. Journal of the Operational Research Society 56 (2005) 1006-1015
19. Hornik, K., Stinchocombe, M., White, H.: Multilayer Feedforward Networks are Universal Approximators. Neural Networks 2 (1989) 359-366
20. White, H.: Connectionist Nonparametric Regression: Multilayer Feedforward Networks Can Learn Arbitrary Mappings. Neural Networks 3 (1990) 535-549
21. Yang, S., Browne, A.: Neural Network Ensembles: combining multiple models for enhanced performance using a multistage approach. Expert Systems 21 (2004) 279-288
22. Yu, L., Lai, K.K., Wang, S.Y., Huang, W.: A Bias-Variance -Complexity Trade-off Framework for Complex System Modeling. Lecture Notes in Computer Science 3980 (2006) 518-527
23. Yu, L., Wang, S.Y., Lai, K.K.: A Novel Nonlinear Ensemble Forecasting Model Incorporating GLAR and ANN for Foreign Exchange Rates. Computers and Operations Research 32 (2005) 2523-2541

Competitive and Collaborative Mixtures of Experts for Financial Risk Analysis*

José Miguel Hernández-Lobato and Alberto Suárez

Computer Science Department
Escuela Politécnica Superior, Universidad Autónoma de Madrid
C/ Francisco Tomás y Valiente 11, Madrid 28049, Spain
josemiguel.hernandez@uam.es,
alberto.suarez@uam.es

Abstract. We compare the performance of competitive and collaborative strategies for mixtures of autoregressive experts with normal innovations for conditional risk analysis in financial time series. The prediction of the mixture of collaborating experts is an average of the outputs of the experts. If a competitive strategy is used the prediction is generated by a single expert. The expert that becomes activated is selected either deterministically (hard competition) or at random, with a certain probability (soft competition). The different strategies are compared in a sliding window experiment for the time series of log-returns of the Spanish stock index IBEX 35, which is preprocessed to account for the heteroskedasticity of the series. Experiments indicate that the best performance for risk analysis is obtained by mixtures with soft competition, where the experts have a probability of activation given by the output of a gating network of softmax units.

1 Introduction

During the last decades risk management has acquired great importance in financial institutions. Market risk analysis attempts to characterize the variations in the value of the investment portfolio of a financial institution associated with fluctuation in market conditions: asset prices, interest rates, exchange rates, volatilities, correlations and other risk factors. It is common to summarize the risk exposure by standard risk measures, such as Value-at-Risk (VaR) or Expected Shortfall (ES) [1]. To calculate the values of these measures, one assumes that the time series of returns of the institution's investment portfolio can be modeled as a stochastic process. The parameters of these models are usually determined by a fit to recent historic return data. For a given time horizon τ (for instance, one day) and a probability level p (usually high, e.g. 95% or 99%) the value of p-VaR is the negative $(1-p)$-quantile of the distribution of the returns for that time horizon. Because of its simplicity, Value-at-Risk has become the

* This work has been supported by the Spanish *Dirección General de Investigación*, project TIN2004-07676-C02-02. J. M. Hernández-Lobato acknowledges the support of *Universidad Autónoma de Madrid* under an FPU grant.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 691–700, 2006.

standard measure of market risk [1]. However, it is not a coherent measure of risk [2]. In particular, it is not subadditive, which means that it is possible to find two portfolios A and B that satisfy $VaR(A+B) > VaR(A)+VaR(B)$. This contradicts the intuitive notion that diversification should lead to a reduction of risk. An alternative to VaR that satisfies the subadditivity requirement and other desirable properties is Expected Shortfall (ES), which is also known in the literature as Expected Tail Loss (ETL) or conditional VaR (c-VaR). This quantity is defined as the average portfolio loss in a fixed time period τ, assuming that the loss exceeds the p-VaR.

To determine the value of VaR or Expected Shortfall it is necessary to model the distribution of portfolio returns. Empirical evidence shows that the distribution of daily returns is leptokurtic; that is, its tails are heavier than those predicted by a normal distribution [3]. A number of alternatives have been proposed to take into account heavier tails: mixtures of Gaussians [4], stable distributions [5], etc. The assumption of independence of returns is contradicted by the presence of correlations between consecutive returns [6] and a time-dependent structure in volatility [7,8,9]. Previous work by the authors [10,11] focused on autoregressive models based on mixtures of experts [12,13]. In those models autoregressive experts with soft competition are used to account for correlations and for the excess of kurtosis in the distribution of portfolio returns. The goal of the current research is to perform an exhaustive comparison among models using collaboration, soft competition or hard competition for conditional risk estimation. We carry out a detailed risk analysis of the different models based on different risk measures and advanced statistical tests.

2 Models for Time Series of Financial Prices

The future prices of a financial asset that is freely traded in an ideal market are unpredictable. By arguments of market efficiency any expectations on the future evolution of the asset value should be immediately reflected in the current price. Hence, the time series of asset prices follows a stochastic process, where the variations correspond to new unexpected information being incorporated into the market price. The time series of prices $\{S_t;\ 1 \leq t \leq T\}$ is generally not stationary. It is common to model the quasi-stationary series of log-returns that can be obtained by log-differencing the original series

$$X_t = \log \frac{S_t}{S_{t-1}},\ 1 \leq t \leq T. \tag{1}$$

Before formulating a model based on mixtures of autoregressive experts for the series of returns, we perform a transformation to take into account its heteroskedastic structure. GARCH(1,1) processes [9,15] are among the most successful models for describing the time-dependent structure of the volatility in financial time series. If the time series $\{X_t, 1 \leq t \leq T\}$ with mean μ follows a GARCH(1,1) model, then

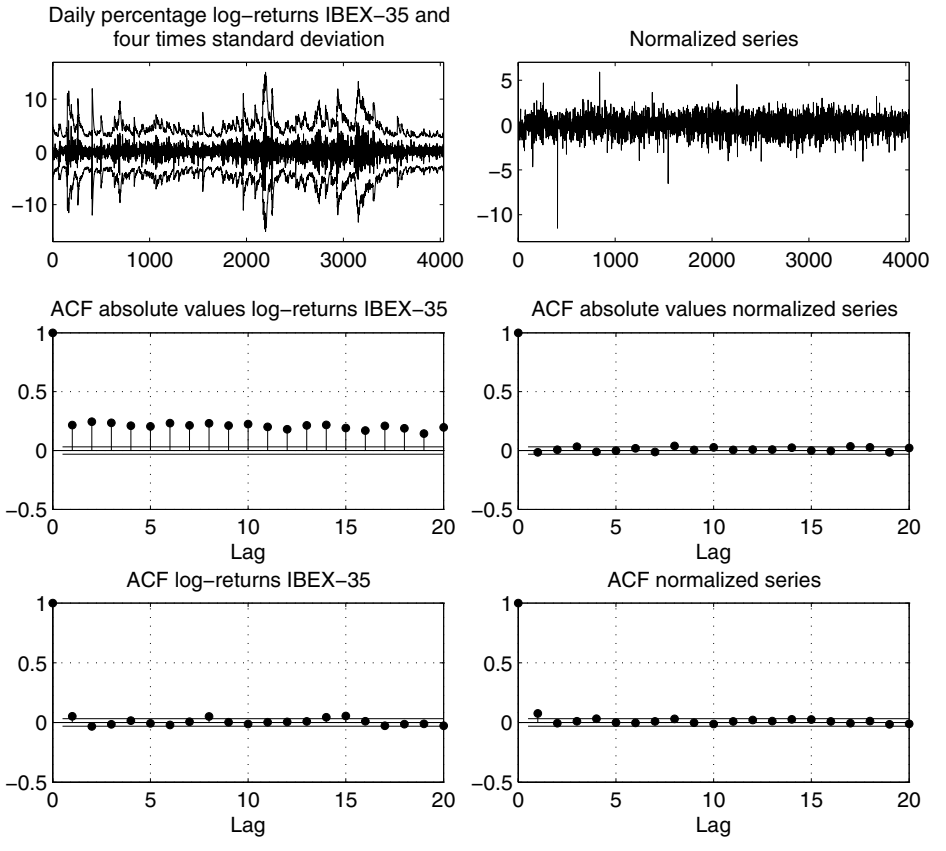


Fig. 1. Daily log-returns (multiplied by 100) of the Spanish IBEX 35 stock index from 12/29/1989 to 1/31/2006 (4034 values) provided by [14]. Graphs on the left column correspond to the original series of log-returns (first row) and to the sample autocorrelation functions (ACF) of the absolute values of the log-returns (second row) and of the log-returns themselves (third row). The outer lines in the top left plot correspond to $\mu \pm 4\sigma_t$, where μ and σ_t are obtained from a fit to a GARCH(1,1) model (2). The right column displays the corresponding plots for the homoskedastic series obtained after normalization (3).

$$\begin{aligned} X_t &= \mu + \sigma_t \varepsilon_t \\ \sigma_t^2 &= \gamma + \alpha (X_{t-1} - \mu)^2 + \beta \sigma_{t-1}^2 \,, \end{aligned} \tag{2}$$

where $\{\varepsilon_t,\ 1 \leq t \leq T\}$ are iidrv's generated by a $N(0,1)$ distribution and the parameters γ, α and β satisfy the constraints $\gamma > 0$, $\alpha, \beta \geq 0$ and $\alpha + \beta < 1$. Assuming that the time series of returns approximately follows a GARCH(1,1) process, it is then possible to obtain a homoskedastic time series $\{Z_t,\ 1 \leq t \leq T\}$ by performing the transformation

$$Z_t = \frac{X_t - \mu}{\sigma_t} \,, 1 \leq t \leq T, \tag{3}$$

where σ_t follows equation (2). The parameters μ, γ, α and β are estimated by maximizing the likelihood of the GARCH(1,1). Note that the GARCH process is not being trained in an optimal way. In particular the residuals $\{\varepsilon_t, 1 \leq t \leq T\}$ are not independent and their distribution is leptokurtic. Nonetheless, given that the deviations from independence and normality are small, the variance σ_t^2 estimated under the hypothesis of normal independent residuals should be a good approximation to the actual variance of the process.

On the left column, Fig. 1 displays the graph and autocorrelations of the time series of log-returns of the Spanish stock index IBEX 35 and, on the right column, the corresponding plots for the normalized time series (3). The features of this series are representative of typical time series of financial portfolio returns. The presence of medium-term correlations for the absolute values of the log-returns ($\{X_t,\ 1 \leq t \leq T\}$) is a clear mark of heteroskedasticity. These autocorrelations are not present in the normalized series, which appears to be homoskedastic. If we focus on the sample autocorrelations of the normalized log-returns, it is apparent that they still exhibit small but non-negligible short-term correlations, which can be modeled by an autoregressive process [15].

3 Mixtures of Autoregressive Experts

In this work we propose to model the series of normalized log-returns by a mixture of M autoregressive processes with a single delay, $AR(1)$, in a single level [12]. These types of models can be thought of as dynamical extensions of the mixture of Gaussians paradigm, which has been successfully applied to modeling the excess of kurtosis in the unconditional distribution of log-returns [4]. Our goal is to give an accurate and robust description of the conditional distribution of log-returns that can be used for risk analysis. For this reason we evaluate the performance of the mixture models based not only on point predictions, but also on their capacity to model the whole distribution of returns, especially of extreme events, which are determinant for the risk profile of a portfolio.

The way in which the outputs of the $AR(1)$ models are combined to generate a prediction is controlled by a gating network [13] with a single layer. The input for this network is the same as the input for the experts (i.e., the delayed value of the normalized series, Z_{t-1}). The output layer contains as many nodes as the number of experts in the mixture. Their activation is modulated by a *softmax* [16] function so that the outputs are within the interval $[0, 1]$ and add up to 1. Because of these properties they can be interpreted either as activation probabilities or as weights. In particular, if $\zeta_0^{(m)}$ and $\zeta_1^{(m)}$ are the weights of the m-th node of the gating network, its outgoing signal is

$$g_m(Z_{t-1}) = \frac{exp(\zeta_0^{(m)} + \zeta_1^{(m)} Z_{t-1})}{\sum_j exp(\zeta_0^{(j)} + \zeta_1^{(j)} Z_{t-1})}, m = 1, 2, \ldots, M. \tag{4}$$

Let $\phi_0^{(m)}$, $\phi_1^{(m)}$ and σ_m be the parameters of the m-th $AR(1)$ expert. There are different strategies to determine how the outputs of the experts in the mixture

are combined. We consider and compare three different paradigms: collaboration, hard competition and soft competition.

Collaboration: The output of the mixture is a weighted average of the outputs from each of the experts. These weights are determined by the output of the gating network. The output of the mixture is

$$Z_t = \sum_{m=1}^{M} g_m(Z_{t-1}) \left[\phi_0^{(m)} + \phi_1^{(m)} Z_{t-1} + \sigma_m \varepsilon_t \right], \tag{5}$$

Hard competition: Experts compete, so that only one expert is active at a given time. The output of the gating network is either 1 for m_t^*, the expert that generates the output, or 0 for the other experts. This strategy was also proposed in [17]

$$Z_t = \phi_0^{(m_t^*)} + \phi_1^{(m_t^*)} Z_{t-1} + \sigma_{m_t^*} \varepsilon_t. \tag{6}$$

Soft competition: The output is generated by a single expert. However, in contrast to hard competition, every expert has a probability of being chosen to generate the output of the system. This probability is given by the output of the gating network

$$Z_t = \sum_{m=1}^{M} \xi_t^{(m)} \left[\phi_0^{(m)} + \phi_1^{(m)} Z_{t-1} + \sigma_m \varepsilon_t \right], \tag{7}$$

where the random variables $\{\xi_t^{(m)},\ m = 1, 2, \ldots, M\}$ take value one with probabilities $\{g_m(Z_{t-1}),\ m = 1, 2, \ldots, M\}$, respectively. At a given time t only one of them can have value 1, and the rest are zero. This strategy was proposed in [12] and used in [10,11].

In all these models, we assume that $\{\varepsilon_t\} \sim \text{IID } N(0, 1)$ and require that $\{|\phi_1^{(m)}| < 1;\ m = 1, 2, \ldots, M\}$ to guarantee stationarity.

In order to fit the parameters of the AR(1) processes and of the gating networks to the time series $\{Z_t,\ 1 \le t \le T\}$ we condition the distribution of Z_t to Z_{t-1} and maximize the likelihood function. The expressions of the conditional likelihood for collaborative (CL), hard competitive (HC) and soft competitive (SC) mixtures of M experts, given a series of observations $\{Z_t,\ 2 \le t \le T\}$ and an initial value Z_1 are, respectively,

$$\mathcal{L}_{CL}(\Theta; \{Z_t\} \mid Z_1) = \prod_{t=2}^{T} \psi \left(Z_t, \sum_{m=1}^{M} g_m(Z_{t-1}) AR_m(Z_{t-1}), \sqrt{\sum_{m=1}^{M} g_m^2(Z_{t-1}) \sigma_m^2} \right)$$

$$\mathcal{L}_{HC}(\Theta; \{Z_t\} \mid Z_1) = \prod_{t=2}^{T} \prod_{m=1}^{M} \psi \left(Z_t,\ AR_m(Z_{t-1}),\ \sigma_m \right)^{g_m(Z_{t-1})}$$

$$\mathcal{L}_{SC}(\Theta; \{Z_t\} \mid Z_1) = \prod_{t=2}^{T} \sum_{m=1}^{M} g_m(Z_{t-1}) \psi \left(Z_t,\ AR_m(Z_{t-1}),\ \sigma_m \right), \tag{8}$$

where $\psi(x, \mu, \sigma)$ is the normal probability density function with mean μ and standard deviation σ evaluated at x, $\Theta = \{\phi_0^{(m)}, \phi_1^{(m)}, \sigma_m, \zeta_0^{(m)}, \zeta_1^{(m)},\ m = 1, 2, \ldots, M\}$ are the parameters that determine the model and $AR_m(Z_{t-1}) = \phi_0^{(m)} + \phi_1^{(m)} Z_{t-1}$. The previous expressions are maximized by a gradient-descent optimization algorithm to the log-likelihood, taking into account the restrictions of the AR parameters. We also restrict the parameters of the gating network to be in the interval [-50,50] in order to avoid floating point overflows in the calculation of the softmax function values. The optimization routine *fmincon* from the Matlab Optimization Toolbox [18] is used.

One well-known problem in the maximization of the observed likelihood in mixture models is that there is no global maximum [19]. Expert m can get anchored to a single data point in the sample if $\phi_0^{(m)} + \phi_1^{(m)} Z_{t-1} = Z_t$, $\sigma_m \to 0$ and $g_m(Z_{t-1}) > 0$, which causes a divergence in the likelihood function. To address this problem we adopt the solution proposed in [19] and modify the a priori probabilities of the variances of each expert in order to avoid that their values get too close to zero. The prior information is equivalent to a direct observation of T points known to have been generated by each expert and with sample variance $\hat{\sigma}^2$. Accordingly, the logarithmic conditional likelihood of each mixture of AR processes is modified and includes a term of the form

$$\sum_{m=1}^{M} -\frac{T}{2} log(\sigma_m^2) - \frac{T\hat{\sigma}^2}{2\sigma_m^2}\,. \tag{9}$$

In our experiments the values $T = 0.1$ and $\hat{\sigma}^2 = 1.5$ are used. The results are not very sensitive to reasonable choices of these parameters.

4 Testing the Models

To test how accurately the different mixtures of experts fit the data $\{Z_t,\ 1 \leq t \leq T\}$, we transform each point from the series to its percentile in terms of the conditional distribution specified by the mixture of experts (ME) and then apply the inverse of the standard normal cumulative distribution function [20].

$$Y_t = \Psi^{-1}[cdf_{ME}(Z_t | Z_{t-1})], \tag{10}$$

where $\Psi^{-1}(u)$ is the inverse of the cumulative distribution function for the standard normal. This transformation is monotonic and preserves the rank order of the normalized log-returns (i.e. the tails of the distribution in Z_t are mapped into the tails of the distribution in Y_t. If the hypothesis that the values $\{Z_t,\ 2 \leq t \leq T\}$, given Z_1, have been generated by the model considered is correct, then the transformed values $\{Y_t, 2 \leq t \leq T\}$ should be distributed as a standard normal random variable. In consequence, it is possible to apply statistical tests for normality to these transformed values, to determine whether the original hypothesis should be rejected.

The cumulative distribution functions of the collaborative (CL) and competitive (CP) mixtures evaluated on Z_t and conditioned to Z_{t-1} are

$$cdf_{CL}(Z_t \mid Z_{t-1}) = \Psi\left(Z_t,\ \sum_{m=1}^{M} g_m(Z_{t-1}) AR_m(Z_{t-1}),\ \sqrt{\sum_{m=1}^{M} g_m^2(Z_{t-1})\sigma_m^2}\right)$$

$$cdf_{CP}(Z_t \mid Z_{t-1}) = \sum_{m=1}^{M} g_m(Z_{t-1}) \Psi\left(Z_t,\ AR_m(Z_{t-1}),\ \sigma_m\right), \tag{11}$$

respectively, where $\Psi(x, \mu, \sigma)$ is the normal cumulative distribution function with mean μ and standard deviation σ evaluated at x.

4.1 Statistical Tests for Risk Analysis

In order to assess and compare the performance of the models we have implemented three statistical tests described in [21]. These tests are especially designed to evaluate the quality of risk estimation models. The risk measures considered can be characterized by a functional, $\phi : \mathcal{Q} \to \mathbb{R}$, of the cumulative distribution function $\mathcal{Q}$. The data used in the tests are the normalized log-returns transformed to follow a $N(0, 1)$ distribution (10). Hence, in our experiments $\mathcal{Q}$ is a standard normal. The functionals for VaR, for expected shortfall and for exceedances over a level V are

$$\begin{aligned}
\phi_{VaR}(\mathcal{Q}) &= -\mathcal{Q}^{-1}(1-p) \\
\phi_{ES}(\mathcal{Q}) &= -\frac{1}{1-p}\int_{-\infty}^{\mathcal{Q}^{-1}(1-p)} y\, d\mathcal{Q}(y) \\
\phi_{Exc}(\mathcal{Q}) &= \int_{-\infty}^{\infty}\left(\sum_{t=1}^{T} I_{(-\infty,V]}(y)\right) d\mathcal{Q}(y)\,,
\end{aligned} \tag{12}$$

where p is the probability used to calculate VaR. The last functional corresponds to the average number of elements in a sample $\{Y_t,\ 1 \le t \le T, Y_t \sim \mathcal{Q}\}$ that are smaller than the constant value V. If $V = \mathcal{Q}^{-1}(1-p)$ this functional implements the binomial test for exceedances over VaR [22]. Provided that ϕ is Hadamard differentiable on $\mathcal{Q}$ and $\mathcal{Q}_T = T^{-1}\sum \delta_{Y_t}$ is the empirical distribution of a sample $\{Y_t,\ 1 \le t \le T, Y_t \sim \mathcal{Q}\}$, we can apply the functional delta method [23] so that

$$\sqrt{T}(\phi(\mathcal{Q}_T) - \phi(\mathcal{Q})) \approx \phi'_{\mathcal{Q}}(\sqrt{T}(\mathcal{Q}_T - \mathcal{Q})) \approx \sqrt{T}\frac{1}{T}\sum_{t=1}^{T}\phi'_{\mathcal{Q}}(\delta_{Y_t} - \mathcal{Q})\,, \tag{13}$$

where the function $y \to \phi'_{\mathcal{Q}}(\delta_y - \mathcal{Q})$ is the influence function of the functional ϕ. This influence functional measures the change in $\phi(\mathcal{Q})$ if an infinitesimally small part of $\mathcal{Q}$ is replaced by a point probability mass at y. In the last step of the previous expression we have made use of the linearity of the influence function. The quantity $\phi(\mathcal{Q}_T) - \phi(\mathcal{Q})$ behaves as an average of independent

random variables $\phi'_{\mathcal{Q}}(\delta_{Y_t} - \mathcal{Q})$ which are known to have zero mean and finite second moments. Therefore, the central limit theorem states that $\sqrt{T}(\phi(\mathcal{Q}_T) - \phi(\mathcal{Q}))$ has a normal limit distribution with mean 0 and variance $\mathbb{E}_t[(\phi'_{\mathcal{Q}}(\delta_t - \mathcal{Q}))^2]$. We can then use the statistic

$$S_T = \frac{\sqrt{T}(\phi(\mathcal{Q}_T) - \phi(\mathcal{Q}))}{\sqrt{\mathbb{E}_y[(\phi'_{\mathcal{Q}}(\delta_y - \mathcal{Q}))^2]}} \xrightarrow{d} \mathcal{N}(0,1) \tag{14}$$

to determine how well $\mathcal{Q}_T$ approximates $\mathcal{Q}$ according to the functional ϕ. The expression of the variances for the three proposed functionals are given in [21].

5 Experiments and Results

We assess the accuracy of the models investigated by means of a sliding window analysis of the series of IBEX 35 log-returns (see Fig. 1). Each of the models is trained on a window containing 1000 values and then tested on the first out-of-sample point. The origin of the sliding window is then displaced by one point and the training and evaluation processes repeated. To avoid getting trapped in local maxima of the likelihood, we restart the optimization process several times at different initial points selected at random and retain the best solution. Every 50 iterations in the sliding window analysis, we perform an exhaustive search by restarting the optimization process 2000 times for the HC models, 500 times for the CL and SC and 5 times for the GARCH(1,1) process. In the remaining 49 iterations we use the solution from the previous iteration as an initial value for a single optimization. Once an exhaustive optimization process is completed we restart the previous 50 optimizations (49 simple and 1 exhaustive) using the new solution found as the initial point and replace the older fits if the values of the likelihood are improved.

Table 1 displays the results of the statistical tests performed and the mean square prediction error (MSE) of each model. All the models investigated perform well in the prediction of VaR and exceedances over VaR at a probability level of 95%. At a probability level of 99% the only models that cannot be rejected are mixtures of 2 and 3 experts with soft competition and mixtures of 2 experts with hard competition. The tests for Expected Shortfall are more conclusive and reject all models except mixtures of 2 and 3 experts with soft competition. Furthermore, the p-values obtained for the rejected models are very low, which indicates that they are clearly insufficient to capture the tails of the conditional distribution. This observation is confirmed by the normal probability plots displayed in Fig. 2. To summarize, soft competition between experts outperforms the other strategies considered. According to the experiments and statistical tests it is not possible to tell which of the mixtures (2 or 3 experts) performs best. More than 3 experts would probably lead to overfitting.

A Wilcoxon rank test [24] has been carried out to detect differences in mean square prediction error between models with the same number of experts. The only significant difference appears between soft competitive and collaborative

Table 1. p-values for the different statistical tests and Mean Square prediction Error (MSE). The values highlighted in boldface correspond to the highest p-values for mixtures of 2 and 3 experts, respectively.

#experts	Strategy	VaR 99%	VaR 95%	Exc 99%	Exc 95%	ES 99%	ES 95%	MSE
1	-	0.01	0.93	0.07	0.95	$3 \cdot 10^{-8}$	$2 \cdot 10^{-3}$	0.9909
2	CL	0.01	0.19	0.03	0.39	$2 \cdot 10^{-9}$	10^{-4}	0.9950
	SC	**0.26**	**0.56**	**0.50**	**0.72**	**0.15**	**0.13**	0.9916
	HC	0.05	0.48	0.07	0.60	$3 \cdot 10^{-6}$	$2 \cdot 10^{-3}$	0.9925
3	CL	0.02	0.08	0.02	0.17	$8 \cdot 10^{-7}$	$2 \cdot 10^{-4}$	0.9962
	SC	**0.44**	**0.31**	**0.39**	**0.54**	**0.16**	**0.10**	0.9925
	HC	0.02	0.07	0.03	0.39	$4 \cdot 10^{-6}$	$5 \cdot 10^{-4}$	0.9952

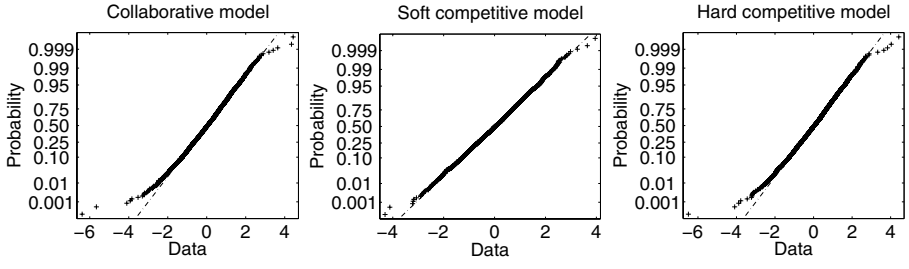


Fig. 2. Normal probability plots of the transformed sample points for the models with 2 experts. Collaboration and Hard Competition strategies fail to correctly describe the loss tail of the distribution. Plots for the models with 3 experts are very similar.

models with 2 experts (i.e. it is possible to reject the hypothesis that those models have the same error. The p-value obtained is 0.03). A similar test indicates that there are no significant differences between the prediction error of a single AR(1) compared with the mixtures of 2 and 3 experts with soft competition.

It is interesting to analyze why collaboration and hard competition are comparatively less accurate than soft competition in capturing the tails of the distribution. The collaborative strategy models the conditional distribution as a single Gaussian whose mean and variance are a weighted average of the means and variances of the Gaussians that correspond to each of the experts. In hard competition the conditional distribution predicted is the Gaussian corresponding to the expert that is active at that particular time. Apparently a single Gaussian distribution, even with time-dependent mean and variance, can not account for the heavy tails of the distribution of returns (see Fig. 2). By contrast, the soft competition strategy predicts a time-dependent mixture of Gaussians, one per expert. The resulting hypothesis space is more expressive and can account for the excess of kurtosis in the conditional distribution of log-returns. Hence, we conclude that the proper dynamical extension of the mixture of Gaussians paradigm to model the conditional probability of log-returns is a mixture of autoregressive experts with soft competition.

References

1. Jorion, P.: Value at Risk. McGraw-Hill Professional (2000)
2. Artzner, P., Delbaen, F., Eber, J.M., Heath, D.: Coherent measures of risk. Mathematical Finance **9**(3) (1999) 203–228
3. Cont, R.: Empirical properties of asset returns: stylized facts and statistical issues. Quantitative Finance **1**(2) (2001) 223–236
4. Kon, S.J.: Models of stock returns–a comparison. Journal of Finance **39**(1) (1984) 147–165
5. Mandelbrot, B.: The variation of certain speculative prices. Journal of Business **36**(4) (1963) 394–419
6. Fama, E.F., French, K.R.: Permanent and temporary components of stock prices. The Journal of Political Economy **96**(2) (1988) 243–276
7. Akgiray, V.: Conditional heteroscedasticity in time series of stock returns: Evidence and forecasts. The Journal of Business **62**(1) (1989) 55–80
8. Engle, R.: Autoregressive conditional heteroskedasticity with estimates of the variance of U.K. inflation. Econometrica **50** (1982) 987–1008
9. Bollerslev, T.: Generalized autoregressive conditional heteroscedasticity. Journal of Econometrics **31** (1986) 307–327
10. Suárez, A.: Mixtures of autoregressive models for financial risk analysis. Lecture Notes in Computer Science **2415** (2002) 1186
11. Vidal, C., Suárez, A.: Hierarchical mixtures of autoregressive models for time-series modeling. Lecture Notes in Computer Science **2714** (2003) 597–606
12. Jacobs, R.A., Jordan, M.I., Nowlan, S., , Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation **3** (1991) 1–12
13. Jordan, M.I., Jacobs, R.A.: Hierarchical mixtures of experts and the EM algorithm. Neural Computation **6** (1994) 181–214
14. Sociedad de Bolsas: Histórico Cierres Índices Ibex. http://www.sbolsas.es (2006)
15. Hamilton, J.D.: Time Series Analysis. Princeton University Press (1994)
16. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press (1996)
17. Jacobs, R.A., Jordan, M.I., Barto, A.G.: Task decompostiion through competition in a modular connectionist architecture: The what and where vision tasks. In: Machine Learning: From Theory to Applications. (1993) 175–202
18. Mathworks: Matlab Optimization toolbox 2.2. Mathworks, Inc., Natick, MA (2002)
19. Hamilton, J.D.: A quasi-bayesian approach to estimating parameters for mixtures of normal distributions. Journal of Business & Economic Statistics **9**(1) (1991) 27–39
20. Rosenblatt, M.: Remarks on a multivariate transformation. The Annals of Mathematical Statistics **23**(3) (1952) 470–472
21. Kerkhof, J., Melenberg, B.: Backtesting for risk-based regulatory capital. Journal of Banking & Finance **28**(8) (2004) 1845–1865
22. Kupiec, H.: Techniques for verifying the accuracy of risk management models. Journal of Derivatives **3** (1995)
23. van der Vaart, A.W.: Asymptotic Statistics. Cambridge University Press (2000)
24. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bulletin **1**(6) (1945) 80–83

Kernel Regression Based Short-Term Load Forecasting

Vivek Agarwal, Anton Bougaev, and Lefteri Tsoukalas

Purdue University, Applied Intelligent Systems Laboratory (AISL),
West Lafayette, IN 47907, USA
{agarwal1, bougaev, tsoukala}@purdue.edu

Abstract. Electrical load forecasting is an important tool in managing transmission and distribution facilities, financial resources, manpower, and materials at electrical power utility companies. A simple and accurate electrical load forecasting scheme is required. Short-term load forecasting (STLF) involves predicting the load from few hours to a week ahead. A simple non-parametric kernel regression (KR) approach for STLF is presented. Kernel regression is a linear approach with the ability to handle nonlinear information. A Gaussian kernel whose bandwidth selected by the Direct Plug-in (DPI) method is utilized. The performance comparison of the proposed method with artificial neural network (ANN), ordinary least squares (OLS), and ridge regression (RR) predictions on the same data set is presented. Experimental results show that kernel regression performs better than ANN forecaster on the given data set. The method proposed provides analytical solution, features optimal bandwidth selection, which is more instructive compared to ANN architecture and its other parameters.

1 Introduction

An efficient management of transmission and distribution facilities, financial resources, manpower, and material by electrical power utility companies is vitally important to meet the growing demand for electric power. Electrical load forecasting schemes developed to predict the power demand ahead of time have earned importance and popularity among utility companies over years. It can be classified into three types, namely, short term forecasting which is usually from one hour to one week, medium term forecasting which is usually from a week to a year, and long term forecasting which is more than one year. Apart from these, Charytoniuk et al. [4] proposed a very short-term load forecasting scheme predicting load for single minute to several dozen of minutes using ANN.

Electrical load forecasting is affected by number of factors, such as weather factors, time factors, and other random factors. Weather factors includes temperature, humidity, wind, and so on. Weather factors are considered to be of primary importance especially for STLF. Hippert et al. [8] showed that most of the STLF techniques take temperature information into consideration. Time factors include the time of the year, day of the week, and hour of the day. The load pattern differs over holidays, weekdays, and weekends (Saturday and Sunday).

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 701–708, 2006.

Even the days preceeding and following holidays and weekends show different load patterns. The load pattern for a week is shown in Figure 1. Other random factors affecting STLF are population of the area, consumer products and their age, economic and demographic data.

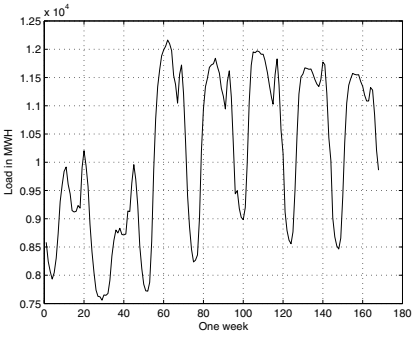


Fig. 1. Weekly load pattern

Load forecasting research has led to the development of different forecasting methods and models, that have been applied with varying degree of success. Hippert et al. [8] classified these models and methods into time series (univariate) models and causal models. In first class, the load is modeled as a function of its past observed values and in second class, the load is modeled as a function of factors such as past load, temperature, and social variables. Reviews [8] [14] show that artificial neural networks have been one of the most widely used tool for load forecasting. Researchers in the field of electric load forecasting have applied different types of ANNs for STLF. Vila et al. [12] applied recurrent neural network for STLF. Ranaweera et al. [10] presented radial basis function neural network based STLF. Researchers have also ensembled ANNs with fuzzy and wavelet approaches for load forecasting. Daneshdoost et al. [5] used fuzzy set along with ANN for STLF. Gao et al. [6] proposed neural-wavelet approach for load forecating and Zhang et al. [15] presented an adaptive neural-wavelet for STLF. Papalexopoulos et al. [9] proposed a regression based approach for STLF. Gao et al. [7] used short-term elasiticity for STLF using intelligent tools. Alves da Silva et al. [1] and Bartkiewicz et al. [3] presented methods to compute the confidence interval for ANN based STLF.

We present a simple non-parametric kernel regression approach for short-term load forecasting. This approach is equivalent to locally weighted learning that fits the data points only near the query point based on the distance function. The distance function is weighted using a kernel function and the bandwidth of the kernel function controls the amount of local averaging to be performed. There are many types of kernel functions that can be applied, but [2] illustrates that the choice of type of kernel function becomes insignificant, if the data set is sufficiently large. The kernel regression estimate depends upon the location

of the query point and the process defers the estimation until a query point is provided, hence it is also known as *lazy learning*. In this paper, the Euclidean distance computed between the query point and the neighboring data is weighted using a Gaussian kernel. The bandwidth of the Gaussian kernel is selected using DPI method [11]. The kernel regression estimate is compared with the multilayer perceptron (MLP) feedforward ANN, OLS, and RR estimations. The metric mean absolute percentage error (MAPE) is used to evaluate the accuracy of the prediction.

The rest of the paper is organized as follows. A discussion on kernel regression is presented in Section 2. A theory on the selection of a smoothing parameter (bandwidth) of the kernel function using Direct Plug-in method is discussed in Section 3. Experimental results obtained using kernel regression, ANN, OLS, and RR are discussed and presented in Section 4. Finally, conclusion is drawn in Section 5.

2 Kernel Regression

Kernel regression is equivalent to locally weighted polynomial fitting and is a non-parametric regression technique [2] [13]. There are two forms of local regression models, namely, univariate regression and multivariate regression. We use multivariate regression for STLF. The multivariate kernel regression model is given by,

$$\varepsilon_i + y_i = f\left(\mathbf{x}_i, \beta\left(\mathbf{q}\right)\right) \quad , \tag{1}$$

where y_i is the response values, $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ik})$ is a vector of predictors for the ith of n observations having a common density confined to a compact set $S \supset \Re$, k denotes the number of bins, ε_i is normally and independently distributed noise, $\beta(\mathbf{q})$ are the kernel coefficients at every query $\mathbf{x_q}$, and $f(.)$ is the function relating the values of response y_i to the predictors. $f(\mathbf{x}_i, \beta(\mathbf{q}))$ is a local model and can have a different set of parameters $\beta(\mathbf{q})$ for each query. In kernel regression, every computation is with respect to a query. The output response at every query for a nonlinear model is obtained by minimizing the cost function,

$$c\left(\mathbf{q}\right) = \sum_{i=1}^{n} \left(\left(f\left(\mathbf{x}_i, \beta\left(\mathbf{q}\right)\right) - y_i\right) K\left(\frac{d\left(\mathbf{x}_i, \mathbf{x_q}\right)}{H}\right)\right) \quad , \tag{2}$$

where K is the kernel function, H is the bandwidth matrix, and d is the distance function. The best estimate $\hat{y}_i(\mathbf{q})$ is obtained by minimizing the cost at every query $\mathbf{x_q}$. Thus, the process of fitting a multivariate kernel regression involves, (i) defining a distance function, (ii) selecting the kernel or the weighting function, (iii) selecting bandwidth of the kernel function, and (iv) specifying the order of the polynomial fit. There are different distance measures [2]. We use scalar Euclidean distance measure between the given data vector and the query $\mathbf{x_q}$,

$$d(\mathbf{x}_i, \mathbf{x_q}) = \sqrt{\sum_i (\mathbf{x}_i - \mathbf{x_q})^2} = \sqrt{\sum_i \left((\mathbf{x}_i - \mathbf{x_q})^T (\mathbf{x}_i - \mathbf{x_q})\right)} \quad . \tag{3}$$

The distance computed is weighted using a kernel function. There are different kernel functions [2] [13]. The choice of kernel function becomes insignificant if the data set is large [2]. We choose Gaussian kernel function,

$$w_{\mathbf{q}} = K_H\left(\frac{d(\mathbf{x}_i, \mathbf{x_q})}{H}\right) = \exp\Big(-\big(d(\mathbf{x}_i, \mathbf{x_q})/H\big)\Big)^2 \ , \tag{4}$$

where $w_{\mathbf{q}}$ the weight computed at a particular query and K_H is the Gaussian kernel function. The kernel regression coefficients are obtained using,

$$\beta(\mathbf{q}) = (\mathbf{X_q}^T\mathbf{W_q}\mathbf{X_q})^{-1}\mathbf{X_q}^T\mathbf{W_q}\mathbf{Y} \ , \tag{5}$$

where,

$$\mathbf{Y} = (y_i, \ldots, y_n)^T \text{is the output vector,}$$

$$\mathbf{W_q} = \text{diag}(w_{q1}, w_{q2}, \ldots, w_{qm}) \text{ is the weight matrix,}$$

$$\mathbf{X_q} = \begin{pmatrix} 1 & \mathbf{x}_i - \mathbf{x_q} & \ldots & (\mathbf{x}_i - \mathbf{x_q})^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \mathbf{x}_n - \mathbf{x_q} & \ldots & (\mathbf{x}_n - \mathbf{x_q})^p \end{pmatrix} \ ,$$

and p is the order of the polynomial fit. The output estimate $(\widehat{Y})$ is obtained using the expression,

$$\widehat{Y}(\mathbf{x_q}, p, H) = e_j^T\beta(\mathbf{q}) \ , \tag{6}$$

where e_j^T is the $(p+1)\times 1$ vector having 1 in the first entry and all other entries equal to zero.

3 Parameter Selection

In any non-parametric regression procedure, an important choice to be made is the amount of local averaging to be performed to obtain the regression estimate. For kernel type estimator, this is controlled by a parameter known as the bandwidth. The choice of the bandwidth controls whether the data is over fitted or under fitted. Mean Square Error (MSE) is the most common approach to measure the closeness of the predicted value to its actual value. MSE is a sum of squared bias and the variance. The bias consideration favors the choice of small value of the bandwidth which may lead to over fitting while the variance consideration favors the large choice of the bandwidth which leads to under fitting. Optimal selection of the bandwidth is essential in order to avoid either of the extreme cases.

Ruppert et al. [11] proposed a data driven "Plug-in" bandwidth selector that estimates the correct amount of smoothing. In this section, we will show the bandwidth estimation for a univariate case. Thus equation (6) for a univariate

case is $\widehat{Y}(x_{\mathrm{q}}, p, h)$, where x_{q} is a query point and h is the single bandwidth value. The h that is used for entire range of data is known as *global* bandwidth. We in our implementation perform univariate estimation of h and apply the same value to the bandwidth matrix, i.e., all the elements of bandwidth matrix H have same h. An optimal h has to be found. Ruppert et al. [11] used conditional weighted mean integrated squared error (MISE) of $\widehat{Y}(x_{\mathrm{q}}, p, h)$ given by,

$$\mathrm{MISE}\{\widehat{Y}(x_{\mathrm{q}}, p, h)|x_1, \ldots, x_k\} = E\left[\int \{\widehat{Y}(x_{\mathrm{q}}) - Y(x_{\mathrm{q}})\}^2 f(x_{\mathrm{q}}) dx_{\mathrm{q}} | x_1, \ldots, x_k\right] , \tag{7}$$

to obtain an asymptotically optimal bandwidth. They showed that MISE-optimal bandwidth has the asymptotic approximation as,

$$h_{\mathrm{MISE}} \approx \left[\frac{(p+1)(p!)^2 R(K_{hp}) \int_S v(x_{\mathrm{q}}) dx_{\mathrm{q}})}{2\mu_{p+1}(K_{hp})^2 \int_S Y^{(p+1)}(x_{\mathrm{q}})^2 f(x_{\mathrm{q}}) dx_{\mathrm{q}} n}\right]^{1/(2p+3)} , \tag{8}$$

where K_{hp} is the $(p+1)$th order kernel, $\int_S v(x_{\mathrm{q}}) dx_{\mathrm{q}}$ is the variance, and $\int_S Y^{(p+1)}(x_{\mathrm{q}})^2 f(x_{\mathrm{q}}) dx_{\mathrm{q}}$ is the regression functional whose kernel estimate is shown in detail in [11]. The (8) can be written as,

$$h_{\mathrm{MISE}} \approx C_1 \left[\frac{(p+1)(p!)^2 \int_S v(x_{\mathrm{q}}) dx_{\mathrm{q}})}{\int_S Y^{(p+1)}(x_{\mathrm{q}})^2 f(x_{\mathrm{q}}) dx_{\mathrm{q}} n}\right]^{1/(2p+3)} , \tag{9}$$

where $C_1 = [R(K_{hp})/\mu_{p+1}(K_{hp})^2]$ is equal to $\{1/2\sqrt{\pi}\}^{1/5}$ in the case of Gaussian kernel. The derivation of variance and kernel estimate of the regression can be found in [11].

4 Experimental Results

In this section, we present the implementation details of the kernel regression and other algorithms for STLF. The data set was provided by ComEd/Excelon in the scope of research effort conveyed by the Consortium for the Intelligent Management of the Electric Power Grid (CIMEG). The data set contains information about hourly load, hourly temperature, and other weather details like wind chill index and humdity. In our experiment, we use only the historic load data of the year 2000 for future short-term load predictions. The experimentation involves two main steps, namely, training and testing. The load data is split into three subsets: training data, validation data, and test data. The training load data is used to obtain a trained model. The parameters of the trained model are optimized and evaluated using the validation data. The optimized trained model is used to predict load of previously unseen test data. The prediction accuracy is measured using MAPE given by,

$$MAPE(\%) = \frac{1}{N} \frac{|L_{\mathrm{actual}} - L_{\mathrm{predicted}}|}{L_{\mathrm{actual}}} \times 100 , \tag{10}$$

where N is the length of the test sample, L_{actual} is the actual load value, and $L_{\text{predicted}}$ is the predicted load value. By using a fixed size window and ahead prediction time, i.e., lead time, which is 24 hours and 1 hour to 48 hours respectively in our case, the input design matrix $\mathbf{X}$ and corresponding output vector $\mathbf{Y}$ are obtained. The dimensions of $\mathbf{X}$ and $\mathbf{Y}$ are 24×400 and 1×400 respectively for the training data. Similarly, for testing its 24×150 and 1×150 for design and output matrices respectively.

In the case of kernel regression, during the training process, a column vector is selected as a query vector and its distance from other column vectors is computed using the expression in (3). The distance computed is weighted using a Gaussian kernel as in (4). The global bandwidth of the Gaussian kernel is selected using DPI method discussed in Section 3. Using (5) and (6), the estimate of the load for a single query vector is obtained. The same procedure is repeated by taking remaining 399 column vectors as query vectors individually and computing the estimate each time. The optimal global bandwidth (h) obtained using a univariate DPI method for each lead time of 1 hour, 1 day, and 2 day is 0.01, 0.0085, and 0.02 respectively. The global bandwidth values obtained during the training process are used in the testing process. A zero order ($p = 0$) polynomial fit is applied in our implementation. Thus, we obtain kernel regression forecasting for every lead time.

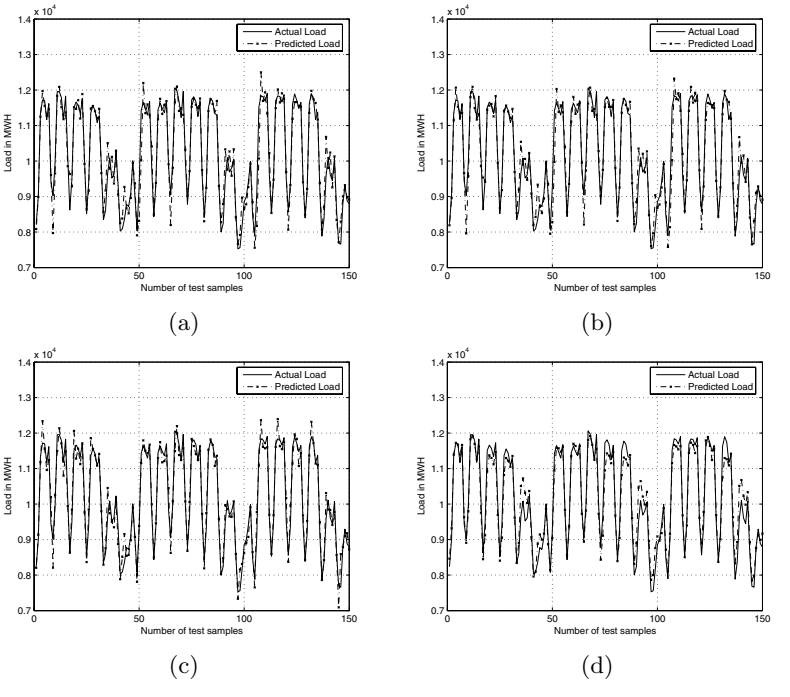


Fig. 2. Hourly load prediction using (a) OLS, (b) RR, (c) ANN, and (d) KR

Table 1. MAPE(%) obtained by each of the algorithm

Algorithms / Lead time	1 hour	1 day	2 day
OLS	2.8	8.5	10.9
RR	2.7	8.5	10.7
ANN	2.4	4.9	4.5
KR	2.6	4.7	4.1

The ANN used for comparison in this paper is MLP feedforward network using backpropagation training algorithm and sigmoid activation function. The ANN of architecture 24-24-1 is used for STLF on the same data set, where 24 corresponds to the number of input nodes, 24 corresponds to the number of hidden nodes in a single hidden layer, and 1 corresponds to the number of output node.

The ridge regression and ordinary least squares are also tested for STLF. The ridge regression coefficients are obtained using,

$$W_{ridge} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{Y} \ , \tag{11}$$

where λ is the regularization parameter and $\mathbf{I}$ is an identity matrix. Equation (11) is equivalent to OLS estimate, when $\lambda = 0$. The regularization parameter is selected using a cross validation method. The performance of the each of the algorithm in predicting 1 hour load ahead is shown in Figure 2. The MAPE values obtained by each of the algorithm for predicting 1 hour, 1 day, and 2 day ahead load is presented in Table 1. From Figure 2 and Table 1, we observe that kernel regression not only matched the performance of ANN, but also betters it. In Figure 2(d), we observe that at certain sections of the load profile, KR overestimates the load, which is primarily due to the choice of global bandwidth. If bandwidth is selected locally, the issue of over estimation can be resolved.

Thus this locally weighted scheme with the ability to handle nonlinear information shows the ability to perform STLF in a simple and accurate way.

5 Conclusions

In this paper, we reported an experimental study of a non-parametric kernel regression method for STLF. The method is simple, local, and linear with the ability to handle nonlinear information. A Gaussian kernel was used in our implementation and bandwidth was selected using DPI method. Experiments show that kernel regression method is able to perform more accurate STLF compared to ANN, OLS, and RR on the given data set.

References

1. Alves da Silva, A.P. and Moulin, L.S.: Confidence intervals for neural network based short-term load forecasting. IEEE Trans. Power Sys. **15(4)** (2000) 1191–1196

2. Atkeson, C.G., Moore, A.W., and Schaal, S.: Locally Weighted Learning. Art. Intell. Review. **11** (1997) 11–73
3. Bartkiewicz, W., Gontar, Z., Zielinski, J.S., and Bardzki, W.: Uncertainty of the short-term electrical load forecasting in utilities. Int. Joint Conf. on Neural Networks. **6** (2000) 235–240
4. Charytoniuk, W. and Chen, M.S.: Very short-term load forecasting using neural networks. IEEE Trans. Power Sys. **15(1)** (2000) 263–268
5. Daneshdoost, M., Lotfalian, M., Bumroonggit, G., and Ngoy, J.P.: Neural network with fuzzy set-based classification for short-term load forecasting. IEEE Trans. Power Sys. **13(4)** (1998) 1386–1391
6. Gao, R. and Tsoukalas, L.H.: Neural-wavelet methodology for load forecasting. J. of Intell. and Robotic Sys. **31** (2001) 149–157
7. Gao, R., Wang, X., Bougaev, A., Schooley, D.C., and Tsoukalas, L.H.: Short-term elasticities via Intelligent tools for modern power systems. IEEE MedPower02 3rd Mediterranean Conference and Exhibition on Power Generation, Transmission, Distribution and Energy Conversion. (2002)
8. Hippert, S.H., Pedreira, C.E., and Souza, R.C.: Neural networks for short-term load forecasting: A review and evaluation. IEEE Trans. Power Sys. **16(1)** (2001) 44–55
9. Papalexopoulos, A.D. and Hesterberg, T.C.: A regression-based approach to short-term system load forecasting. IEEE Trans. Power Sys. **5(4)** (1990) 1535–1547
10. Ranaweera, D.K., Hubele, N.F., and Papalexopoulos, A.D.: Application of radial basis function neural network model for short-term load forecasting Proc. IEE–Gen. Trans. Distri. **142(1)** (1995) 45–50
11. Ruppert, D., Sheather, S.J., and Wand, M.P.: An effective bandwidth selector for local least squares regression. J. of the Amer. Stat. Asso. **90(432)** (1995) 1257–1270
12. Vila, J.P., Wagner, V., and Neveu, P.: Recurrent neural network for short-term load forecasting. IEEE Trans. Power Sys. **13(1)** (1998) 126–132
13. Wand, M.P. and Jones, M.C.: Kernel Smoothing. (2000) CRC Press LLC, Florida
14. Zhang, G., Patuwo, B.E., and Hu, M.Y.: Forecasting with artificial neural networks: The state of the art. Int. J. of Forecasting. **14** (1998) 35–62
15. Zhang, B.L. and Dong, Z.Y.: An adaptive neural-wavelet model for short-term load forecasting. Electric Power Sys. Research. **59** (2001) 121–129

Electricity Load Forecasting Using Self Organizing Maps

Manuel Martín-Merino and Jesus Román

Universidad Pontificia de Salamanca
C/Compañía 5, 37002, Salamanca, Spain
manuel@upsa.es, jaromanga.eui@upsa.es

Abstract. Electricity load forecasting has become increasingly important for the industry. However, the accurate load prediction remains a challenging task due to several issues such as the nonlinear character of the time series or the seasonal patterns it exhibits.

Several non-linear techniques such as the SVM have been applied to this problem. However, the properties of the load time series change strongly with the seasons, holidays and other factors. Therefore global models such as the SVM are not suitable to predict accurately the load demand.

In this paper we propose a model that first splits the time series into homogeneous regions using the Self Organizing Maps (SOM). Next, an SVM is locally trained in each region.

The algorithm proposed has been applied to the prediction of the maximum daily electricity demand. The experimental results show that our model outperforms several statistical and machine learning forecasting techniques.

1 Introduction

Electricity load forecasting has become increasingly important in the last years for the industry due to the deregulation of the electricity markets. In particular, accurate short term load forecasting has a significant impact on the operational efficiency of the power system [5,6]. However, the accurate load prediction remains a difficult task due to several issues such as the nonlinear character of the time series or the periodical and seasonal patterns it exhibits [5,1].

A large variety of techniques have been proposed to this aim such as statistical models [2], fuzzy systems [13] or artificial neural networks [5]. More recently, several authors have applied Support Vector Machines (SVM) to time series forecasting [15,1] with remarkable results. SVM are powerful non-linear techniques proposed under a soundness statistical theory that keep a high generalization ability [17]. Unfortunately, the properties of the load time series change locally with time due to seasonal effects, holidays and other factors [1,10]. For instance, the load patterns differ significantly in winter and summer seasons or in weekends and working days. Therefore global models such as the SVM are not suitable to predict accurately the load demand.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 709–716, 2006.

It has been suggested in the literature that splitting the time series into homogeneous regions helps to improve the forecasting accuracy [1,10,8]. Thus [10,8] have applied the Self Organizing Maps (SOM) to identify days of similar load profiles. However, human experts are needed to classify the SOM prototypes which is a serious drawback. In [9,4] a SOM is employed to create Voronoi regions for input and output spaces. Next a frequency table is built that relates both spaces. However the results are not satisfactory when the sample size is not large. Besides, the neighborhood relations induced by the SOM are not considered. Finally, the method proposed by [3] takes advantage of the neighborhood relations induced by the SOM to adjust a locally weighted linear regression. However the SOM is organized without considering the target to be predicted. Therefore relevant information is lost.

In this paper, we present an algorithm to split the time series into homogeneous regions using the SOM. The new model builds a partition of the input space considering the target to be predicted, avoids the overfitting and takes advantage of the neighborhood relations induced by the SOM. The method proposed has been applied to the prediction of the maximum daily electricity demand and compares favorably with wide spread statistical and machine learning techniques.

This paper is organized as follows. Section 2 presents the segmentation algorithm to split the time series into homogeneous regions. In section 3 the proposed algorithm is applied to predict the maximum daily load demand and finally section 4 gets conclusions and outlines future research trends.

2 Time Series Segmentation Using the SOM

As we have mentioned earlier, the patterns of electricity demand change locally due to seasons, holidays and other factors. This suggests that the time series should be segmented into homogeneous regions before any forecasting technique is applied.

In this section we apply the SOM algorithm to the segmentation of the time series. Section 2.1 introduces briefly the SOM batch algorithm. Next, section 2.2 presents the SOM algorithm proposed to segment the time series and discusses the related work.

2.1 Self Organizing Maps

The SOM [7] is a nonlinear visualization technique for high dimensional data. Input vectors are represented by neurons arranged according to a regular grid (usually $1D$-$2D$) in such a way that similar vectors in input space become spatially close in the grid.

In the case of discrete data, the SOM can be obtained from the optimization of the following energy function [12]:

$$E(\mathcal{W}) = \sum_{r} \sum_{x_\mu \in V_r} \sum_{s} h_{rs} D(\mathbf{x}_\mu, \boldsymbol{w}_s)\,, \tag{1}$$

where D denotes the square Euclidean distance and V_r is the Voronoi region corresponding to prototype $\boldsymbol{w}_r$. h_{rs} is a neighborhood function (for instance a Gaussian kernel) that transforms nonlinearly the neuron distances (see [7] for other possible choices). The SOM energy function (1) is minimized when objects that are close together in input space (according to the Euclidean distance) are mapped to neighboring neurons in the grid.

The SOM energy function may be optimized by an iterative algorithm made up of two steps [12].

- First a quantization algorithm is run that represents each pattern by the nearest neighbor prototype.
- Next, the prototypes are organized along the grid of neurons by minimizing the error function (1). The optimization problem can be solved explicitly resulting in a simple iterative adaptation rule for each prototype:

$$\boldsymbol{w}_s = \frac{\sum_{r=1}^{M} \sum_{x_\mu \in V_r} h_{rs} \boldsymbol{x}_\mu}{\sum_{r=1}^{M} \sum_{x_\mu \in V_r} h_{rs}} \tag{2}$$

  where M is the number of neurons and h_{rs} is for instance a Gaussian kernel of width $\sigma(t)$. The kernel width is adapted in each iteration using for instance the rule proposed by [11] $\sigma(t) = \sigma_i(\sigma_f/\sigma_i)^{t/N_{iter}}$, where $\sigma_i \approx M/2$ is usually considered in the literature [7]. Finally, σ_f is a parameter that determines the degree of smoothing of the principal curve generated by SOM [11].

2.2 A SOM Algorithm to Segment the Load Time Series

In this section we describe the algorithm proposed to segment the load time series. To this aim, first a daily load profile is defined considering the load of the previous days and the load to be predicted. Next a SOM batch algorithm (with circular topology) organizes the load profiles along a grid of neurons using a metric that weighs more heavily the more relevant variables. Next, the SOM prototypes are clustered together considering the neighborhood relations induced by the network. Finally, a linear SVM is trained locally in each cluster.

Let $X(t) = [l(t-p+1), \ldots, l(t), u(t-p+1), \ldots, u(t)]$ be the input vector for time t that groups together the load and exogenous variables for the pth previous observations. Most common exogenous variables are the temperature and calendar (day of the week). Let $x(t+1)$ be the target load. The segmentation algorithm we propose proceeds as follows:

First, the input patterns to train the SOM are defined as:

$$\boldsymbol{y}(t) = (X(t), x(t+1)), \tag{3}$$

where $x(t+1)$ is known only for the training data and allow us to organize the SOM taking into account the relation between the input patterns $X(t)$ and the predictions $x(t+1)$ [18].

Next the input patterns are organized by a modification of the SOM batch algorithm introduced in section 2.1. A circular topology for the network has been

chosen to take into account the periodicity of the load demand. Besides, the proximities of the load profiles are evaluated by a weighted Euclidean distance defined as:

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{k=1}^{p} \omega_k (x_{ik} - x_{jk})^2 \,, \tag{4}$$

where ω_k is a coefficient that should give more weight to those variables considered more relevant in the input patterns. Obviously, the variables that are strongly correlated with the prediction should be weighted more heavily. Therefore ω_k is defined as the correlation coefficient between the kth variable and the prediction.

$$w_k = \frac{\sum_{t=k}^{N-1} (x_{t+1} - \bar{x})(x_{t-k+1} - \bar{x})}{\sum_{t=k}^{N-1} (x_{t+1} - \bar{x})^2} \tag{5}$$

where $\bar{x}$ denotes the average of the load time series. These coefficients will give rise to local neighborhoods that improve particularly the performance of linear models.

Substituting the distance (4) in the energy function (1) we can easily derive an updating rule for each prototype coordinate considering the new metric:

$$\boldsymbol{w}_{sk} = \frac{\sum_{r=1}^{M} \sum_{x_\mu \in V_r} \omega_k h_{rs} \boldsymbol{x}_{\mu k}}{\sum_{r=1}^{M} \sum_{x_\mu \in V_r} \omega_k h_{rs}} \tag{6}$$

Once the SOM is organized, the prototypes are clustered into a certain number of groups. Several algorithms have been proposed in the literature (see for instance [7,20,19]) that cluster the data using the SOM. In this paper we follow the approach of [19]. The SOM prototypes are grouped considering an agglomerative hierarchical clustering algorithm. Merges are carried out in the tree considering the ratio of the between clusters distance and the within-cluster distance. The distance between clusters is defined as the average distance among the objects belonging to both clusters and the withing-cluster distance as the average distance for the objects assigned to the same cluster.

However, the method proposed by [19] does not consider the relations induced by the grid of neurons and therefore, relevant information about the local topology of the data is lost. To avoid this problem, the Euclidean distance between the prototypes in the hierarchical algorithm is substituted by a geodesic distance. This measure evaluate the distance between prototypes as the sum of the Euclidean distances between adjacent prototypes following the shortest path along the grid of neurons. The idea is similar to the one proposed in [16] but substituting the adjacency graph by the SOM.

Once the prototypes have been clustered, a linear SVM is trained locally in each cluster. The linear character of the SVM will help to avoid overfitting the data. The forecasting for new test patterns can be done in two steps. First the new data is assigned to the cluster corresponding to the nearest prototype. Next the SVM associated to this cluster is considered to perform the prediction.

The model presented above is related to [18] but there are several differences that are worth to mention.

First, our model organizes the SOM considering a weighted Euclidean measure which will help to improve the prediction of linear models inside each cluster. Second, the model presented in [18] does not cluster the SOM and adjusts one model inside each Voronoi region. Therefore the model is prone to overfitting when the sample size is not large. Our method clusters the data thus preventing the overfitting when the sample size is not large. Moreover, the high generalization of the linear SVM will help to avoid this problem. Finally, notice that the clustering of the SOM has been done considering a geodesic distance which will allow to discover the local non-linear structure of the data.

3 Experimental Results

In this section we have applied the algorithm proposed to the prediction of the maximum daily electricity demand.

The experimental data were provided by the Eastern Slovakian Electricity Corporation for the EUNITE competition [14]. The data set is available from the Internet (neuron.tuke.sk/competition/index.php).

The company provided the maximum daily load demand for the years 1997 and 1998. The average daily temperature is provided as well for the whole period. The problem is to predict the maximum daily demand for the test set (January 1999).

The evaluation of the forecasting techniques should be done carefully in order to guaranty the objectivity of the results. Notice that the industry complains because the results about the model performance are often meaningless. In this paper we have considered three objective measures that are well accepted by the industry [5].

- MAPE (Mean Absolute Percent Error): It is somewhat of a standard in the electricity supply industry and is defined as:

$$\text{MAPE} = 100\frac{\sum_{i=1}^{n}\left|\frac{l(i)-\hat{l}(i)}{l(i)}\right|}{n} \tag{7}$$

  where $l(i)$ denotes the load and $\hat{l}(i)$ is the predicted load for $t = i$. n is the sample size.
- RMSE (Root Mean Square Error): It is a quadratic error function that gives more weight to large errors. This measure complements the previous one because large error may have disastrous consequences for an utility. It can be written as:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}\left(\frac{l(i)-\hat{l}(i)}{l(i)}\right)^2}{n}} \tag{8}$$

- ME (Maximum Error): It evaluates the maximum error. As we have mentioned earlier a single large error could be disastrous for the industry. It is defined as:

$$\text{ME} = \max_i |l(i) - \hat{l}(i)|, \quad i = 1, \ldots, n \tag{9}$$

Table 1. Prediction errors for the proposed method (SOM+SVM). Three widely used techniques such as Heuristic, Holt-Winters and SVM have been considered as reference.

Method	**MAPE**	**ME**	**RMSE**
Heuristic	2,78%	47	25.08
Holt Winters	1.84 %	34.02	15.91
Linear SVM	1.69%	33.98	14.95
SVM (RBF kernel)	1.56%	37.24	15.15
SOM + SVM	1.46%	32.36	13.66

Parameters: $\epsilon = 0.2$; linear SVM , $C = 1$; SVM (RBF), $C = 0.5$, $\gamma = 0.1$; *Holt-Winters* $p = 7$; SOM + SVM $k = 7$, $\epsilon = 0.4$, $C = 0.5$.

Table (1) compares the performance of the proposed algorithm with three well known techniques. The first row is a Heuristic method that predicts the maximum daily electricity demand by the observed load seven days ago. This simple heuristic is frequently considered by the industry. Row 2 corresponds to the Holt-Winters method that has been widely used in the time series literature [2]. Rows 3 and 4 show the performance of the SVM with linear and non-linear RBF kernels. Finally row 5 yields the results for the proposed algorithm. All the parameters of the forecasting techniques have been determined by crossvalidation.

Input patterns to the SOM are defined as:

$$\boldsymbol{x}_t = (u_{t-7}, \ldots, u_t, l_{t-6}, \ldots, l_t, l_{t+1}) \tag{10}$$

where $(u_{t-7}, \ldots, u_t)$ are eight binary variables that codify the day of the week including holidays. l_t denotes the maximum load for day t, l_{t+1} is the target load and obviously is only known for the training data. Notice that the temperature has been discarded because the correlation with the output depends strongly with time and therefore the predictions are worse considering the temperature. This has been observed by other authors when working on the same dataset [1]. The window size of 7 has been chosen empirically.

From the analysis of table (1) we can draw the following conclusions:

- The algorithm proposed improves significantly the heuristic method employed by some electric companies according to all the objective functions considered.
- Our algorithm outperforms a representative of the statistical methods such as the Holt-Winters. Particularly the the MAPE error is significantly reduced.
- The segmentation algorithm proposed allow us to improve the performance of both, linear and non-linear Support Vector Machines (SVM). We report that our algorithm reduces the MAPE error of the kernel SVM and improves significantly the ME error. This suggests that our model is more robust to overfitting than the SVM with RBF kernel.

Finally figure (1) illustrates the performance of the proposed algorithm from a qualitative point of view. The observed load has been drawn in continuous trace. The proposed algorithm seems to perform well both for working days and

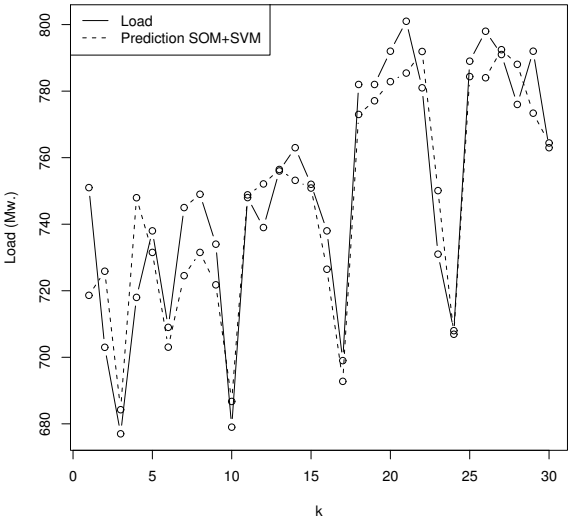


Fig. 1. Electricity load prediction for January 1999

for weekends. Moreover we remark that the prediction error for holidays such as the 6 January is very small. The error is larger just for 1 and 2 January because the load profile of previous days include several holidays which distorts the predictions.

4 Conclusions and Future Research Trends

In this paper we have proposed a new method to segment the load time series into homogeneous regions using the Self Organizing Maps. The new model has been applied to the forecasting of the maximum daily electricity demand. The proposed technique has been compared with several well known alternatives and evaluated exhaustively through several objective functions accepted by the industry.

The empirical results suggest that our model outperforms widely used statistical and machine learning forecasting techniques. Particularly, the proposed method is able to reduce the error of non-linear methods keeping a high generalization ability.

Future research will focus on the prediction of outliers that involve for instance large holidays.

References

1. M.-W. Chang, B.-J. Chen, and C.-J. Lin. EUNITE network competition: Electricity load forecasting, November 2001. Winner of EUNITE world wide competition on electricity load prediction.

2. C. Chàtfield. *The Analysis of Time Series: An Introduction*. Chapman & Hall/CRC Press, New York, fifth edition, 1996.
3. V. Cherkassky, D. Gehring, and F. Mulier. Comparison of adaptive methods for function estimation from samples. *IEEE Transactions on Neural Networks*, 7(4):969-984, July 1996.
4. S. Dablemont, G. Simon, A. Lendasse, A. Ruttiens, F. Blayo, and M. Verleysen. Time series forecasting with SOM and local non-linear models- application to the DAX30 index prediction. In *Workshop on Self-Organizing Maps (WSOM)*, pages 340-345, Hibikino (Japan), September 2003.
5. H. S. Hippert, C. E. Pedreira, and R. C. Souza. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on Neural Networks*, 16(1):44-55, February 2001.
6. A. Khotanzad, R. Afkhami-Rohani, T-L. Lu, A. Abaye, M. Davis, and D. J. Maratukulam. ANNSTLF–a neural-network-based electric load forecasting system. *IEEE Transactions on Neural Networks*, 8(4):835-846, July 1997.
7. T. Kohonen. *Self-Organizing Maps*. Springer Verlag, Berlin, second edition, 1995.
8. R. Lamedica, A. Prudenzi, M. Sforna, M. Caciotta, and V. O. Cencelli. A neural network based technique for short-term forecasting of anomalous load periods. *IEEE Transactions on Power Systems*, 11(4):1749-1756, November 1996.
9. A. Lendasse, M. Cottrell, V. Wertz, and M. Verleysen. Prediction of electric load using kohonen maps- application to the Polish electricity consumption. In *Proceedings of the American Control Conference*, pages 3684-3689, Anchorage, May 2002.
10. F. J. Marín, F. García-Lagos, G. Joya, and F. Sandoval. Peak load forecasting using kohonen classification and intervention analysis, November 2001. EUNITE world wide competition on electricity load prediction.
11. F. Mulier and V. Cherkassky. Self-organization as an iterative kernel smoothing process. *Neural Computation*, 7:1165-1177, 1995.
12. E. Oja and S. Kaski, editors. *Kohonen Maps*, chapter: Energy Functions for Self-Organizing Maps, pages 303-315. Elsevier, Amsterdam, 1999.
13. S. E. Papadakis, J. B. Theocharis, S. J. Kiartzis and A. G. Bakirtzis. A novel approach to short-term load forecasting using fuzzy neural networks. IEEE Transactions on Power Systems, 13(2):480-492, May 1998.
14. I. Rojas and H. Palomares. Soft-computing techniques for time series forecasting. In *Proc. of the European Symposium on Artificial Neural Networks* , pages 93-102, Bruges, Belgium, April 2004.
15. B. Schlkopf, C. J. C. Burges, and A. J. Smola, editors. *Advances in Kernel Methods: Support Vector Learning*, pages 243-253. MIT Press, Massachusetts, 1999.
16. J. B. Tenenbaum, V. de Silva and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319-2323, December 2000.
17. V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, New York, 1998.
18. J. Vesanto. Using the SOM and local models in time-series prediction. In *Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6*, pages 209-214. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, 1997.
19. J. Vesanto and E. Alhoniemi. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, 11(3):586-600, May 2000.
20. S. Wu and T. S. W. S. Chow. Clustering of the self-organizing map using a clustering index based on inter-cluster and intra-cluster density. *Pattern Recognition*, 37:175-188, 2004.

A Hybrid Neural Model in Long-Term Electrical Load Forecasting

Otávio A.S. Carpinteiro, Isaías Lima, Rafael C. Leme,
Antonio C. Zambroni de Souza, Edmilson M. Moreira,
and Carlos A.M. Pinheiro

Research Group on Computer Networks and Software Engineering
Federal University of Itajubá
37500–903, Itajubá, MG, Brazil
{otavio, isaias, leme, zambroni, edmarmo, pinheiro}@unifei.edu.br

Abstract. A novel hierarchical hybrid neural model to the problem of long-term electrical load forecasting is proposed in this paper. The neural model is made up of two self-organizing map nets — one on top of the other —, and a single-layer perceptron. It has application into domains which require time series analysis. The model is compared to a multilayer perceptron. Both the hierarchical and the multilayer perceptron models are endowed with time windows in their input layers. They are trained and assessed on load data extracted from a North-American electric utility. The models are required to predict once every week the electric peak-load and mean-load during the next two years. The results are presented and evaluated in the paper.

1 Introduction

Time series analysis and prediction has found application in many areas of knowledge, such as economy, meteorology, and engineering. The problem involves prediction of future points of the series. It is related with the horizon of predictions — the larger the time horizons, the less accurate the predictions [1].

Many linear and non-linear statistical models have been tried on the problem of time series prediction. Research on statistical models, however, has considered their low accuracy and parameters adjustment to be the points of major concern.

Recently, non-linear neural models have also been tried on such problem [2]. Neural models should include some kind of mechanism to analyse, and possibly, memorize the context information of the series in order to produce reliable predictions.

Neural models which include such mechanisms to time series analysis and prediction are often referred to as spatiotemporal neural models. Time windows [3] and time integrators [4] are by far the most employed mechanisms. Surveys of spatiotemporal neural models are available in literature [5,6,7].

In the electric engineering domain particularly, neural models have been successfully applied to predicting future points of historical load series in short-term horizons [8,9,10,11,12]. Many electric utilities are now employing short-term load

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 717–725, 2006.

forecasting tools based on neural models [13]. Hippert et al. [14] provide a comprehensive review of the application of neural models to short-term load forecasting. The authors examine a collection of papers published between 1991 and 1999.

Long-term forecasting of electric load is nevertheless a different, a much more difficult, and challenging problem. There are very few works in literature which approach it through the use of neural models [15]. The problem is also significant, particularly in Brazil, for Brazilian electric utilities have to predict the load demand for horizons varying from two to five years, according to the new regulations.

This paper introduces a new hierarchical hybrid neural model (HHNM) to approach the problem of long-term load forecasting. The model is an extension of the Kohonen's original self-organizing map [16].

HHNM is a hierarchical model. The hierarchical topology gives to the model the power to process efficiently the context information embedded in the input time series. HHNM holds a very good memory for past events, enabling it to produce better forecasts.

HHNM is compared with the multilayer perceptron (MLP), which has been extensively applied to short-term load forecasting [14], as well as to general time series forecasting [17]. Both HHNM and MLP include time windows in their input layers as memories for analysing and predicting the historical load series.

The paper is divided as follows. The second section presents the data representation. HHNM is introduced in the third section. Training of HHNM and MLP is detailed in the fourth and fifth sections respectively. The sixth section describes the experiments. The seventh section discusses the results on load forecasting. The last section presents the main conclusions of the paper, and indicates some directions for future work.

2 Data Representation

The input data consists of window sequences of load data extracted from a North-American electric utility [18]. Several input files including such data are set. The files contain either weekly peak-loads or weekly mean-loads.

Input windows of four, six, and eight neural units are used in the representation. Two of these units represent a trigonometric coding for the week to be forecast, i.e., $sin(2\pi \cdot week/52)/4$ and $cos(2\pi \cdot week/52)/4$. The other ones represent either the peak- or mean-load at the current and previous weeks. Each unit receives real values.

The electric peak-load and mean-load range in the intervals [1528, 4635] and [1265.917, 3974.333] MWatts, respectively. The load data is pre-processed using ordinary normalization (minimum and maximum values in the [-0.5,0.5] range). There is no particular treatment for holidays.

3 The Hierarchical Hybrid Neural Model

The hierarchical hybrid neural model (HHNM) is shown in Figure 1. It is made up of two distinct neural models — a hierarchical self-organizing model (HSOM),

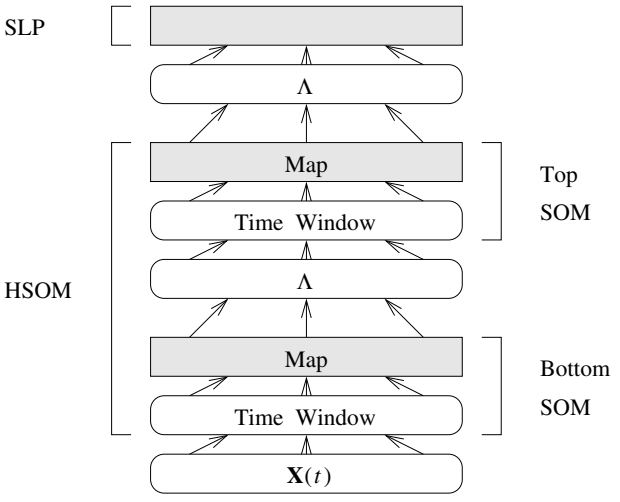


Fig. 1. HHNM

and a single-layer perceptron (SLP). The HSOM, in its turn, is made up of two self-organizing maps (SOMs).

The input to the model is a sequence in time of m-dimensional vectors, $\mathbf{S_1} = \mathbf{X}(1), \mathbf{X}(2), \ldots, \mathbf{X}(t), \ldots, \mathbf{X}(z)$, where the components of each vector are real values. The sequence is presented to the input layer of the bottom SOM, one vector at a time. The input layer has a window of m units, one unit for each component of the input vector $\mathbf{X}(t)$.

For each input vector $\mathbf{X}(t)$, the winning unit $i^*(t)$ in the map is the unit which has the smallest distance $\Psi(i,t)$. For each output unit i, $\Psi(i,t)$ is given by the Euclidean distance between the input vector $\mathbf{X}(t)$ and the unit's weight vector $\mathbf{W}_i$.

Each output unit i in the neighbourhood $N^*(t)$ of the winning unit $i^*(t)$ has its weight $\mathbf{W}_i$ updated by

$$\mathbf{W}_i(t+1) = \mathbf{W}_i(t) + \alpha\Upsilon(i)[\mathbf{X}(t) - \mathbf{W}_i(t)] \tag{1}$$

where $\alpha \in (0,1)$ is the learning rate. $\Upsilon(i)$ is the *neighbourhood interaction function* [19], a Gaussian type function, and is given by

$$\Upsilon(i) = \kappa_1 + \kappa_2 e^{-\frac{\kappa_3[\Phi(i,i^*(t))]^2}{2\sigma^2}} \tag{2}$$

where κ_1, κ_2, and κ_3 are constants, σ is the radius of the neighbourhood $N^*(t)$, and $\Phi(i,i^*(t))$ is the distance in the map between the unit i and the winning unit $i^*(t)$. The distance $\Phi(i',i'')$ between any two units i' and i'' in the map is calculated according to the maximum norm,

$$\Phi(i',i'') = max\left\{|l'-l''|, |c'-c''|\right\} \tag{3}$$

where (l', c') and (l'', c'') are the coordinates of the units i' and i'' respectively in the map.

The neighbourhood interaction function has proved to be useful, indeed. It provokes two main effects. First, it speeds up the training of the network by reducing the number of epochs required. Second, it improves the quality of the map by enforcing its topological order [20]. In rough terms, the neighbourhood interaction function avoids the existence of *local winning units*. The values of the distances $\Psi(i, t)$ increase orderly as the values of the distances $\Phi(i, i^*(t))$ increase.

The input to the top SOM is determined by the distances $\Phi(i, i^*(t))$ of the n units in the map of the bottom SOM. The input is thus a sequence in time of n-dimensional vectors, $\mathbf{S_2} = \Lambda(\Phi(i, i^*(1)))$, $\Lambda(\Phi(i, i^*(2)))$, ..., $\Lambda(\Phi(i, i^*(t)))$, ..., $\Lambda(\Phi(i, i^*(z)))$, where Λ is a n-dimensional *transfer function* on a n-dimensional space domain. Λ is defined as

$$\Lambda(\Phi(i, i^*(t))) = \begin{cases} 1 - \kappa\Phi(i, i^*(t)) & \text{if } i \in N^*(t) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where κ is a constant, and $N^*(t)$ is a neighbourhood of the winning unit.

The sequence $\mathbf{S_2}$ is then presented to the input layer of the top SOM, one vector at a time. The input layer has a window of n units, one unit for each component of the input vector $\Lambda(\Phi(i, i^*(t)))$. The dynamics of the top SOM is identical to that of the bottom SOM.

The input to the SLP is also determined by the distances $\Phi(i, i^*(t))$ of the p units in the map of the top SOM. The input is thus a sequence in time of p-dimensional vectors, $\mathbf{S_3} = \Lambda(\Phi(i, i^*(1)))$, $\Lambda(\Phi(i, i^*(2)))$, ..., $\Lambda(\Phi(i, i^*(t)))$, ..., $\Lambda(\Phi(i, i^*(z)))$, where Λ is a p-dimensional *transfer function* on a p-dimensional space domain, and is also given by equation 4.

The sequence $\mathbf{S_3}$ is then presented to the input layer of the SLP, one vector at a time. The input layer has a window of p units, one unit for each component of the input vector $\Lambda(\Phi(i, i^*(t)))$.

The SLP is employed to map the output produced by the top SOM. It is trained with the usual delta rule [21,22].

4 HHNM Training

The input to HHNM comes through the input layer of the bottom SOM. The input layer of the bottom SOM holds a window of input units. Window sizes of four, six, and eight input units are tested.

The training of the two SOMs takes place in two phases — coarse-mapping and fine-tuning. In the coarse-mapping phase, the learning rate and the radius of the neighbourhood are reduced linearly whereas in the fine-tuning phase, they are kept constant. Several architectures are tested. The bottom SOM is trained with map sizes of 9×9 in 420 epochs, 12×12 in 560 epochs, and 13×13 in 600 epochs. The top SOM is trained with map sizes of 15×15 in 700 epochs, 19×19

in 850 epochs, and 21×21 in 950 epochs. The initial weights are given randomly to both SOMs.

The SLP holds a single unit in its output layer. The output unit has linear activation function. Training is performed through cross validation. Therefore, it is halted whenever the error increases on the testing set. Training is carried out on an epoch-by-epoch basis. Learning rate is reduced by 50% when total error increases, and increased by 2% when error decreases. Momentum is disabled until the end of training if total error increases. The initial weights are given randomly.

5 MLP Training

The MLP holds a window of input units and one output unit. Window sizes of four, six, and eight input units are tested. Hidden units have sigmoid activation functions, whereas the output unit has linear activation function. Several architectures including from one up to twenty-five hidden units are tested.

Training is performed through cross validation on the testing set. It is carried out on an epoch-by-epoch basis. Learning rate is reduced by 50% when total error increases, and increased by 2% when error decreases. Momentum is disabled until the end of training if total error increases. The initial weights are given randomly.

6 Experiments

Four experiments are carried out. In the first experiment, the training set contains 104 weekly peak-loads from 1985 to 1986. The testing set contains 52 weekly peak-loads from 1987. The models are required to foresee the weekly peak-loads from 1988 to 1989. The second experiment is quite similar to the first. In it, the training set includes 208 weekly peak-loads from 1985 to 1986, and from 1988 to 1989. The testing set remains the same, and forecasting spans the time horizon from 1990 to 1991.

In the third experiment, the training set contains 104 weekly mean-loads from 1985 to 1986. The testing set contains 52 weekly mean-loads from 1987. The models are required to foresee the weekly mean-loads from 1988 to 1989. The fourth experiment is quite similar to the third. In it, the training set includes 208 weekly mean-loads from 1985 to 1986, and from 1988 to 1989. The testing set remains the same, and forecasting spans the time horizon from 1990 to 1991.

The forecasts are performed on two models — HHNM and MLP. A comparison of such models is carried out to verify their performance in each experiment.

7 Results

Figures 2 and 3, and Table 1 present the best results achieved by the models on load forecasting. Figures 2 and 3 display the actual load and forecast loads in the

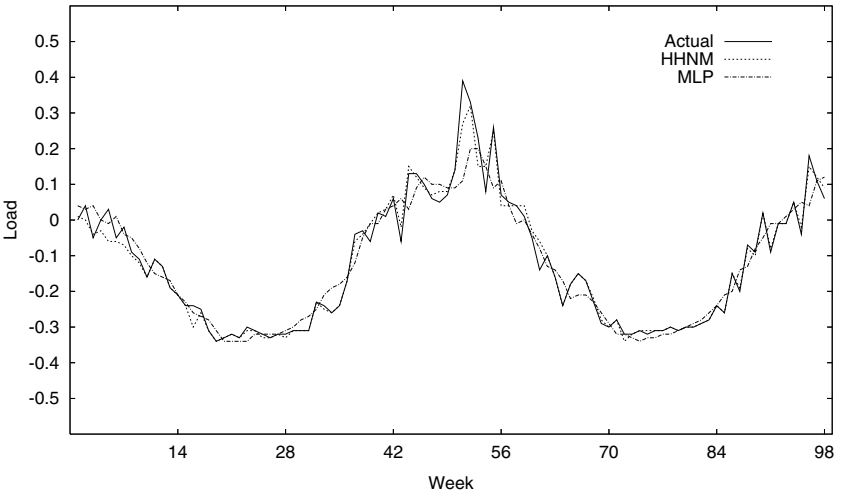


Fig. 2. First experiment

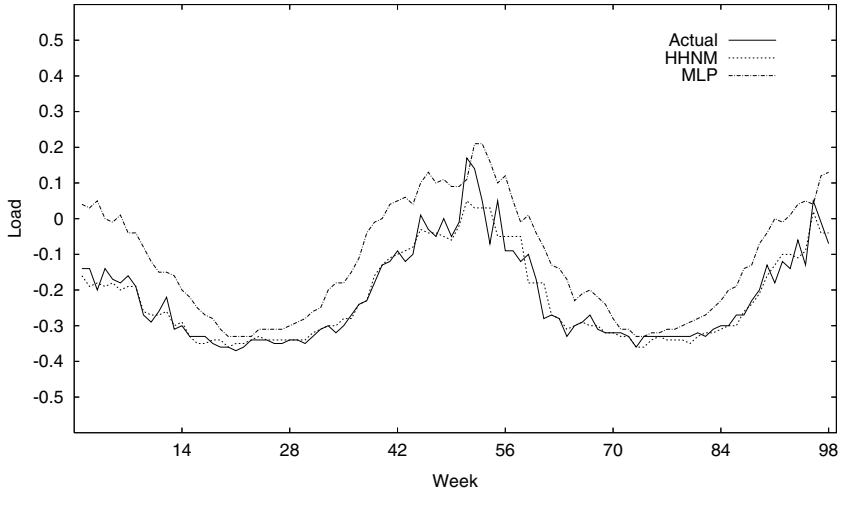


Fig. 3. Third experiment

first and third experiments. Table 1 shows the forecasting errors — mean, maximum, and minimum absolute error, mean absolute percentual error (MAPE), mean square error (MSE), and standard deviation of error — of the models in the four experiments.

The results from HHNM are very promising. HHNM performs much better than MLP both on peak-load and mean-load forecasts.

Figures 2 and 3 show that the forecast load curves produced by HHNM follow the actual ones more accurately than those produced by MLP. Table 1 shows

Table 1. Forecasting errors — mean, maximum, and minimum absolute error, mean absolute percentual error (MAPE), mean square error (MSE), and standard deviation of error — of the models in the four experiments

Models	Errors	Experiments			
		1	2	3	4
HHNM	Mean	0.01	0.03	0.02	0.02
	Max	0.12	0.20	0.12	0.18
	Min	0.00	0.00	0.00	0.00
	MAPE	1.37	2.93	2.68	2.59
	MSE	0.00	0.00	0.00	0.00
	St.Dev.	0.02	0.03	0.03	0.03
MLP	Mean	0.04	0.08	0.10	0.06
	Max	0.28	0.41	0.25	0.46
	Min	0.00	0.00	0.00	0.00
	MAPE	4.05	8.49	13.39	6.87
	MSE	0.00	0.01	0.01	0.01
	St.Dev.	0.04	0.06	0.06	0.07

that the forecasting error values obtained by HHNM are much lower than those obtained by MLP.

The superior performance displayed by HHNM seems to be justified by its hierarchical topology. By encoding and memorizing efficiently context information, HHNM is capable of producing better predictions.

8 Conclusion

The paper presents a novel artificial neural model to the problem of long-term load forecasting. The model has a topology made up of two self-organizing map networks — one on top of the other —, and a single-layer perceptron. It encodes and manipulates context information effectively.

The novel hierarchical model is compared to a multilayer perceptron. Both models are endowed with time windows in their input layers. They are trained and assessed on load data extracted from a North-American electric utility. No pre-processing is made on data apart from ordinary normalization.

The experiments show that the performance of the hierarchical model both on long-term peak-load and mean-load forecasts is much better than that of the multilayer perceptron. The superior performance displayed by the model seems to be justified by its hierarchical topology.

The results achieved by the hierarchical model still have space for improvements. Fine adjustments on its parameters — map sizes, radius of the Gaussian, and radius of the neighbourhood — as well as the usage of pre-processing techniques on load data will certainly lead to improvements.

Finally, it is worth mentioning that multilayer perceptrons have been widely employed in short-term load forecasting so far. The results achieved may thus suggest that the hierarchical model may offer a better alternative to approach the problem of load forecast in general.

Acknowledgment

This research is supported by CNPq and FAPEMIG, Brazil.

References

1. Makridakis, S., Hibon, M.: The M3-Competition: results, conclusions and implications. International Journal of Forecasting **16** (2000) 451–476
2. Remus, W., O'Connor, M.: Neural networks for time-series forecasting. In Armstrong, J.S., ed.: Principles of Forecasting: a Handbook for Researchers and Practioners. Klüwer, Massachusetts (2001) 245–256
3. Kangas, J.: On the Analysis of Pattern Sequences by Self-Organizing Maps. PhD thesis, Laboratory of Computer and Information Science, Helsinki University of Technology, Rakentajanaukio 2 C, SF-02150, Finland (1994)
4. Chappell, G.J., Taylor, J.G.: The temporal Kohonen map. Neural Networks **6** (1993) 441–445
5. Barreto, G.A., Araújo, A.F.R.: Time in self-organizing maps: An overview of models. International Journal of Computer Research, Special Issue on Neural Networks: Past, Present and Future **10** (2001) 139–179
6. Kremer, S.C.: Spatio-temporal connectionist networks: A taxonomy and review. Neural Computation **13** (2001) 249–306
7. Barreto, G.A., Araújo, A.F.R., Kremer, S.C.: A taxonomy for spatiotemporal connectionist networks revisited: The unsupervised case. Neural Computation **15** (2003) 1255–1320
8. Zhang, B., Dong, Z.: An adaptive neural-wavelet model for short term load forecasting. Electric Power Systems Research **59** (2001) 121–129
9. Kim, C., Yu, I., Song, Y.H.: Kohonen neural network and wavelet transform based approach to short-term load forecasting. Electric Power Systems Research **63** (2002) 169–176
10. Marin, F., Garcia-Lagos, F., Joya, G., Sandoval, F.: Global model for short-term load forecasting using artificial neural networks. IEE Proceedings Generation, Transmission & Distribution **149** (2002) 121–125
11. Mori, H., Itagaki, T.: A fuzzy inference neural network based method for short-term load forecasting. In: Proceedings of the International Joint Conference on Neural Networks, IEEE (2004)
12. Reis, A.J.R., da Silva, A.P.A.: Feature extraction via multiresolution analysis for short-term load forecasting. IEEE Transactions on Power Systems **20** (2005) 189–198
13. Khotanzad, A., Afkhami-Rohani, R., Maratukulam, D.: ANNSTLF – artificial neural network short-term load forecaster – generation three. IEEE Trans. on Power Systems **13** (1998) 1413–1422
14. Hippert, H., Pedreira, C., Souza, R.: Neural networks for short-term load forecasting: A review and evaluation. IEEE Trans. on Power Systems **16** (2001) 44–55

15. Kermanshahi, B.: Recurrent neural network for forecasting next 10 years load of nine Japanese utilities. Neurocomputing **23** (1998) 125–133
16. Kohonen, T.: Self-Organizing Maps. Third edn. Springer-Verlag, Berlin (2001)
17. Galván, I.M., Isasi, P.: Multi-step learning rule for recurrent neural models: An application to time series forecasting. Neural Processing Letters **13** (2001) 115–133
18. El-Sharkawi, M.A.: Internet web page *http://www.ee.washington.edu/class/559/2002spr/* (2002)
19. Lo, Z., Bavarian, B.: Improved rate of convergence in Kohonen neural network. In: Proceedings of the International Joint Conference on Neural Networks. Volume 2. (1991) 201–206
20. Lo, Z., Fujita, M., Bavarian, B.: Analysis of neighborhood interaction in Kohonen neural networks. In: Proceedings of the Fifth International Parallel Processing Symposium. (1991) 246–249
21. Widrow, G., Hoff, M.E.: Adaptive switching circuits. *Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record, Part 4* (1960)
22. Sutton, R.S., Barto, A.G.: Toward a modern theory of adaptive networks: Expectation and prediction. Psychological Review **88** (1981) 135–170

Application of Radial Basis Function Networks for Wind Power Forecasting

George Sideratos and N.D. Hatziargyriou*

National Technical University of Athens
Greece
joesider@power.ece.ntua.gr, nh@power.ece.ntua.gr

Abstract. In this paper, an advanced system based on artificial intelligence and fuzzy logic techniques is developed to predict the wind power output of a wind farm. A fuzzy logic model is applied first to check the reliability of the numerical weather predictions (NWPs) and to split them in two sub-sets, of good and bad quality NWPs, respectively. Two Radial Basis Function (RBF) neural networks, one for each sub-set are trained next to estimate the wind power. Results from a real wind farm are presented and the added value of the proposed method is demonstrated by comparison with alternative methods.

1 Introduction

Wind energy generation has more than tripled all over the world the last five years. In 1999 the global installed capacity was 13600 MW and at the end of 2005 it was expanding to 59300 MW. The highest amounts of that capacity are concentrated in European countries. In 2004 the installed capacity in Germany was 16,629 MW, in Spain 8,263 MW and in Denmark 3,117 MW [1]. European countries account for more than 6000 MW new wind power capacity added in the year 2005, which represents an annual growth rate of 20%. This has come as a result of the European Commission's White Paper in 1997, setting the target to double the share of renewable energy in Europe from 6% to 12% by 2010. This meant that the wind power installed capacity would reach 40,000 MW until the year 2010, which could produce 80 TWh of electricity [1]. However, this target was already met in 2005 raising Commission's estimates for installed wind power 75,000 MW. Also, EWEA published an advanced scenario (Wind Force 12) that the global wind power penetration will amount the 12% of the overall electricity demand until the year 2030. In U.S. the current installed capacity is 9,500 MW, of which about 2,500 MW were installed in 2005 alone. AWEA expects that wind energy will grow to 6% of the US electricity supply by 2020 [2]. The Canadian Wind Energy Association targets to increase the installed capacity from the current 1000 MW to 10000 MW until the year 2010. In addition, the global energy demand is growing fast. The International Energy Agency (IEA) estimates that by 2030 some 4,800 GW of new power generation capacity will be needed. Wind energy can substantially help meet this demand [3].

* Senior member, IEEE.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 726 – 735, 2006.

The high growth of intermittent wind power penetration in electricity systems poses a number of challenges to the grid operators, who are called to manage this energy as efficiently as possible. Due to the dependence of wind power on the volatility of wind, extent fluctuations of wind farm output may increase costs for the electricity system and for the consumers and pose potential risks to the reliability of electricity supply. A priority of a grid operator is to anticipate the changes in wind power production, in order to schedule the spinning reserve and to manage the grid operations. Besides like Transmission and Distribution System Operators, different end-users, like Independent Power Producers and Energy Traders need wind power forecasting, in order to participate effectively in the Energy Markets. Persistence-type methods are frequently used; however such methods cannot provide satisfactory wind predictions.

A number of research efforts to provide an accurate wind power forecasting tool have been made. Depending on their input, these efforts are classified in physical or statistical approaches or a combination of both. The physical models use physical considerations, as meteorological (numerical weather predictions) and topological (orography, roughness, obstacles) information, and technical data from the wind turbines (hub height, power curve, thrust coefficient). Their purpose is to find the best possible estimate of the local wind speed and then use model output statistics (MOS) to reduce the remaining error. Statistical models use explanatory variables and on-line measurements and usually employ recursive techniques, like Recursive Least Squares or Artificial Neural Networks. Furthermore, Physical models must and Statistical models may use Numerical Weather Prediction (NWP) models. Models not using NWP might have good accuracy for the first 3-4 hours, but generally produce very inaccurate results for longer prediction horizons. Often, the optimal model is a combination of both, using physical considerations to capture the airflow in the region of the Wind Turbines and using advanced statistical modeling to supplement the information given by the physical models.

2 Reliability of Weather Predictions

The wind power production of a wind farm is straightforward dependent on the speed and direction of local wind. Therefore, in order to predict accurately the wind power production of a wind farm, any available information should be taken account. The main data used by the existing prediction tools are the numerical weather predictions (NWP). Usually NWPs consist of wind speed, wind direction and temperature. They come from meteorological models simulating atmospheric processes in order to estimate the atmospheric conditions in several levels. This simulation is accomplished firstly with the collection of measurements from synoptic meteorological stations and data from satellites. After erroneous data are removed, the information from the stations has to be processed. The performance of the meteorological models is dependent on this data assimilation and the quality of the measurements.

In order to estimate the future atmospheric processes and to predict accurately the wind speed in the area of a wind farm, the model represents the selected area as a grid. For each point of the grid the atmospheric state variables are calculated. The minimum spatial distance of these grid points is defined in order to obtain a stable

solution from the algorithm used. Reduction of this spatial resolution is a major challenge for the developed meteo models. High-resolution meteo models provide more accurate forecasts, however these models need long execution times to provide results and they need high computational capacity. For this reason the numerical weather predictions, which come from meteorological models, like HIRLAM, are updated only a limited number of times per day by meteorological services.

Concluding, the low density of the meteo stations, the low quality of their measurements, the incapability to reduce spatial resolution and the long execution times influence the performance of the meteorological models, especially in short-term predictions, e.g. up to 6 hours.

For this reason, a fuzzy logic based model is proposed in this paper that identifies NWPs with low accuracy. The fuzzy logic based model accepts as input the wind speed, the number of steps ahead (1-48 hours) and the ratio of the cumulative probability density functions of the last (known or predicted) value of wind power and of the wind speed value coming from NWPs, as expressed in (1):

$$\lambda = \frac{\Pr_{windpower}}{\Pr_{windspeed}} \tag{1}$$

where:

$$\Pr = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-(t-\mu)^2}{2\sigma^2}} dt \tag{2}$$

Pr is the cumulative probability density function
x is the sample for detection,
σ is the standard deviation of curve,
μ is the mean value of curve.

Each input (linguistic variable) has three fuzzy sets (small, medium, high). The fuzzy sets are defined by trial and error and the fuzzy model can be expressed by 27 rules of the type:

"IF x_i is A_i THEN y is B_i"

where:

x_i represents the input variable of the system
A, B are the fuzzy sets
y is the output of the model with a value between 0 and 1.

The fuzzy sets are modeled using Gaussian functions:

$$\mu_{Ai}(x_i) = \exp\left(-\left(\frac{x_i - a_i}{b_i} \right)^2 \right) \tag{3}$$

The ratio of the two probability functions operates as normalized correlation between the power measurements and the forecasted wind speed. The ratio fuzzy numbers are compared with the fuzzy numbers of the wind speed, in order to detect

'poor' forecasts. For example if the ratio is high and the value of wind speed is low, it is concluded that this power could not be produced by this wind speed. In the same way, the fuzzy model detects cases when the wind speed is high and the wind power low. Normally, the wind speed changes faster than the wind power and based on this fact, the wind power of the current time is compared with the wind speed of the next hour.

The last linguistic variable of the fuzzy model is the number of steps ahead and it is used to make more resilient the decision of the model for long-term horizons. For the first hour ahead, the fuzzy model, receives as input the last available wind power value and for the rest hours ahead it uses the previous wind power predictions. Due to the increase of the prediction uncertainty in longer term horizons, the impact of the fuzzy rules that determine the unreliable NWPs, at the fuzzy aggregation process is reduced for forecasts above 16 hours by 20% and above 28 hours by 40%.

The results of the fuzzy model for the first hour ahead are shown in figure 1. The system divides the data set into one set with good quality NWPs (Fuzzy logic model output above 0,5) and to another set with poor quality NWPs (Fuzzy logic model output below 0,5).

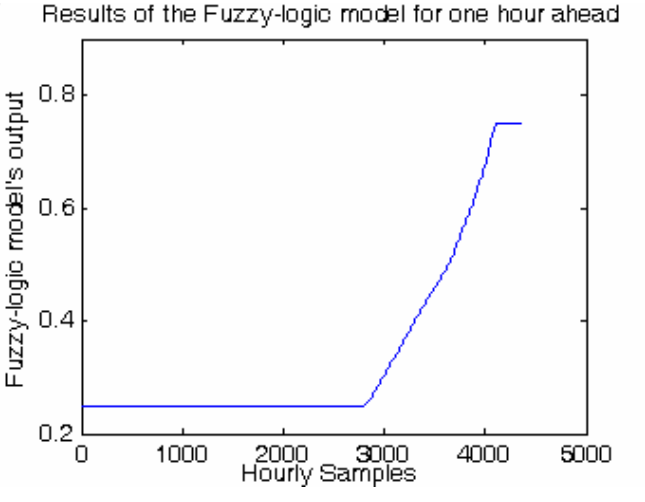


Fig. 1. Fuzzy model output

3 Wind Power Prediction Model

Radial basis (RBf) neural networks have been applied in the power system area for load forecasting [5, 6, 7] and other applications. In this paper RBfs are applied for wind power prediction. RBf networks are capable to give an acceptable solution for such a highly nonlinear system as the prediction of wind power owning to their structure that is characterized by a combination of non-supervised (in the hidden layer) and supervised (in the output layer) training. In the hidden layer a classification of the training set's samples to universes is accomplished and the kernels of these

universes (the most remote samples) consist of the weight matrix of the hidden layer. The second layer is linear and is trained thanks to the real power values (target vector). The characteristic function of an RBf network is Gaussian of the following form:

$$f(x) = e^{-x^2} \tag{4}$$

For the application of the wind power forecasting the neural networks are capable to capture all these parameters affecting the wind power estimation, like cases that the weather predictions are inaccurate. For this purpose two radial basis networks are trained, one only with the cases that the forecasted wind speed is considered unreliable by the fuzzy model of Section II and one with the cases the forecasted wind speed in considered reliable. Depending on the quality of the NWPs detected by the fuzzy model, the respective RBf network provides wind power prediction. Both RBf networks have the same structure consisting of 13 neurons in the hidden layer and both receive as input the same variables. Their only difference is that they are trained with different learning sets.

The RBf networks receive as input the most recent value of wind power, data from numerical weather predictions, such as wind speed and wind direction, and the hour that we make the prediction. Two values of wind speed provided by NWPs are used, which correspond to the hour wind power is predicted and to the next hour. The second wind speed value is applied to determine the tendency of wind to increase or to decrease. The input of each network has the following form:

I(t)= [P(t),WS(t+1),WS(t+2),WD(t+1),H(t+1)]

Where:

P(t) is the wind power production in MW for short-term horizons and in GW for long-term,

WS is the wind speed from NWPs,

WD is the wind direction from NWPs in “rad” and

H is the hour that prediction is made.

The architecture of the implemented RBf networks is shown in figure 2. The networks have two layers. The neurons of the first hidden layer have been produced by the classification of the training set. For each testing sample, that is evaluated, its Euclidean distance from the weighted input matrix is calculated and the result passes through the Gaussian function f (4). The output of the hidden layer has the following form:

$$a_i = f(\sum_{j=1}^{n} (IW_{i,j} - I_j)^2 * b \ ) \tag{5}$$

IW is the weighted matrix and b the bias; I is the input vector; n is the size of the input vector and i is the number of neurons of the hidden layer. So, the output of the hidden layer is dependent on the kernels (neurons) of the first layer, which are closest

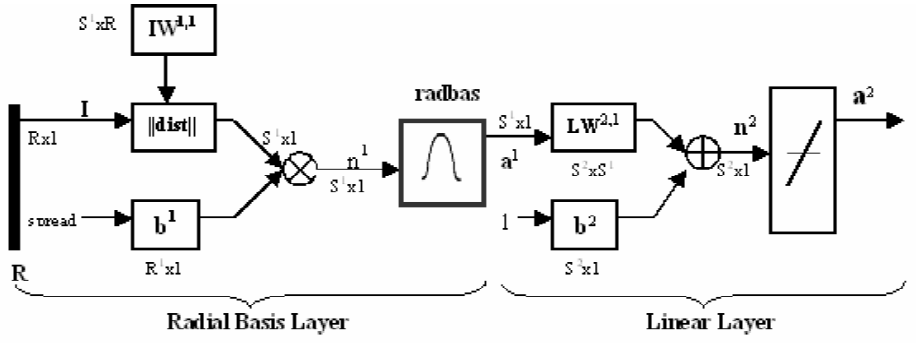


Fig. 2. RBf architecture

to the testing sample. The final result of the network is obtained from the superposition of the output of the hidden layer to the linear equation in the second layer.

An RBf characteristic that makes this neural network suitable for a wind power forecasting application is that its performance can be improved by normalization of the input variables, as follows:

$$P(t+1)=f(P(t),WS(t+1)*20,WS(t+2)*10,WD(t+1),H(t+1)/480) \quad (6)$$

P(t) is the wind power production in MW
WS is the forecasted wind speed in m/sec,
WD is the forecasted wind direction in "rad" and
H is the hour for which prediction is made.

4 Results

Wind power prediction of an actual wind farm in Ireland is presented. The farm contains 25 wind turbines and is located in the northwestern part of Ireland (Donegal County) 370 m above the sea level in complex terrain. The power production is measured in the period from 1st August 2002 to 31st March 2003. The time series cover a period of 5830 hours from which 4200 were used for training and 1630 for testing. Irish HIRLAM forecasts with spatial resolution 0.2° longitude/latitude have been used at level 30. They are updated four times a day, every six hours and cover a 48 hours horizon [8]. The developed forecasting method is able to operate with numerical weather predictions from different systems.

Due to the complexity of the terrain where the wind farm is located, the wind speed estimation is very difficult and the respective numerical weather predictions are in many cases very inaccurate [9]. However the proposed system performs acceptably, as shown next.

In the following figures the system performance error normalized by the installed capacity of wind farm P_{nom} is shown. The results are compared with the results obtained from the Persistence method. Persistence is a simple method, which

considers that the wind power production remains the same in all look-ahead times, as in the present time and is used as benchmark [10,11]. Figure 3 shows the normalized mean absolute error of the model and of persistence as a function of look-ahead times, expressed as:

$$NMAE = \left[\frac{1}{N} \sum_{i=1}^{N} e_i(t+k/t) \right] / P_{nom} \tag{7}$$

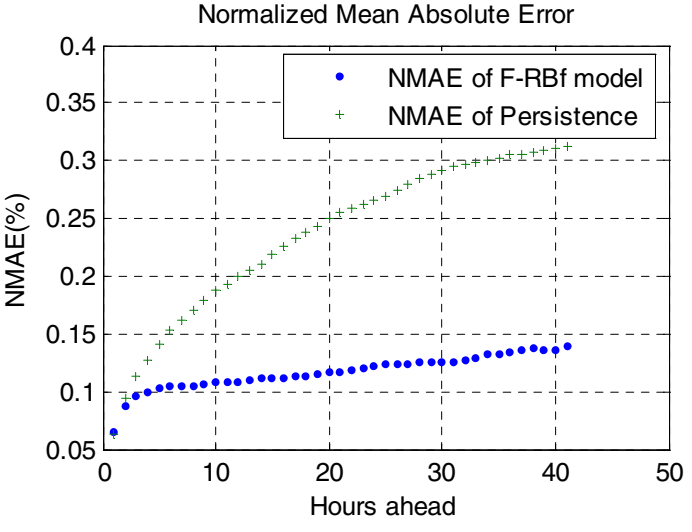


Fig. 3. The Normalized Mean Absolute Error of the proposed model and of Persistence for various look ahead times

Figure 4 shows the normalized root mean square error of the model and of persistence as a function of look-ahead times.

$$NRMSE = \left[\frac{1}{N} \sqrt{\sum_{i=1}^{N} e_i(t+k/t)^2} \right] / P_{nom} \tag{8}$$

It can be seen in these figures that the proposed model performs better than persistence except of the first time step. This is very important, because 'physical' models start to have positive improvement with respect to persistence after 3-5 hours ahead. Regarding the NMAE criterion, the proposed model provides results with errors ranging between 5% and 14% for all look ahead times, while the persistence model provides results with errors in the range between 5% and 32%. Also the NRMSE criterion is always less than 20%, while the results obtained by Persistence reach 40%.

The following formula gives the Improvement or Skill of a model for look-ahead time k and the table 1 contains the skill for various look ahead times:

$$skill\ (k) = \frac{NMAE\ _{P}(k) - NMAE\ _{RBf}\ (k)}{NMAE\ _{P}(k)} * 100\ \% \tag{9}$$

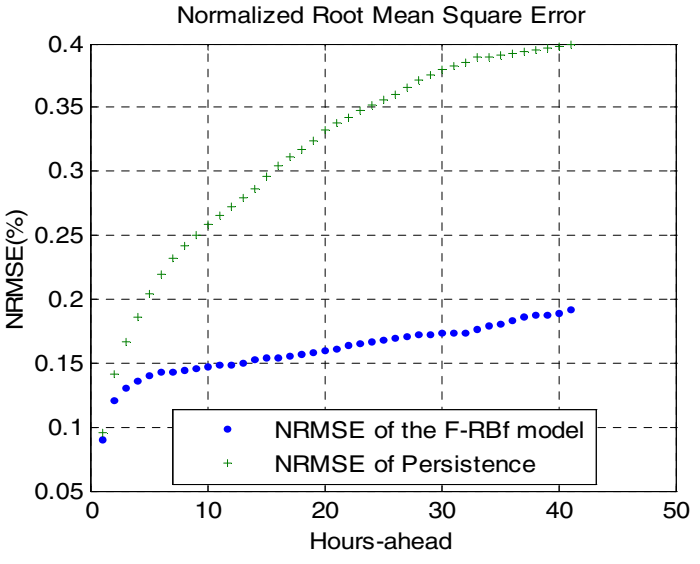


Fig. 4. The Normalized Root Mean Square Error of the proposed model and of Persistence for various look ahead times

Table 1. Improvement of proposed system with respect to Persistence for both criterions MAE and RMSE for various look-ahead times

Time horizon	Improvement	
	For NMAE criterion	For RMSE criterion
1 hour	-1.48%	5.64%
4 hour	21.6%	27.25%
8 hour	38.16%	40.34%
18 hour	52.1%	50.64%
32 hour	57.21%	54.93%
41 hour	55.25%	52.09%

Fig. 5 shows the improvement of the proposed model with respect to Persistence for each look-ahead time, which rises up to 57% for 32 hours look-ahead time for NMAE criterion. Also for both criteria the improvement of the model outreaches 20% for the forecasts after four hours ahead. The model's performance for the Irish study case is considered very satisfactory, if we take into account the complexity of the terrain and that the improvement remains above 40% after 16 hours ahead.

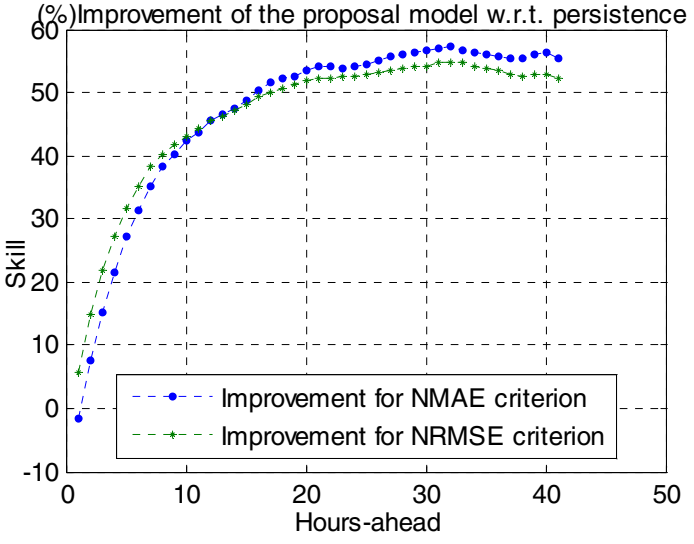


Fig. 5. Improvement of the proposed system with respect to Persistence for both criteria and various look ahead times

5 Conclusions

In this paper Radial Base Neural Networks have been applied to wind power forecasting using numerical weather predictions (NWPs). A fuzzy logic model is tuned in order to recognize erroneous wind speed forecasts. The proposal method divides the learning set to two subsets, one with accurate numerical weather predictions and one with 'poor' forecasts, depending on the fuzzy decision. Different Two similar RBf networks are used to learn separately the unreliable wind speed forecasts and the more reliable cases. Application of the method to a wind farm located in complex terrain shows clear benefits over the persistence method and the direct use of NWPs, especially after 4-6 hours, when forecasting of wind power production is most needed.

Acknowledgement

This work has been performed in the framework of the ANEMOS project NNE5-2001-00857. The authors wish to thank the EC DG Research for funding this work.

References

[1] "Wind power development in Europe", Hatziargyriou, N.; Zervos, A.; Proceedings of the IEEE, Volume: 89 Issue: 12 , Dec 2001, Page(s): 1765 –1782.
[2] IEA Wind Energy Annual Report 2003

[3] “Wind Energy: The Facts, An Analysis of Wind Energy in the EU-25”, EWEA, 2004
[4] “Wind Power Outlook 2004”, http://www.awea.org
[5] “Short-term electric power load forecasting based on cosine radial basis function neural networks: An experimental evaluation”, Nicolaos B. Karayiannis , Mahesh Balasubramanian, Heidar A. Malki ; International journal of intelligence systems, Vol.: 20, Issue: 9 Pages: 591-605
[6] “Extended Normalised Radial Basis Function for Short Term Load Forecasting”, E. Wadge, V. Kodogiannis; Proc (429) Modelling, Simulation, and Optimization, 2004
[7] “Short-Term Load Forecasting Using Radial Basis Function Networks”, Z. Gontar, G. Sideratos, N. Hatziargyriou, Proc. of SETN’04 Conference.
[8] “State-of-the-Art on Methods and Software Tools for Short-Term Prediction of Wind Energy Production”, G. Giebel, L. Landberg, G. Kariniotakis, R. Brownsword, Proc. of EWEC 2003, Madrid, Spain.
[9] “What performance can be expected by short-term wind power prediction models depending on site characteristics”, G. Kariniotakis, et al., Proc. of the EWEC04, London, UK, 22-25 November 2004.
[10] "A Protocol for Standardising the Performance Evaluation of Short-Term Wind Power Prediction Models", Madsen, H., Kariniotakis, G., Nielsen, H.Aa., Nielsen, T.S., Pinson, P., CD-Rom Proceedings of the Global WindPower 2004 Conference, Chicago, Illinois, USA, March 28-31, 2004
[11] "Towards Next Generation Sort-term Forecasting of Wind Power", Kariniotakis, G., & the Anemos Team CD-Rom Proceedings of the Global WindPower 2004 Conference, Chicago, Illinois, USA, March 28-31, 2004

The Application of Neural Networks to Electric Power Grid Simulation

Emily T. Swain, Yunlin Xu, Rong Gao,
Thomas J. Downar, and Lefteri H. Tsoukalas

School of Nuclear Engineering, Purdue University,
West Lafayette, IN 47907, United States

Abstract. A neural network approach is being developed to enable real time simulations for large scale dynamic system simulations of the electric power grid. If the grid is decomposed into several subsystems, neural networks can be utilized to simulate computationally intensive subsystems. An electrical generator sub-system was created in MATLAB using the SIMULINK interface. The SIMULINK model provided corresponding input/output pairs by varying parameters in sample transmission lines. A feed-forward backpropagation neural network was created from this data. Integration of the generator neural network into the SIMULINK interface was also performed. The original SIMULINK model requires about 342,000 iterations to simulate a 30 second simulation and consumes about 27 minutes of execution time. Conversely, the neural network based system is able to determine accurate solutions in less than 75 seconds and 300 iterations, which is more than an order of magnitude reduction in the execution time.

1 Introduction

Increased assessment of the national security of the United States has highlighted inadequate protection within the infrastructure of several large-scale systems, such as the electric power system grid [1]. The ability to simulate large-scale systems in a timely manner can provide an important contribution to national security. Real-time simulation of the electric power grid would increase response time during an event such that electrical disturbances could potentially be isolated.

The prospects of accurately simulating large-scale systems has improved due to advancements in technology and computing power. However, exact mathematical representations of large-scale systems include computationally demanding systems of differential equations. One method of reducing the computational burden would be to replace the exact equations with computationally less demanding neural networks.

Neural networks effectively decrease the required computational time of a system solution by utilizing pre-computed results to train a network describing a subsystem response, which can then be used in real time such that the overall system can be solved without the complete set of differential equations. The objective is to reduce the computational burden without sacrificing solution accuracy.

In the work here, an artificial neural network approach was utilized to simulate the response of an electrical generator to its load. The mathematical representation of an

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 736 – 745, 2006.

electrical generator involves solving several computationally costly differential equations at each time step. A neural network representing an electrical generator was integrated as a module into the SIMULINK simulation environment, which allows the module to be easily coupled to other components of an electrical grid system for more general simulation purposes.

1.1 Objectives

The objective of this project was to investigate the use of neural networks to achieve significant reductions in the computation time required for simulation of the electric power grid. Physics based modeling of an electrical generator system typically requires very long execution times. The work here was to perform a preliminary investigation into execution time reductions that could be achieved using neural networks with the ultimate goal of enabling real time power grid simulation.

A neural network approach approximates the response of an electrical generator to various load configurations. The network provides pre-computed states to a simulation system that originated from the physics based modeling of the generator. This computer logic system maps the neural network input to the appropriate output with minimal computational run time. Several simplifications were employed during the preliminary phase of this research. Although the physical modeling of the generator utilized time-dependent equations, the neural network modeling of the generator was initially developed assuming that only steady state simulations would be performed. This simplification eliminated the need for a complex, time-dependent neural network structure during the preliminary phase and decreased the amount of needed network training. The amount of data collection required from the physics based generator model is also decreased.

The real time electrical generator simulation interacts with its electrical load through the use of the simulation program SIMULINK. This object-oriented program allows straightforward integration of the real time electrical generator into a large-scale electrical system. The work here was also targeted for eventual integration into the well established Distributed Heterogeneous Simulation (DHS) [2]. This approach allows heterogeneous subsystems, such as an electric power grid system, a mechanical system, and a thermal system, to be solved simultaneously with detailed, dynamic results. These subsystems can also be solved in a geographically dispersed manner. Thus, the development of a real time generator as an interface could ease stability analyses as well as analyses outside the electrical system scope.

1.2 Methodology

A flow chart of the different processes that were completed during the course of this project is shown in Figure 1. The present phase of the project is to design a neural network based model to effectively replace the modeling of an electrical generator under steady state conditions which would provide accurate feedback to various load conditions but in a fraction of the time than is currently used.

The Original Model was developed within the simulation program SIMULINK. The model uses blocks located within the SimPowerSystems module library of SIMULINK. This model will provide a template for the development of the neural

network based model as well as provide a means of run time comparisons between the two systems. The Measurement Model was developed from the Original Model template and was modified such that steady state data could be obtained for each load configuration tested. These load configurations provided boundary conditions for the generator system. Neural network creation was performed after the collection of all necessary data was completed. The network could then be integrated into a neural network driven model within SIMULINK that could replace the electrical generator in the Original Model.

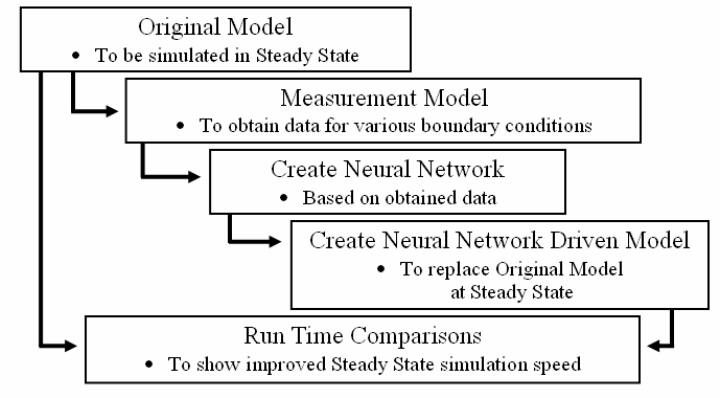


Fig. 1. Process Flow Chart

2 Project Preparation

2.1 Original Model

The Original Model was created to serve as a template for the development of the neural network simulation model. The goal of the neural network generator system would be to provide similar responses to the load as the original generator system in the Original Model, but would provide these responses much faster rate than the original generator system. The template simulation model was created from pre-built blocks with the SIMULINK library. The model consists of an electrical generator and its load. The pre-built generator blocks are based upon the set of complex differential equations shown in Figure 2 that are used to describe the physical actions of an electrical generator system. The pre-built load blocks are based upon the physical interactions of an example system of transmission lines.

The original physics based model is shown in Figure 3. The graphical representation of the model shows the organization and interactions of all the components. The generator system consists of a coupled system between the Synchronous Machine block and the Excitation System block. These blocks are located on the top of the figure. An example load was created to provide interactions between the load and generator. The example load consists of two Breaker blocks, a PI Section block, and AC Voltage Source blocks on each of the three phases. These blocks are located on the bottom of Figure 2.

$$V_d = R_s i_d + \frac{d}{dt}\varphi_d - \omega_R \varphi_q \qquad \varphi_d = L_d i_d + L_{md}(i'_{fd} + i'_{kd})$$

$$V_q = R_s i_q + \frac{d}{dt}\varphi_q + \omega_R \varphi_d \qquad \varphi_q = L_q i_q + L_{mq} i'_{kq}$$

$$V'_{fd} = R'_{fd} i'_{fd} + \frac{d}{dt}\varphi'_{fd} \qquad \varphi'_{fd} = L'_{fd} i'_{fd} + L_{md}(i_d + i'_{kd})$$

$$V'_{kd} = R'_{kd} i'_{kd} + \frac{d}{dt}\varphi'_{kd} \qquad \varphi'_{kd} = L'_{kd} i'_{kd} + L_{md}(i_d + i'_{fd})$$

$$V'_{kq1} = R'_{kq1} i'_{kq1} + \frac{d}{dt}\varphi'_{kq1} \qquad \varphi'_{kq1} = L'_{kq1} i'_{kq1} + L_{mq} i_q$$

$$V'_{kq2} = R'_{kq2} i'_{kq2} + \frac{d}{dt}\varphi'_{kq2} \qquad \varphi'_{kq2} = L'_{kq2} i'_{kq2} + L_{mq} i_q$$

$$\frac{V_{fd}}{ef} = \frac{1}{Ke + sTe}$$

Fig. 2. Electrical Generator Equation Set

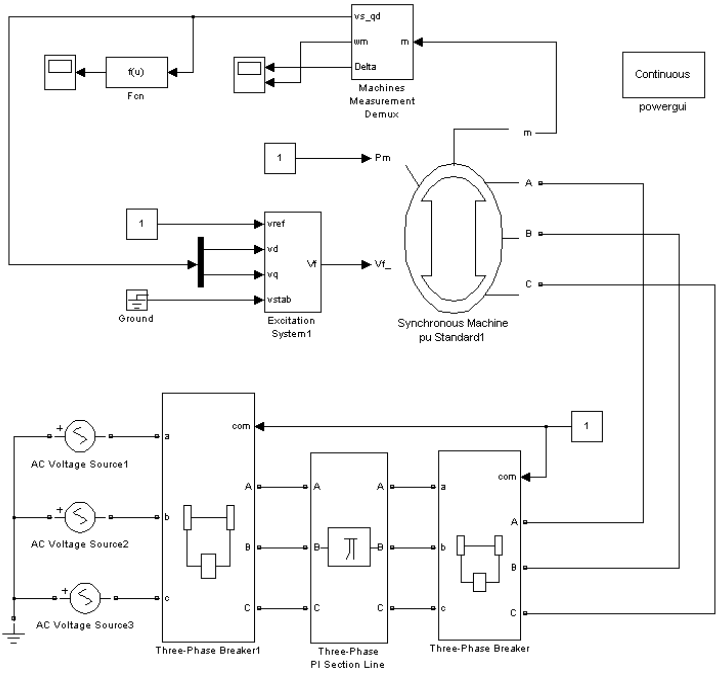


Fig. 3. Original Physics Based Model

2.2 Measurement Model

The Measurement Model was created to obtain data from the Original Model template. The obtained data was used to develop the neural network portion of the real time generator. The Measurement Model shown in Figure 4 contains many of the same components as the Original Model. For instance, the Generator Subsystem shown on the far left side of the figure contains the same generator components and parameters as described in Section 2.1. The Load Subsystem on the upper right of the figure also contains the same load components as previously described.

In addition to the components found in the Original Model, the Measurement Model contains some additional subsystems. The Current Measurement Subsystem (top-middle of Figure 4) and the Voltage Measurement Subsystem (bottom-right of Figure 4) provide the means to collect the appropriate data from the simulation. The Voltage Measurement Subsystem compares the voltage between ground and the generator load on each phase of the system. The voltage measurement data from each phase is then stored to a file during each simulation run. Thus, the file will contain the voltage magnitude and angle for Phase A, B, and C at each time step specified by the system parameters. The Current Measurement Subsystem measures the current passing between the generator and the load on each phase of the system. The current measurement data from each phase is then stored to a file during each simulation run in a similar fashion as the storage of the voltage measurement data.

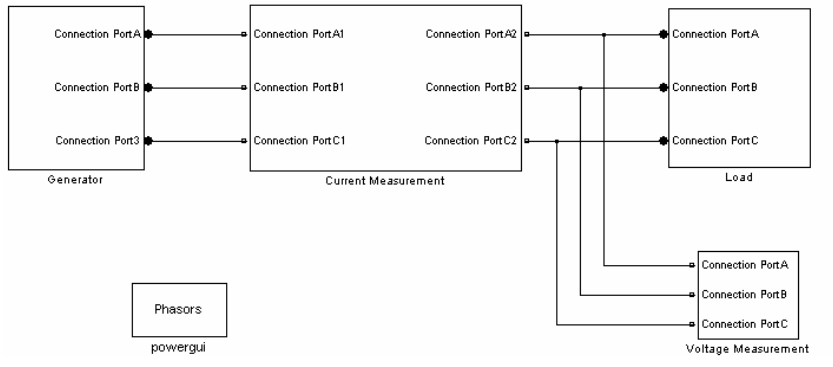


Fig. 4. Measurement Model

2.3 Data Collection

Data was collected from variations of the Measurement Model for use in training of the neural network. Variations of the Measurement Model were used to simulate a variety of boundary conditions for the electrical generator. Component parameters within the generator load were varied to produce different responses from the generator. Data was collected across a large distribution of parameters within the load.

The parameters of the Measurement Model were specified such that accurate steady state data could be collected. The solver utilized here, ode23tb, solved the problem using a variable time step selection method such that appropriate step sizes were used based upon changes in the previous steps of the simulation. The measurements collected from these simulations were to represent the system in a stable, steady state situation. Most simulations that were capable of a stable system reached steady state within a 30 second simulation. However, longer simulations were required for some boundary condition settings. A 30 second simulation using the above configurations typically required about 27 minutes of execution time and about 342,000 time steps on a 531 MHz computer when storing data to the two measurement files. A parametric study of various load configurations produced voltage

measurements within a range between 0.990 volts and 0.998 volts where stable solutions could be obtained. Over 160 different data points were collected in this range.

3 Solution System

The simulation of the electrical generator was performed using artificial neural networks which were able to interact and respond to variations in the electrical load. This interaction was achieved by placing the neural network within a SIMULINK model. A MATLAB script was also needed to perform an iterative search for the correct voltage and current values. All three of these components interact with each other to accomplish the simulation.

The Master MATLAB Script serves as the direct interface between the user and the solution system during a simulation. The major purpose of the Master Script is to serve as the primary command and control of the solution system. The secondary purpose of the script is to perform an iterative search for the appropriate voltage values for a given load set-up and return the final solution to the MATLAB command window. The SIMULINK model provides a platform for the neural network to directly interact with the generator load. The model performs simulations between the network and the load, as well as provides measurement values. These measurements provide feedback to the Master MATLAB Script such that the next iteration step can be determined. The SIMULINK model incorporates a neural network through the use of a neural network block. Any neural network created in MATLAB can be transformed into a SIMULINK block. The block can then be directly incorporated into the SIMULINK model, just as if it were a standard library block.

3.1 Master MATLAB Script

The principal purpose of the Master MATLAB Script is to provide solution system command and control. This included initiation and termination of the solution system, as well as controlling the proper sequence of events within both SIMULINK and MATLAB. The secondary purpose of the script is to execute the portion of the script devoted to an iterative search of the final solution. The search directly iterates voltage values for each of the three phases until convergence is met.

The command and control functions of the Master Script are key to the proper execution of the solution system. The command and control functions include the following duties: give control to the SIMULINK model, load MATLAB Workspace variables into the SIMULINK model, perform the SIMULINK simulation, overwrite Workspace variables with SIMULINK output determined during the simulation, and return control to the Master MATLAB Script.

The Master MATLAB Script is also used to perform an iterative search for a solution. This iterative search is executed by supplying the specified SIMULINK model with a “next guess” for each iteration step. Feedback from previous SIMULINK simulations of the model is used to calculate this guess. For instance, the “next guess” voltage values supplied to the model are compared to the measured voltage values provided by the simulation. Once the errors between these two values meet the

specified convergence tolerance, the final solution is printed to the MATLAB command window.

3.2 SIMULINK Model

The SIMULINK model is used to simulate the interactions between a generator load and a neural network that has been trained to represent an electrical generator. The load used in this model is the same configuration that was used to collect data. The generator subsystem in this model, called the Steady State NN Generator subsystem, was specifically created for this project.

The Steady State Neural Network Generator Subsystem is shown in Figure 5. This subsystem contains all of the necessary components and interactions such that a neural network can interact with a load created in SIMULINK. The subsystem contains connection ports to the far right of the figure that connect the subsystem to the generator load. The generator subsystem contains the following four subsystems: Current Production, Voltage Signal Production, Script Interactions, and Signal Based NN Generator.

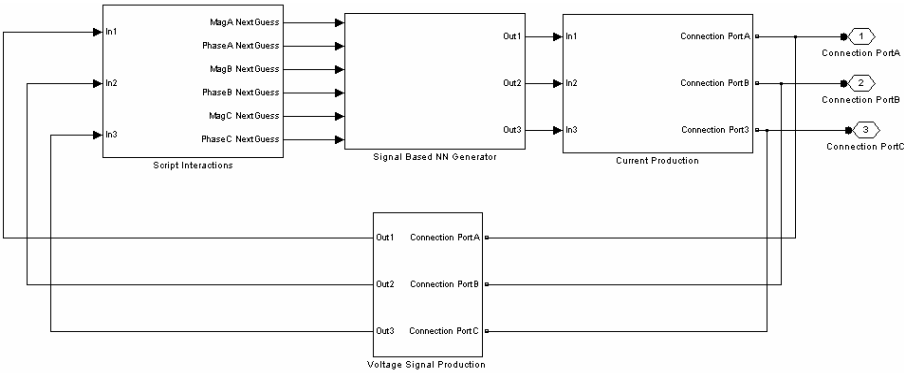


Fig. 5. Steady State NN Generator Subsystem

The Current Production Subsystem (top-right of Figure 5) contains three controlled AC current sources, one for each electrical phase. The sources produce an electrical current that is sent to the load. It is interesting to note that the use of electrical connections within the SIMULINK model only exist between the controlled AC current sources within the Current Production Subsystem, the Voltage Signal Production Subsystem, and the generator load. All other connections within the model utilize measured values to communicate information. Thus, one interpretation of the SIMULINK model is to consider the model to represent a three-phase controlled AC current source, its controller, and its load. This interpretation suggests that the neural network merely serves as a controller for the three AC current sources.

The Voltage Signal Production Subsystem (bottom-middle of Figure 5) contains voltage measurement blocks that determine the voltage between the generator load

and ground. The purpose of this subsystem is to provide the Master MATLAB Script with the necessary feedback to determine the next iteration step.

The Script Interactions Subsystem (top-left of Figure 5) provides the model a means of interaction with the Master MATLAB Script. One set of blocks provides the Master Script with data, while a second set of blocks obtains data from the Master Script. A natural break in the simulation loop occurs between the two sets of blocks within the Script Interactions Subsystem. This break is used by the Master Script to determine the "next guess" voltage values to be used in the next simulation.

The Signal Based NN Generator Subsystem (top-middle of Figure 5) utilizes the voltage input signals provided by the SIMULINK model in order to obtain appropriate current output signals through the use of a neural network. The components of this subsystem acquire the input values provided by the Script Interactions Subsystem, modify the input so that it can be properly utilized by the neural network, obtain output from the neural network, modify the output as necessary, and produce output signals to be utilized by other system components.

3.3 Artificial Neural Network

A fully connected, feed-forward backpropagation neural network was created to approximate the response of an electrical generator. The purpose of this neural network approximation is to allow simulations to be performed with less computational time but similar accuracy to the generator modeled in the Original Model. The basic equations of an electrical generator require computationally costly derivatives. Neural networks, however, require only multiplication and addition and thus greatly decrease the computational burden.

The supervised training of the network assumes that the SIMULINK generator found in the Original Model is exact, since the Original Model is the sole source of training and testing input/output pairs. The input to the neural network consists of the six components of the voltage measurements taken between the load and ground. These six components are the voltage magnitude and angle for each of the three phases in the system. The network output is the six components of the electrical current produced by the generator. These six components are the current magnitude and angle for each of the three system phases.

The neural network structure exploited for the electrical generator approximation is illustrated by Figure 6. Due to the nonlinear response of the electrical generator to its load, a hidden layer utilizing a hyperbolic tangent-sigmoid transfer function was employed. A hidden layer containing eight nodes minimizes the testing error. In addition, the outer layer utilizes a linear transfer function. This combination of transfer functions, which typically behave as a "general function approximator" [3], was capable of modeling the generator response with a high level of accuracy.

Accuracy was also improved by scaling the input such that all of the inputs into the network have similar values. The voltage angle values were normalized by dividing the values by 180 degrees. However, the voltage magnitude values did not require normalization since these values range between 0.9902 and 0.9982 due to the Original Model utilizing normalized power values. To further improve the accuracy and precision of the neural network system, some additional modifications were used. The small range of possible voltage magnitude values caused the neural network to

become overly sensitive to the voltage angle values. By increasing the range of possible voltage magnitude values, the neural network was modified to be equally sensitive to all six of the voltage measurement components. The magnitude range was increased by taking the natural logarithm of the voltage magnitude and then multiplying this value by a factor of 100. These scaling factors increase the voltage magnitude value range to between -0.9848 and -0.1801.

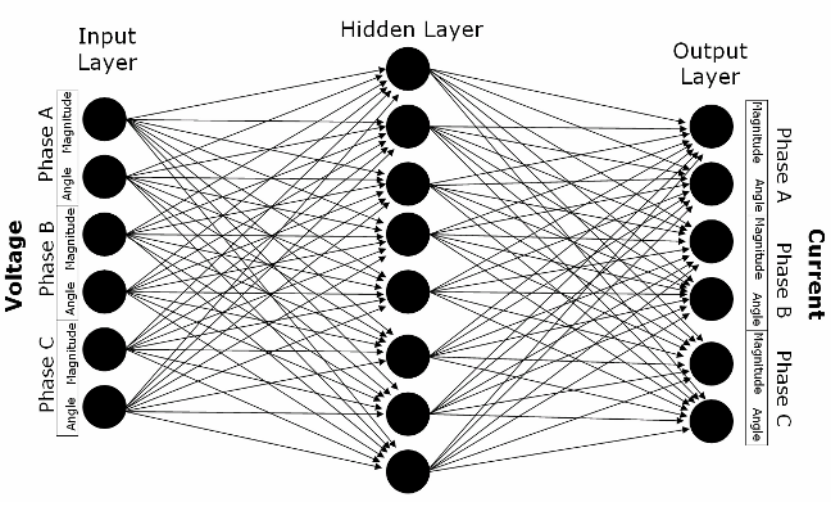


Fig. 6. Neural Network Structure

4 Results and Conclusions

The neural network was trained and tested utilizing data collected from the Measurement Model. This training resulted in the mean square errors shown in Table 1. Integrating the neural network into the solution system resulted in a maximum relative voltage error of 0.06% and a maximum relative current error of 0.23% as compared to the data generated using the Measurement Model. These results were obtained when the user-specified data within the Master MATLAB Script were the following: the initial maximum relative change is set to 0.01, the convergence tolerance is set to a relative voltage difference of 0.0004 between consecutive iteration steps, and the maximum number of iteration steps is set to 300 steps.

Table 1. Mean Square Error Between Testing Data and Trained Neural Network

Mean Square Error	Current Magnitude vs. Voltage Magnitude	Current Phase Angle vs. Voltage Magnitude
Phase A	2.27E-05	2.34E-05
Phase B	2.27E-05	0.000123
Phase C	2.27E-05	0.001586

The use of a neural network decreased the run time required to obtain a solution. The Measurement Model used to collect data required about 342,000 iterations to simulate a 30 second simulation and consumes about 27 minutes of execution time. These simulations were performed on a 531 MHz computer that was storing data to files using SIMULINK 5.0. Conversely, the neural network based system using SIMULINK 6.2.1 was able to determine solutions in less than 75 seconds and 300 iterations on a 2.16 GHz computer. Over half of the cases were capable of reaching a solution in less than 15 seconds.

The results here are encouraging and suggest that neural networks can provide significant relief in the computational burden without compromising the solution accuracy in power grid simulation. In the next phase of the work neural networks will be used to replace other parts of the system and the steady-state simulation will be extended to transient conditions.

References

1. Amin, M. "Infrastructure Security: Reliability and Dependability of Critical Systems." *IEEE Security & Privacy.* May/June 2005. pp. 15-17.
2. Hoffman, C., et al. "DDDAS For Autonomic Interconnected Systems: The National Energy Infrastructure." *2006 International Conference on Computational Science.* Manuscript submitted for publication.
3. Bernieri, A., et al. "Neural networks and pseudo-measurements for real-time monitoring of distribution systems." IEEE Transactions on Instrumentation and Measurement. Version 45, No. 2. April 1996. pp. 645-650.
4. Swain, E. "The Application of Neural Networks to Electric Power Grid Simulation" Unpublished master's thesis. Purdue University, West Lafayette, IN. May 2006.

Early Detection of Winding Faults in Windmill Generators Using Wavelet Transform and ANN Classification

Zacharias Gketsis, Michalis Zervakis, and George Stavrakakis

Electronic and Computer Engineering Dept. Technical University of Crete
73100 Chania Crete Greece, tel: (+30)-28210-37206, fax: (+30)-28210-37542
{zgetsis, michalis}@danai.systems.tuc.gr,
gstavr@electronics.tuc.gr

Abstract. This paper introduces the Wavelet Transform (WT) and Artificial Neural Networks (ANN) analysis to the diagnostics of electrical machines winding faults. A novel application is presented, exploring the potential of automatically identifying short circuits of windings that can appear during machine manufacturing and operation. Such faults are usually the result of the influence of electrodynamics forces generated during the flow of large short circuit currents, as well as of the forces occurring when the transformers or generators are transported. The early detection and classification of winding failures is of particular importance, as these kinds of defects can lead to winding damage due to overheating, imbalance, etc. Application results on investigations of windmill generator winding faults are presented. The ANN approach is proven effective in classifying faults based on features extracted by the WT.

1 Introduction

The aim of modern monitoring and diagnostic methods is to ensure the optimal and reliable utilization of motors or generators in respect to the outgoing power and their lifetime. In this regard several diagnostic methods are investigated and applied. Each method can be applied for a specific type of problem and has its own merits [1]. Insulation resistance measurement is useful for detecting cracked insulation that has e.g. absorbed moisture. Surge comparison test is useful for detecting areas where insulation has been removed or scraped from windings. A high-voltage test is performed to detect weak points in an insulation system, while winding resistance is mainly useful for checking the connections of windings. The frequency response analysis (FRA) of the transfer function or the winding admittance is used for windings fault detection [1]. The transfer function/admittance contains a number of peaks occurring at the natural oscillation frequencies resulting from the resonance between capacities and winding leakage inductance. The FRA enables us to detect faults that could not be detected by measuring just the winding inductance. Such faults usually result from the influence of electrodynamic forces generated during the flow of large circuit currents, as well as the forces occurring when the transformers are transported

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 746 – 756, 2006.

[2]. Of special importance is the early detection of winding failures, because this type of defects can lead to winding damage due to overheating, imbalance, etc [1].

However, there are some limitations in the interpretation and sensitivity using these methods for the detection of winding turn-to-turn faults. This class of faults is the most critical and frequently occurring failure during motor or generator manufacturing and operation. The research into the application of the WT and ANN method for the investigation of windmill generator windings aims to overcome such limitations in fault detection and classification and is, therefore, of high importance in practice.

In this paper, the WT followed by ANN classifiers are used to detect selected failures of windings in windmill generators. The WT is used for feature extraction, while the ANN is the tool for decision making and classification of the faults. Based on the work of [1, 2], we provide a fully automated approach for computing characteristic features of faults and classifying them. The influence of turn-to-turn faults between adjacent winding wires on the admittance is investigated. The proposed scheme has the ability to detect the generator winding fault and classify the type of fault with higher sensitivity and stability boundaries as compared to other techniques. Details concerning the design, implementation and testing of the proposed scheme are also presented. The required admittance curves are obtained using the numerical model of windings of electrical machines, which has been fully described in [1].

2 Numerical Model of Windings of Electrical Machines

Winding transfer functions (TF) are defined as frequency dependences of the ratios of respective currents or voltages at winding to the supply voltage U_e. In the case where the current I and the voltage U_e refer to the same winding, the transfer function represents the winding admittance. A substitute scheme of winding numerical model presented in Fig. 1 is used to obtain the admittance of electrical machine windings. It was constructed by replacing a section of the winding with the corresponding self and mutual inductance, capacitance to earth, longitudinal capacitance, insulation conductance and the resistance.

The winding admittance is defined as follows:

$$\mathrm{Y}(\mathrm{f}) = \frac{\mathrm{I}(\mathrm{f})}{\mathrm{U_e}(\mathrm{f})} \ , \tag{1}$$

where

$$\mathrm{I} = \mathrm{Z}^{-1}\left(\mathrm{T_L U} + \mathrm{T_{Lu}} \mathrm{U_e}\right) \ , \tag{2}$$

$$\mathrm{U} = -\left(\mathrm{Y} + \mathrm{T_L^t Z^{-1} T_L}\right)^{-1}\left(\mathrm{T_L^t Z^{-1} T_{Lu}} \mathrm{U_e} + \mathrm{Y_u U_e}\right) \ . \tag{3}$$

Y, Y_u, Z are matrices representing, respectively, the admittance and impedance of the system that are expressed using the following equations: $Y = \mathrm{j}\omega C + G$; $Y_{uz} = \mathrm{j}\omega C_u + G_{uz}$; $Z = \mathrm{j}\omega L + R$, while C, G, R, L are matrices of capacitances (C_c), conductances (G_c, g_c), resistances (R_c) and the matrix of self inductances (L_c) and mutual

inductances (M_c) between coils. T_L is the connection matrix and V_e is the vector of a sinusoidal supply voltage of a frequency f that changes stepwise within a specified frequency band. Here, the frequency was changed in the ranges from 0.1 kHz up to 6 MHz. Equations (2) and (3) represent the currents I in the inductance–resistance branches and the vectors of voltages U to earth in the nodes of the winding equivalent circuits. In the literature this method is often referred to as sweep frequency response analysis.

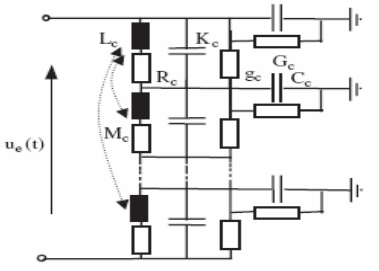


Fig. 1. The substitute scheme of windings of electrical machines

Our numerical simulations of the winding admittance of one phase of the electrical machine are based on the numerical model of machine windings presented in Fig. 1. The simulations were performed on Matlab 7 software. The winding parameters are calculated using the model presented in paper [1] as well as experimental data in [3] for short circuit conditions. In our experiments, each section of the winding model is composed of one winding coil [1].

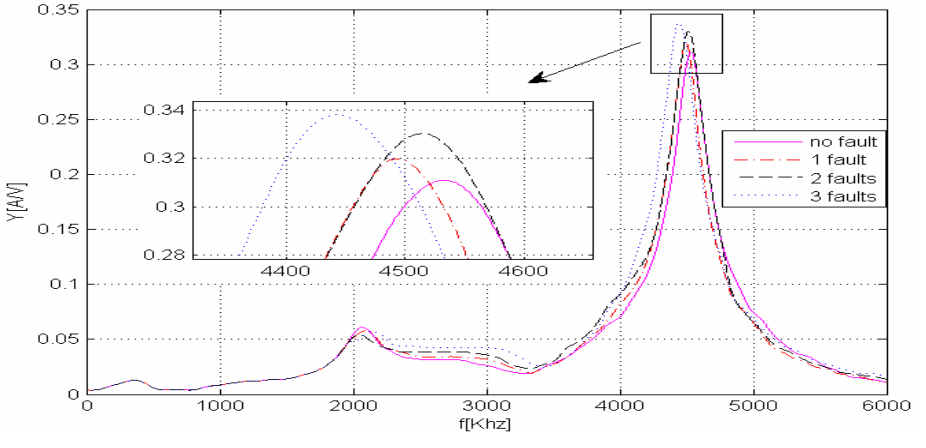


Fig. 2. Theoretical admittance curves of the winding of a windmill generator after successive stages of failure: 1) winding admittance before failure (no fault), 2) winding admittance after short circuit of two (1 fault), 3) three (2 faults), 4) four neighbouring turns (3 faults)

The results of numerical simulations of the admittance of a damaged windmill generator coil are shown in Fig. 2. The figure shows the frequency dependence of admittance of the winding without fault as well as the curves calculated for the winding after faults resulting from short circuit of neighbouring conductors. 3 types of fault are investigated: short circuit of two, three and four neighbouring turns of the winding. The shapes of the theoretical admittance curves are similar to the experimental ones, presented in [1], considering actual short circuits on experimental machines. The location of shorted turns in the coil is random; therefore the changes in maximum admittance values and the frequencies at which they appear are not regular. Based on this we simulate various short circuit instances by allowing a random perturbation of up to 6% in maximum amplitude and up to 2% in the location of this maximum, as shown in Table 2. Notice that the experiments in [1] with actual machines revealed a perturbation of mutual inductance between consequent damage states of up to 2%.

3 Introduction to the Wavelet Transform

Traditional Fourier analysis, which deals with periodic signals and has been the main frequency-domain analysis tool in many applications, fails to describe the eruptions commonly existing in transient processes as in winding faults. Since the Fourier Transform (FT) gives only frequency information of a signal, time information is lost. The approach widely as windowed FT or short-time FT (STFT) has been developed to deal with this problem. However, the STFT has the limitation of a fixed window width. Thus, it does not provide good resolution in both time and frequency.

Wavelets, on the other hand, provide higher resolution in time for high frequency components and higher resolution in frequency for low frequency components of a signal. In a sense, wavelets use a window that automatically adjusts in duration to provide the appropriate resolution. Wavelet analysis is based on the decomposition of a signal according to *scale,* rather than frequency, using *basis* functions with adaptable scaling properties. This method of analysis is generally referred to as *multiresolution analysis.* A wavelet transform expands a signal not in terms of infinite duration sinusoids but by projecting on *wavelets,* generated using the *translation* (shift in time) and *dilation* (compression in time) of a fixed wavelet function. For many signals, the low-frequency content is the most important. It is this content that gives the signal its identity. The high-frequency content, on the other hand, reveals detail-signal information, as in the case of faulty machine operation. In wavelet analysis, we often speak of approximations and details. The approximations are the high-scale, low-frequency components and the details are the low-scale, high-frequency components of the signal. DWT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into coarse approximation and detail coefficients. These coefficients represent different frequency subbands. This ability allows accurate fault detection.

DWT is used for the extraction of frequency features from the winding admittance signal by decomposing the signal into multiple frequency subbands. All wavelet

transforms can be specified in terms of a low-pass filter h, which satisfies the standard quadrature mirror filter condition:

$$H(z)H(z^{-1})+H(-z)H(-z^{-1})=1\ , \tag{4}$$

where $H(z)$ denotes the z-transform of the filter h. Its complementary high-pass filter can be defined as:

$$G(z)=zH(-z^{-1})\ . \tag{5}$$

A sequence of filters with increasing length (indexed by i) can be obtained and expressed as a two-scale relation in time domain:

$$\begin{aligned} h_{i+1}(k)&=[h]_{\uparrow 2^i} * h_i(k)\ ,\\ g_{i+1}(k)&=[g]_{\uparrow 2^i} * h_i(k)\ , \end{aligned} \tag{6}$$

where the subscript $[\cdot]_{\uparrow m}$ indicates the up-sampling by a factor of m, and k is the equally sampled discrete time. DWT employs two sets of functions, called scaling functions $\psi_{i,l}(k)$ and wavelet functions $\varphi_{i,l}(k)$, which are associated with low-pass and high-pass filters, respectively:

$$\begin{aligned} \varphi_{i,l}(k)&=2^{i/2}h_i(k-2^i l)\ ,\\ \psi_{i,l}(k)&=2^{i/2}g_i(k-2^i l)\ , \end{aligned} \tag{7}$$

where factor $2^{i/2}$ is an inner product normalization, and i and l are the scale parameter and the translation parameter, respectively. The discrete wavelet transform decomposition of a signal $x(t)$ can be described as:

$$\begin{aligned} s_{(i)}(l)&=x(k)*\varphi_{i,l}(k)\ ,\\ d_{(i)}(l)&=x(k)*\psi_{i,l}(k)\ , \end{aligned} \tag{8}$$

where $s_{(i)}(l)$ and $d_{(i)}(l)$ are the approximation coefficients and the detail coefficients at resolution i, respectively. In Fig. 3, the details and approximation signal up to level 4 are represented. At each level, a detail and an approximation signal are reconstructed. In the next level, the detail signal is decomposed to new detail and approximation signal.

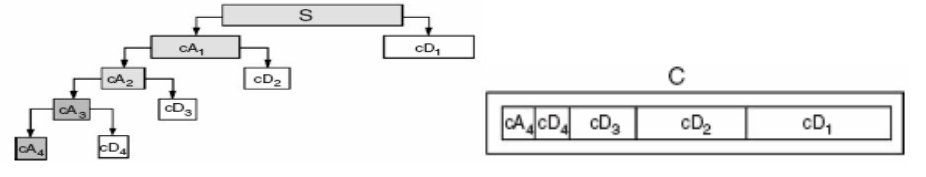


Fig. 3. Representation of four details and one approximation signal at the fourth level

4 Fault Detection and Classification Scheme

By using wavelet analysis, subband information can be extracted from the simulated curves, which contain useful fault features. By analyzing these features of the detail signals using Artificial Neural Networks, different types of fault can be detected and classified. The overall proposed scheme is shown in Fig. 4.

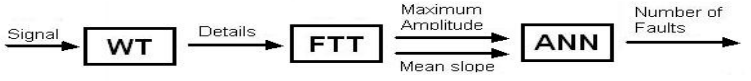


Fig. 4. Procedure of fault detection and classification scheme

4.1 Discrete Wavelet Analysis of Winding Admittance Signals

The choice of analyzing wavelets plays a significant role in fault detection and identification. The optimum wavelet maximize the cross correlation between the signal of interest and the wavelet. The wavelet functions we examine are: Haar, Daubechies 2, 4, Symlet 4 and Coiflet 3. The results for all 4 classes are shown in Table 1. It is obvious, that Daubechies 2 maximizes the cross correlation between the 4 signals and the wavelets, which is expected, since Daubechies 2 is localized, i.e. compactly supported, in time and so is appropriate for short and fast signals analysis [5]. Thus, Daubechies 2 is chosen in this scheme.

Table 1. Maximum cross correlation values between the signals and several wavelets

Wavelet	no-fault signal	1-fault signal	2-fault signal	3-fault signal
Haar	16.696	17.684	18.507	17.877
Daubechies 2	19.926	20.840	20.971	21.125
Daubechies 4	14.773	15.458	15.395	15.056
Symlet 4	17.457	18.320	18.652	18.353
Coiflet 3	16.305	17.160	17.291	16.901

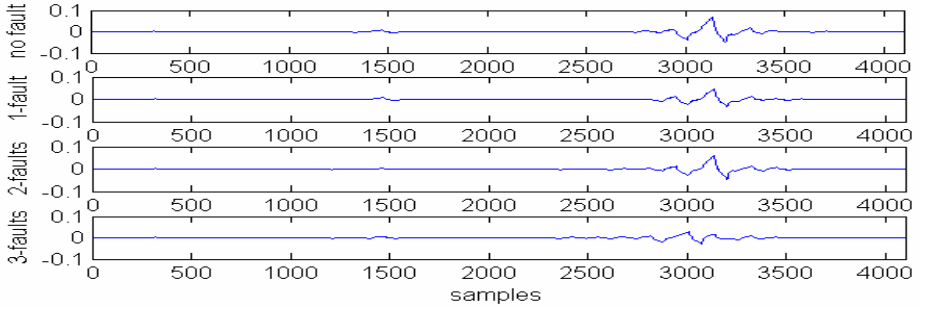


Fig. 5. Decomposition of winding admittance signals with 0, 1, 2 and 3 faults with 'db2' wavelet at level 7

Each original signal has 4096 samples and is decomposed into Daubechies 2 wavelet components at 12 levels ($4096 = 2^{12}$). When a fault occurs, different frequency components are produced. The wavelet level to be selected must reflect the fault characteristics under various fault conditions. In this respect, according to the analyses of different wavelet levels of the admittance curves, the level 7 ($D7$) detail is utilized to extract useful features. The detail wavelet coefficients at level 7 of the signals obtained from control windmill generators and generators with 1, 2 or 3 winding faults are given in Fig. 5. The horizontal axis is the number of samples, whereas the vertical axis is the amplitude. It can be seen that when a fault occurs, the spikes of $D7$ occur at different frequency and have different amplitude.

4.2 The Fast Fourier Transform of Winding Admittance Signals

After the decomposition of the signals, the Fourier Transform is applied to the details of the decomposed signals. The Fast Fourier Transform (FFT) can be used to simply characterize the magnitude and phase of a signal [6]. Notice that we only use the FT for feature extraction from the WT and not for signal characterization. Thus, the stationarity requirement on the signal is relaxed. The corresponding magnitude and phase of FFT of the detail 7 of the signals are shown in Fig. 6 and 7, respectively. For presentation convenience, only a small frequency range of the magnitude of FFT is presented in Fig. 6. These features (maximum of amplitude and mean of slope) are used to detect and classify different types of fault, as presented in next section.

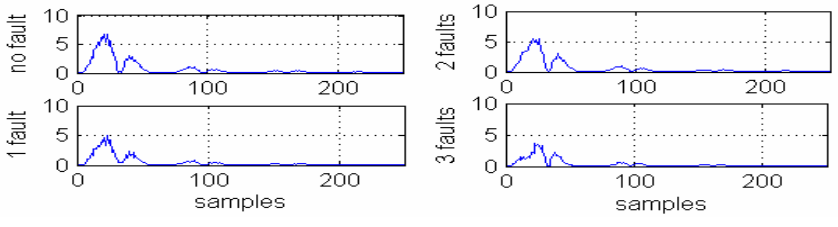


Fig. 6. FFT magnitude of the signals D7

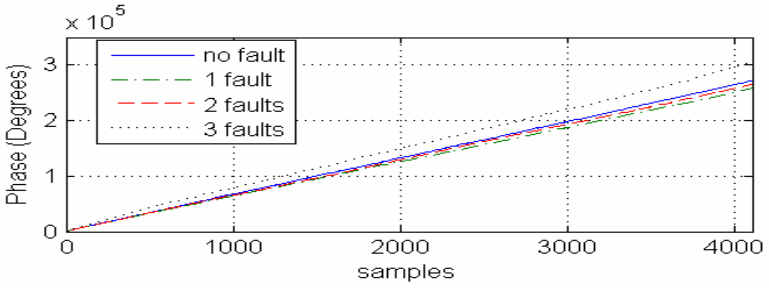


Fig. 7. FFT phase of the signals D7

4.3 Automatic Classification with ANN

The ANN is trained to classify signals into four categories: control (0-fault), 1-fault, 2-fault and 3-fault signals. ANN is an information processing system where information spreads in a parallel direction. It can determine its conditions and adjust itself to provide different responses by using inputs and desired outputs, which are provided to the system. The most important attribute of ANN is that it works as an expert system, which can eventually help the technicians with the decision-making process about the existence of the fault. ANN is trained with the available data samples to explore the relation between inputs and outputs, so that it reaches the proper output when presented with some new data [7].

In our approach, the multilayer feed forward ANN is implemented in the Matlab 7 environment. This choice is appropriate for solving pattern classification problems, where supervised learning is implemented with a Leverberg–Marquart (LM) backpropagation algorithm. The advantage of using this type of ANN is the very fast testing of new data, almost in real time, which is particularly advantageous in signal processing applications. Applications in the literature demonstrate the suitability of ANNs in detecting faults of transient signals, when ANNs are trained satisfactorily [7].

In this study, LM backpropagation neural network is used for the interpretation of admittance waveforms. ANN underdoes supervised learning to perform successful pattern recognition of the signals. During supervised learning, ANN is trained on input vectors with target output vectors and through its interpolation ability it is able to correctly classify previously unseen input vectors. The network is iterated for single hidden layer with combinations of one to 10 neurons. For each layer combination, the target mean square error is set to 0.001 and the epoch number is taken as 100.

The data set consists of 150 control signals, 150 signals with 1 fault, 150 signals with 2 faults and 150 signals with 3 faults. Each signal is a vector of the maximum amplitude and the mean slope of the phase of FFT of wavelet level 7 details, as seen above. The deviations of the signals of each class are uniformly distributed around the mean of each state, as presented in Table 2.

Table 2. The deviations of maximum amplitude and mean phase-slope of FFT

	Maximum Amplitude	Deviations (%)	Mean Slope of Phase (degrees)	Deviations (%)
Control state	6.8042	± 05.75 %	1.1600	± 01.41 %
1-fault state	4.9384	± 05.68 %	1.1002	± 01.32 %
2-faults state	5.4990	± 05.10 %	1.1292	± 01.28 %
3-faults state	3.6203	± 05.78 %	1.3042	± 01.52 %

For the ANN classifier we use a 3-fold cross validation scheme with stratification. The data set is split in 3 approximately equal partitions. Sampling is random in such a way as to guarantee that each class is properly represented in every partition. 2/3 of the data is used for training, while the remaining is used for testing. The whole procedure is repeated 200 times. The overall error rate is the average of error rates on

each partition. Each time, the training input data set consists of 100 control and 100 signals from each fault case, while the test data set is made of 50 control and 50 signals from each fault case. The minimum training and testing errors are accomplished with the combination of a hidden layer consisting of 10 neurons. For activation in the hidden layer, we use 5 non-linear functions: tag-sigmoid (tansig), log-sigmoid (logsig), triangular basis (tribas), satlins and radial basis (radbas). In the output layer we use a linear function.

5 Results and Discussion

In the present study, a new method was developed for the automatic classification of the signals based on DWT and ANN. First, the winding admittance signals were decomposed into details and approximation coefficients using DWT, as seen in section 4.1. Second, we obtained the magnitude and the phase of the FFT wavelet details coefficients, as seen in section 4.2. The maximum value of the magnitude and the mean slope (gradient) of the phase of the FFT of DWT detail coefficients level 7 for 150 control states and 450 fault states (150 of each type of fault) were calculated and used to train and test the ANN, using 3-fold cross-validation with stratification.

The backpropagation ANNs that built from 5 different non-linear neurons and trained with LM have demonstrated to provide excellent fault detecting and classifying results. In Table 3, the average success rate (ASR) in the training and testing stage, the average specificity (ASPE) (a measure of the ability of the classifier to accurately specify states, i.e. to retrieve only the correct samples from each state), the average sensitivity (ASEN) and the average detection rate (ADR) for 200 experiments for each type of hidden layer activation function are demonstrated. The end results are classified as 0 (no fault) and 1, 2 and 3 faults. It is obvious that the tag-sigmoid activation function provides the best results. A 95.72 % success rate of classification was accomplished with the designed feature extraction and the neural network structures. In Fig. 8, the specificity per experiment for tansig function is demonstrated. Exploring the results we can deduct some interesting conclusions. The sensitivity seems to be the same for all functions but specificity differs. The decision boundaries with some activation functions extend form one class distribution into the other, reducing its specificity. Furthermore, the success rate estimated through cross validation accurately reflects each classifier's performance, being in fact a little under-estimated in training.

Table 3. Results for each activation function

	tansig	logsig	tribas	satlins	radbas
ASR in training	95.59%	93.73%	90.16%	87.06%	84.68%
ASR in testing	95.72%	94.14%	91.97%	87.25%	87.55%
ASPE	93.21%	90.11%	89.30%	82.39%	84.01%
ASEN	99.98%	99.98%	99.93%	99.91%	99.89%
ADR	96.60%	95.04%	94.62%	91.15%	91.95%

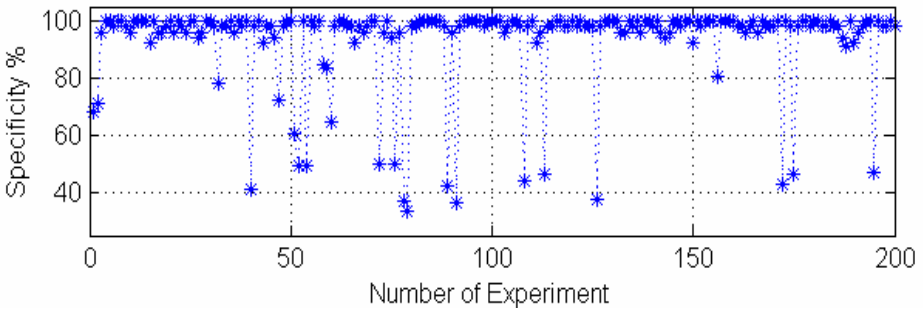


Fig. 8. Specificity Vs Number of experiment (tansig function)

The training and testing of the network demonstrate the accuracy of the proposed automatic classification method. While this paper provides a new and alternative automated method for the classification of winding admittance signals, it should be noted that the accuracy of this method is associated with the admittance recording and spectral analysis method applied for the training of ANN. Admittance is a non-stationary signal and so we use DWT analysis to get the most accurate results. Because the signals in this study are generated by a simulation model, employing real winding admittance signals measured by a digital recorder to improve the proposed method is one of our future plans.

6 Conclusion – Summary

The results of our investigations on detecting winding faults in windmill generators are presented. DWT is used for the extraction of frequency features from the winding admittance signal by decomposing the signal into details and approximation coefficients. Subsequently, ANN is employed for the automated classification of the signals.

Investigative features of the transfer function/winding admittance form the basis of the proposed method. The outputs of the admittance contain a number of peaks occurring at the natural oscillation frequencies resulting from the serial resonance between capacities and winding leakage inductance. The detection process is performed through signal decomposition and analysis of the decomposed components of the signal. The influence of turn-to-turn and inter-turn faults between adjacent winding wires on the admittance is investigated. The maximum value and the corresponding frequency of admittance resulting from these failures are analyzed. Based on proposed measures, the location and the number of winding faults are then possible to be detected with a simple, automated, very fast method (testing duration is approximately 0.15 seconds), with great accuracy and small computation cost.

The presented results demonstrate that the proposed method can provide an effective interpretation of machine performance, in terms of early fault detection.

Acknowledgements

This work was partly supported by HERPs Center, Leonardo da Vinci Project Ref: RO/04/B/P/PP 17 5006.

References

1. Florkowski M., Furgal J.: Detection of windings faults in electrical machines using the frequency response analysis method, Meas. Sci. Technol. 15 (2004) 2067–2074
2. Leibfriend T., Christian J., Feser K.: Transfer function method to diagnose axial displacement and radial deformation of transformer windings *IEEE Trans. Power Deliv.* 18 (2003) 493–505
3. Grandi G., Casadei D., Reggiani U.: Equivalent circuit of mush wound AC windings for high frequency analysis *Proc. ISIE Conf.* (1997) 201–206
4. Keppel G., Zedeck S.: Data Analysis for Research Designs-Analysis of Variance and Multiple regression/Correlation Approaches, New York: W.H. Freeman and Company, (1989)
5. Daubechies I.: The Wavelet Transform, Time Frequency Localization and Signal Analysis, *IEEE Trans. on Info. Theory,* Vol. 36, No. 5, (1990) 961-1005
6. Nawap S.H., Quatieri T.F. (Eds.): Short Time Fourier Transform, in: J.S. Limand, A.V. Oppenheim (Eds.), Prentice-Hall, Englewood Cliffs, NJ, (1988) 239–337
7. Kara S., Dirgenali F., Okkesim S.: Detection of gastric dysrhythmia using WT and ANN in diabetic gastroparesis patients, Computers in Biology and Medicine 36 (2006) 276–290

Combining Artificial Neural Networks and Heuristic Rules in a Hybrid Intelligent Load Forecast System

Ronaldo R.B. de Aquino[1], Aida A. Ferreira[2], Manoel A. Carvalho Jr[1], Milde M.S. Lira[1], Geane B. Silva[1], and Otoni Nóbrega Neto[1]

[1] Federal University of Pernambuco, Academico Helio Ramos s/n, Cidade Universitária, Cep: 50.740-530 – Recife – PE – Brazil
rrba@ufpe.br, macj@ufpe.br, milde@ufpe.br, geaneufpe@yahoo.com.br, otoninobrega@hotmail.com
[2] Federal Center of Technologic Education of Pernambuco, Av Professor Luis Freire, 500, Cidade Universitária, Cep:50.740-530 – Recife – PE – Brazil
aidaaf@gmail.com

Abstract. In this work, an Artificial Neural Network (ANN) is combined to Heuristic Rules producing a powerful hybrid intelligent system for short and mid-term electric load forecasting. The Heuristic Rules are used to adjust the ANN output to improve the system performance. The study was based on load demand data of Energy Company of Pernambuco (CELPE), which contain the hourly load consumption in the period from January-2000 until December-2004. The more critical period of the rationing in Brazil was eliminated from the data file, as well as the consumption of the holidays. For this reason, the proposed system forecasts a holiday as one Saturday or Sunday based on the specialist's information. The result obtained with the proposed system is compared with the currently system used by CELPE to test its effectiveness. In addition, it was also compared to the result of the ANN acting alone.

Keywords: Artificial Neural Networks, Hybrid System, Heuristic Rules, Electric Load Forecast.

1 Introduction

Recently, the Brazilian electric power system experienced important changes, regarding the administrative part as well as the planning part, its commercial regulation being a very complex issue. The changes in the regulation of the electric power market brought, as a consequence, an increase in competitiveness that was imposed by the decentralization of the distribution and by the growing demand on power quality required by the consumer market, resulting in an increasing search for improvements in the system planning.

The daily operation and planning activities of an electric utility requires the prediction of the electrical demand of its customers. Several researches have been carried out in order to improve planning and operation of these systems. Specifically, the required load forecasts may be divided into short, mid and long-term forecasts.

Traditionally, load forecasting techniques use statistical methods of time series analysis, which include linear regression, exponential damping and Box Jenkins [1].

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 757 – 766, 2006.

In recent years, techniques of artificial intelligence such as artificial neural network (ANN) have been used, obtaining promising results [2]-[6].

Currently, the procedure adopted by CELPE for hourly load forecasts is a mixing of statistical techniques with specialists' knowledge. The aim of this work is to improve the hourly load forecast, automating it and incorporating the implicit knowledge of the specialist. The developed system (named PREVER is implemented in MATLAB®) makes use of a hybrid approach of ANN based techniques and heuristic rules to adjust the short and mid-term electric load forecasting in the 3, 7, 15, 30, and 45 days ahead.

2 Data Base Configuration

The problem approached in this work is based on the hourly load forecasting in 3, 7, 15, 30, and 45 days ahead. The data used in this work were made available by CELPE and they correspond to the hourly load demand data in the period from January 2000 until December 2004.

All the data were unified in a single file, where each pattern was arranged by the information of the year, day, month and the load of the day for every hour (24 hours) and the day of the week to be forecast (Sunday, Monday,.... Saturday). The data regarding the more critical period of the rationing (from May to July of 2001) were eliminated from the file. The hourly load data were normalized (L_N) to fall in the range 0 to 1 by using (1):

$$L_N = \frac{L - L_{min}}{L_{max} - L_{min}} \ , \tag{1}$$

where L_N is the hourly load value registered by the CELPE's system, L_{max} and L_{min} are the maximum and the minimum hourly load value among all the observed values, respectively. In this work $L_{min} = 0$ and $L_{max} = 1.1 \cdot L_{A\_max}$, where L_{A_max} is the maximum value of the actual load data. The objective of factor 1.1 is to turn the values of future loads up to 10% above L_{A_max} into values below the unit after their normalization.

In this work, a holiday is considered by the specialist as one Saturday or one Sunday, according to [7]. In other words, the specialist indicates if the load behavior of that specific holiday is more correlated with the load behavior of Saturday or Sunday. Because the load curves of the holidays are close to the load curves of one Saturday or one Sunday, the hourly load data of holidays were just used in the test set. The training of Multilayer Perceptron (MLP) networks follows a paradigm of supervised learning, where each pattern in the training set is represented by an input and a desired output pairs. The patterns of the input set have the following arrangement: The first 24 values correspond to the hourly consumption of (***n+1***) days before the day to be forecast, the next 24 values correspond to the hourly consumption of ***n*** days before the day to be forecast (***n*** = 3, 7, 15, 30, or 45 days), and finally, the next 7 values define the day of the week that will be forecast (Sunday, Monday,.... Saturday). This information used *1-of-m* code.

The data base, formed by 969 examples of each period of time, is distributed in the following way: 60% for the training set, 30% for the validation set and 10% for the test set. The patterns of each group were selected in a random way.

The main objective of the load forecasting system based on ANN is to learn from pattern of known values and to generalize for new ones. The performance of the system will be measured by percentage of the mean-square error (MSE) [8] specified in (2), and by the mean absolute percentage error (MAPE) in (3).

$$MSE_{\%} = 100 \times \frac{L_{máx} - L_{mín}}{N \cdot P} \sum_{p=1}^{P} \sum_{i=1}^{N} (L_{pi} - T_{pi})^2 \ , \tag{2}$$

where $L_{\max}$ and $L_{\min}$ are the maximum and minimum of the hourly load values, in the representation of the problem, respectively; N is the number of output units of the ANN; P is the total number of patterns in data base; L_{pi} and T_{pi} are actual and desired target output of the *ith* neuron in the output layer, respectively.

$$MAPE_{\%} = \frac{1}{P} \sum_{p=1}^{P} \frac{\left| L_p - T_p \right|}{T_p} \times 100 \ , \tag{3}$$

where P is the total number of patterns in data base; Lp and Tp are the actual and desired output value for a given input, respectively.

The ANNs involved are designed using the method of training-and-testing. The basic idea of this method is to divide the set of patterns into three mutually exclusive subsets. The first subset is the training set used for computing the gradient and updating the network weights and biases. The second subset is the validation set. The error on the validation set is monitored during the training process to avoid overfitting. The third subset is the test set used exclusively for measuring the error of the system. The idea is that the performance of the system in a test set is its performance in real world. This means that no information on the test set can be available during the training [8].

Attempting to achieve an estimated error nearest to the true error, the *10-fold cross* validation method was chosen to generate the training, validation and test sets. This method has become a standard method in practical terms [9], [10]. Therefore, the patterns were divided in ten independent partitions, each partition having 10% of the data. For validation, in each experiment three partitions were used, one, to test and the six remaining partitions were used to train the ANNs.

3 Neural Network Structure and Training

All of the experiments accomplished in this work created ANNs with the MLP architecture, using the resilient backpropagation (RPROP) training algorithm [11]. The RPROP performs a local adaptation of the weight-updates according to the behavior of the error function. The RPROP algorithm operates in the batch mode and falls in a supervised training category.

All of the ANNs used have an input layer, a hidden layer and an output layer. The nodes of the hidden layer use the tan-sigmoid activation function and those of the output layer use the log-sigmoid activation function. The maximum number of iterations

for all of the trainings was set to 2500 epochs. The training stopped if the early stopping implemented by MATLAB® happened 20 times consecutively, or if the maximum number of epochs is reached, or if the error gradient reaches a minimum, or still if the error goal in the training set is met. The early stopping method has the objective of improving generalization of the neural networks. MATLAB® implements this technique, monitoring the error on the validation set during the training process.

3.1 Set-Up of the Hourly Load Forecasting Systems

All of the neural networks developed have 55 nodes in the input layer that are distributed in the following way: 24 nodes correspond to the hourly consumption of (n+1) days before the day to be forecasted, 24 nodes correspond to the hourly consumption of n days before the day to be forecasted (n = 3, 7, 15, 30, or 45 days ahead), and finally, the next 7 values define the day of the week that will be forecasted using *1-of-m* code (e.g. Sunday='1000000'). The output layer is characterized by 24 values, one for each hour of the day, which indicates the hourly load consumption of the day to be forecasted. Fig. 1 shows the basic layout of all networks.

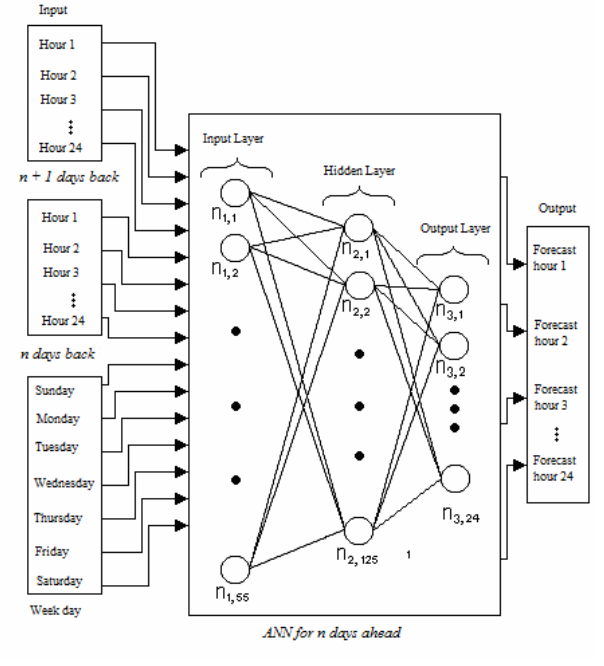


Fig. 1. Basic Layout of all Networks

To decide on the best configuration of nodes in the hidden layer in several days ahead, ten experiments were carried out with random initialization of weights and with varying number of hidden nodes from 30 to 130 with an increment of 5. The number of hidden nodes in the best neural network for the respective forecast horizon is presented in Table 1.

Table 1. Number of hidden nodes

Days ahead	3	7	15	30	45
Hidden Nodes	120	95	100	125	100

4 Heuristic Rules Development

In the area of expert system design, representations of heuristic rules have been extensively studied [12]. In this paper, these rules have been developed with the specific aim of reducing the error of the hourly load forecasting accomplished by the neural network. Two types of heuristic rules have been developed. In the first one (3 and 7 days ahead), the adjustment of the neural network output is made using the average and the standard deviation of the hourly historical load consumption. In the second one (15, 30, and 45 days ahead), the adjustment is accomplished by the average and the monthly historical load consumption.

Rule 1: Short-Term Load Forecast

This heuristic rule is used to adjust the ANN hourly load forecasting output for 3 and 7 days ahead. The adjustment is accomplished by evaluating the average of the consumption of the last 3 days which have the same characteristic as the prediction day and whose date are lower or equal to the difference between the date of the prediction day and the period of time. For instance, let the date of the prediction day be 24/04/2005 (Sunday) in the period of 7 days ahead. The average is computed in the following way: taking the prediction day minus the time periods gives 17/04/2005 (Sunday). As the prediction day is a Sunday, 3 previous consecutive Sundays should be taken before the date 17/04/2005. Thus, the average will be computed using the consumption of the days 17/04/2005, 10/04/2005 and 03/04/2005.

In the case of a normal day, the consumption is taken on the days that have approximately the same load curve as the prediction day. That is, to forecast a Monday, the average will be computed by the consumption of previous Mondays. For the case of holiday, the average will be calculated using Sunday or Saturday, which depends on how the specific holiday was registered in the system by the specialist.

After the calculation of the average hourly consumption, the standard deviation can be computed. In the next step, the upper and lower limits of the confidence interval are computed, where the upper limit is the average consumption plus the standard deviation and the lower limit is the average consumption minus the standard deviation. If the value of the neural network output is out of the confidence interval, its output is adjusted by the average, otherwise, it remains unaffected. This procedure makes the hourly load forecasting, by the neural network, fall in the confidence interval.

Rule 2: Mid-Term Load Forecast

This heuristic rule is used to adjust the ANN hourly load forecasting output for 15, 30, and 45 days ahead. Here again the consumption average was considered, and beside this, the consumption increase or decrease from one month to the next according to the historical seasonality.

Due to the lack of the month of the prediction day in the neural network input, it was necessary to add this information to create this heuristic rule, which was made by an increase or a decrease factor.

To find the increase or decrease factor, the daily load mean consumption was calculated, dividing the monthly consumption by the day's number of the month. Next the load behavior of one month to the next was analyzed. The factors were computed as a function of the time periods and can be summarized as follows:

15 days - The average daily consumption of the current month is divided by the previous one, and makes this result minus one. If the final result is positive, it means that there was an increase in the consumption of the previous month in comparison with current month, otherwise, there was a reduction. In the next step, the single monthly factor which corresponds to the average of the factor calculated previously in the periods from January 1994 until December 2004 was calculated. As the considered load forecast is 15 days ahead, then the monthly factors are divided by two.

30 days - The procedure to find the monthly factors for the load forecast in 30 days ahead is similar to that discussed in the 15 days ahead, except that the factors are not divided by two.

45 days - The mean diary consumption from the current month was divided by the penultimate month, reducing this result by the unit. In 45 days ahead, the calculation of the monthly factor is performed at two monthly intervals, since there are more than 30 days ahead. Thus, there are factors for the 60 days ahead. These values were calculated using the same analysis as in 15 and 30 days ahead. The monthly factors for 60 days ahead are the monthly average in the period. Finally, the monthly factor for 45 days ahead is the average of the factors between the monthly factor of 30 and 60 days.

After computing the factor, an algorithm was achieved so that the network output improves the hourly load forecasting. The rules are then used to adjust the output of the network.

The rules are based on the comparison between the hourly load forecasting by the neural network and a reference value. This reference value is given by the average of the last consumptions added to the portion of the monthly behavior, which corresponds to the multiplication of the monthly factor and the average of the consumptions. This average is the same average described in Rule 1, which is applied to 3 and 7 days ahead.

Finally, the Rule 2 can be stated:

- Positive monthly factor: If the simulated value by the neural network is smaller than the reference value, the neural network output will be adjusted to the reference value. Otherwise, it remains unaffected;
- Negative monthly factor: If the simulated value by the neural network is larger than the reference value, the neural network output will be adjusted to the reference value. Otherwise, it remains unaffected;

Specific Rules for January first and second

January 1, the first day of the New Year has the smallest load consumption of the year, in other words, it is a pattern different from the others kind of patterns presented

during the training process. On the other hand, January 2 suffers the consequence of January 1, and presents a low consumption in relation to the other days of January. So a specific rule for these days had to be made.

The forecast values to these two specific days were calculated as the above rules first, and then reduced from approximately 3%, in accordance with the knowledge of CELPE´s historical load.

It is important to point out that this new adjusted value be the value after the adjustment is accomplished according to the forecast horizon. For instance, if these days are being forecasted in 3 days ahead, the adjusted value will be the value of the network output after the adjustment in accordance with rule 1.

The system is implemented in the way that a user can accomplish his/her own adjustments manually for each day or hours of the day to be forecasted, similar to those made for January 1 and 2.

5 Performance

In order to verify the performance of the forecasting system on the load data pertaining to CELPE, forecasts were accomplished, short and mid-term, in the period from January until December 2005. However, just the results of the 45 days ahead will be compared here, because CELPE distribution utility carries out its load forecasting just in this period. This permits making a comparison of the developed system (PREVER) and the forecasting model currently being used by CELPE. Moreover, the performance of all the other forecast horizons was discussed in [7], [13] and presented here. Fig. 2 shows the graphs of the real consumption of the load forecasting system performed by CELPE, ANN and ANN plus adjustment (PREVER) in 45 days ahead.

After examining the curves of the normalized monthly consumption (Fig. 2), it can be seeing that the curve of ANN plus adjustment is the closest to the curve of the real consumption. This fact demonstrates the superiority of the hybrid system implemented in PREVER over the other systems.

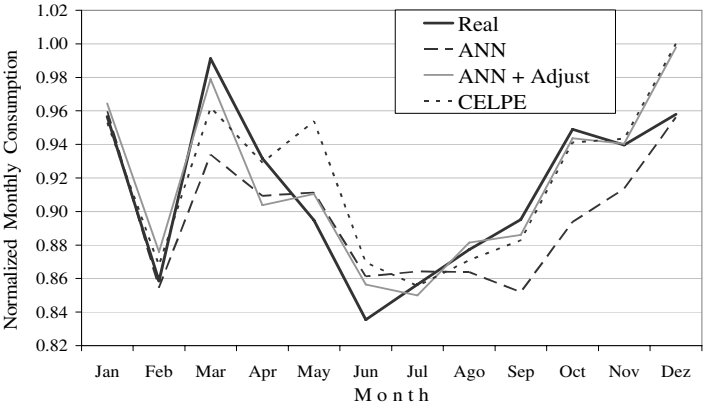


Fig. 2. Normalized Monthly Consumption in 45 days ahead

Table 2 shows the mean hourly MAPE for 3, 7, 15 and 30 days ahead. As shown the last row of this table, 3 days ahead was the only time period in which the mean hourly MAPE of all month corresponding to the ANN was slightly better than PREVER. In all other time periods, PREVER was far superior to ANN acting alone. This can also be observed in Table 3 for 45 days ahead.

Table 2. Mean Hourly MAPE in 3, 7, 15 and 30 days ahead

Mean Hourly MAPE								
	3 days ahead		**7 days ahead**		**15 days ahead**		**30 days ahead**	
Month	ANN	PREVER	ANN	PREVER	ANN	PREVER	ANN	PREVER
January	2.80	2.04	3.80	2.21	3.64	3.44	3.25	2.86
February	2.97	2.49	3.36	2.54	3.20	2.44	2.98	2.55
March	3.33	3.71	4.33	3.99	5.11	4.36	6.37	3.49
April	3.08	2.45	3.40	2.50	3.64	3.65	3.39	3.25
May	2.39	4.46	2.59	4.22	3.51	3.37	3.84	3.69
June	2.76	2.69	2.76	2.73	2.46	2.39	2.63	2.59
July	1.85	1.74	2.14	1.87	2.06	2.12	2.33	2.31
August	1.84	1.99	2.53	2.13	2.29	2.02	2.90	2.37
September	2.14	3.15	3.45	3.35	4.39	2.98	5.43	2.72
October	2.88	2.26	2.98	2.46	3.53	2.21	4.58	2.88
November	2.69	2.30	3.37	2.40	3.74	2.14	4.04	2.12
December	3.09	3.13	3.55	3.29	3.31	3.44	2.88	3.74
Mean	**2.65**	**2.70**	**3.19**	**2.81**	**3.41**	**2.88**	**3.72**	**2.88**

Table 3. Mean Hourly MAPE in 45 days ahead

Mean Hourly MAPE in 45 days ahead			
Month	ANN	PREVER	CELPE
January	3.09	2.55	3.32
February	3.43	3.23	3.28
March	6.13	3.45	5.13
April	3.74	3.75	2.96
May	2.95	2.89	7.32
June	4.04	3.55	5.24
July	2.18	2.32	2.72
August	3.15	2.55	2.98
September	4.90	2.52	3.16
October	5.90	2.27	2.35
November	3.50	2.49	2.49
December	3.59	4.63	4.88
Mean	**3.88**	**3.02**	**3.82**

Table 3 shows the mean hourly MAPE for the same system computed by ANN, PREVER and CELPE. In the period of 12 months from January to December 2005, only in April was the system of CELPE better then PREVER. Moreover, the mean hourly MAPE of all months for the PREVER system in the period was 3.02 against 3.82 presented by the CELPE system. Thus, the new system, PREVER, is certainly an improvement on the CELPE system.

To gain a better appreciation of the previous statement, Fig. 3 shows the mean hourly MAPE in each month for the system of CELPE, ANN and ANN plus adjustment (PREVER) in 45 days ahead.

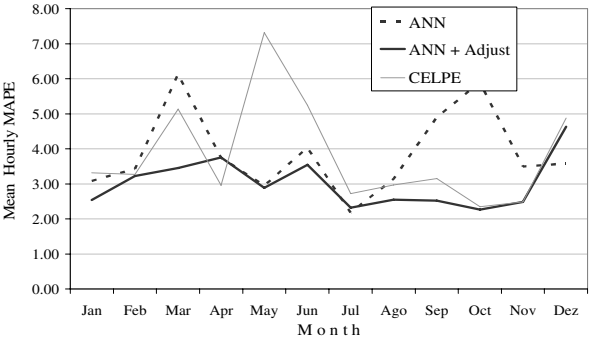


Fig. 3. Mean Monthly Error of MAPE in 45 days ahead

6 Conclusion

This work presents the final product of the research and development project between CELPE and DEESP/UFPE that resulted in short and mid-term load forecasting by software, named PREVER. Applying a hybrid intelligent system approach of ANN based technique and heuristic rules, this software is able to forecast the electric load of CELPE system in 3, 7, 15, 30, and 45 days ahead.

PREVER was evaluated in all forecast horizons mentioned previously. However, only the period of 45 days ahead could be compared with the currently CELPE´s load forecast system, because CELPE distribution utility carries out its load forecasting just in this time period. The results confirmed the potential and suitability of the hybrid intelligent system implemented in PREVER compared to CELPE´s load forecasting system and ANN acting alone. In the period from January until December 2005, PREVER was more precise than CELPE´s load forecasting system over 11 months.

References

1. D. C. Montgomery, L. A. Johnson, and J. S. Gardiner, “Forecasting and Time Series Analysis.” McGraw-Hill International Editions, 1990.
2. A. Bakirtzis, V. Petrldis, S. J. Kiartiz, and M. C. Alexiardis, “A Neural Network Short Term Load Forecasting Model for the Greek Power System,” IEEE Transactions on Power Systems, vol.11, no. 2, 1996, pp 858-863.

3. A. Khotanzad, R. Afkhami-rohani, D. J. Maratukulam, “ANNSTLF - Artificial Neural Network Short-Term Load Forecaster-Generation Three” IEEE Transactions on Power Systems, Vol. 13, No. 4, pp. 1413-1422, November,1998.
4. C. Kim, I. Yu, and Y. H. Song, “Kohonen Neural Network and Wavelet Transform Based Approach to Short-Term Load Forecasting,” Electric Power Systems Research, vol. 63, issue 3, 2002, pp. 169-176.
5. S. E. Papadakis, J. B. Theocharis, S. J. Kiartzis, and A. G. Bakirtzis, “A novel approach to Short-term Load Forecasting using Fuzzy neural networks,” IEEE Transactions on Power Systems, vol. 13, no. 2, 1998, pp. 480-492.
6. A. Silva, L. Moulin, and A. J. R. Reis, “Feature Extraction via Multiresolution Analysis for Short-Term Load Forecasting,” IEEE Transactions on Power Systems, vol. 20, no. 1, 2005, pp. 189-198.
7. R. R. B. Aquino, A. A. Ferreira, G. B. Silva, O. Nóbrega Neto, M. M. S. Lira, and J. B. Oliveira “Previsão de Carga Horária em Médio Prazo Utilizando Redes Neurais com Foco na Previsão dos Feriados,” VII CBRN – Congresso Brasileiro de Redes Neurais, ISSN: 1808-8589, Natal 2005.
8. L. Prechelt, “Proben1 – A Set of Neural Network Benchmark Problems and Benchmarking Rules,” Technical Report, 1994, pp. 21-94.
9. A. A. Ferreira, “Comparação de arquiteturas de redes neurais para sistemas de reconhecimento de padrões em narizes artificiais,” Master dissertation, Center of Informatics, Federal University of Pernambuco, Recife, 2004.
10. W. Witten and E. Frank, “Data Mining, Practical Machine Learning Tools and Techniques with Java Implementations,” Morgan Kaufmann Publishers, 2000, pp. 126.
11. M. Riedmiller and H. Braun, “A Direct adaptive Method for faster Backpropagation Learning: The RPROP Algorithm,” IEEE International Conference on Neural Networks, vol.1, April 1993, pp.586-591.
12. A. J. Gonzales and D. D. Dankel, “The Engineering of Knowledge-Based Systems,” Prentice Hall, 1993.
13. R. R. B. Aquino, A. A. Ferreira, L. H. Medeiros, G. B. Silva, and O. Nóbrega Neto, “Previsão de Carga em Curto e Médio Prazo Utilizando Redes Neurais Artificiais: Uma Aplicação ao Sistema CELPE,” SENDI - XVI Seminário Nacional de Distribuição de Energia Elétrica, Brasília 2004.

New Phenemenon on Power Transformers and Fault Identification Using Artificial Neural Networks

Mehlika Şengül, Semra Öztürk, Hasan Basri Çetinkaya, Tarık Erfidan

Kocaeli University, Engineering Faculty, Electrical Engineering Department,
41040 Kocaeli, Turkey
mehlika@isnet.net.tr,
{semra, cetinkaya, terfidan}@kou.edu.tr

Abstract. In this paper voltage recovery after voltage dip that cause magnetizing inrush current which is a new phenomenon in power transformers are discussed and a new technique is proposed to distinquish internal fault conditions from no-fault conditions that is also containing these new phenomenons. The proposed differential algorithm is based on Artificial Neural Network (ANN). The training and testing data sets are obtained using SIMPOW-STRI power system simulation program and laboratory transformer. A novel neural network is designed and trained using back-propagation algorithm. It is seen that the proposed network is well trained and able to discriminate no-fault examples from fault examples with high accuracy.

1 Introduction

Power transformers are the most important components in power system. Avoiding damage to power transformers is vital. In a power system, when continuity in power delivery is disrupted because of transformer fault, it may be necessary to repair or replace the transformer or other electrical equipments. So protection of power transformers is vital for continuity in power delivery.

Generally, differential relays are used for primary protection of large transformers. The technique is based on the measurement and comparison of currents at both sides of the transformer primary and secondary lines. The problem that may occur in a differential protection is to distinquish internal faults from magnetizing inrush current that may occur in non-linear working conditions like transformer energization, voltage recovery after clearing an external fault and change of the character of an external fault [1]. As such currents have no secondary winding counterpart, the differential relay is exposed to faulty operation.

The most common technique used for preventing false trips during energization is the harmonic restraint relay. The even harmonic component, especially the second harmonic component, is used to restrain the relay operation when the inrush current appears. The second harmonic component of the inrush current is considerably larger than in a typical fault current [2]. If the second harmonic content of the differential current exceeds a pre-defined percentage of the fundamental, inrush is assumed and the protection is prevented from tripping.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 767 – 776, 2006.

Schemes and studies based on the detection of the second harmonic were proposed and implemented in both analog and digital differential relays. In these studies various digital filtering algorithms were used for extract current signals to harmonic components. New numerical protection can also lead to unnecessary operation or operation failures. In addition to these methods, some algorithms reported in the literature use electro-magnetic equations of transformers. These algorithms use currents in the transformer windings to make appropriate decisions. Terminals of delta-connected windings are not usually brought out of the transformer tank, so the winding currents are not available for use in protective relays. This is the limitation of these algorithms.

The enormous capabilities of the artificial ANNs in non-linear mapping through a set of input/output examples are successfully employed to develop different types of protection schemes. The ANN approach work as a pattern classifier and is able to detect the changing power system condition quickly and accurately and consequently results in the improvement of the performance of conventional digital relays. Considering these factors, many researchers continued their work to develop new algorithms for transformer protection [3,4,5]. All these algorithms are either based on the transformer equivalent circuit model and/or some transformer data obtained by using EMTP.

This paper presents a new algorithm based on ANN for the protection of single-phase and three-phase transformers. This method is suitable for the two winding transformers with any type of connections. Proposed algorithm is based on the some transformer data obtained by using SIMPOW-STRI and laboratory transformer. Another difference of this study is training and testing data sets are containing new phenomenon that may cause false tripping of differential relay.

2 Simulated System

The simulated system was created in SIMPOW-STRI. The electrical system is composed of a 210 kV and 170 MVA generator, a 210:10.2 kV and 170 MVA three phase power transformer, transmission lines of different lengths and a 120 MVA load with 0,6 inductive power factor. The power transformer has a delta connection in the primary winding and a star connection in the secondary winding. The saturation of the core is considered. The basic scheme of system is shown in Figure 1.

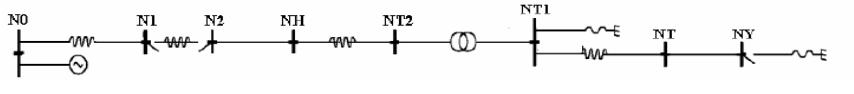


Fig. 1. Single line diagram of the simulated system

Bus N0 is slack bus; Bus N1-N2 is breaker; Bus N2-NH is transmission line; Bus NH-NT2 is series reactor; Bus NT2-NT1 is power transformer; Bus NT1-NT is series reactor; Bus NY is load bus.

The fault, linear and non-linear working conditions that power transformer may subject are analyzed using SIMPOW-STRI and MATLAB-SimPower simulation programs. In these conditions, primary and secondary currents are recorded. These currents are first sampled. Current signals were sampled at 1 kHz, which means 20 samples on

50 Hz power frequency. These currents samples are reduced to a lower level using ratio of current transformers and then differential currents are calculated. Using full cycle data window discrete fourier transform, currents are extracted to harmonic components.

2.1 The Case of Magnetizing Inrush

When circuit breakers interrupt the flow of current in a transformer, the core retains some residual flux. Later, when the transformer is reenergized, the core can saturate. If it does, the primary winding draws large magnetizing currents containing a large and long lasting dc component, is rich in harmonics, from the power system. This phenomenon is called as magnetizing inrush. Fig. 2 is a diagram of inrush current recorded in the SIMPOW program. The waveform is belonging to proposed transformer that is energized through short transmission line. The diagram is recorded at energization angle of 0° and at no-load condition.

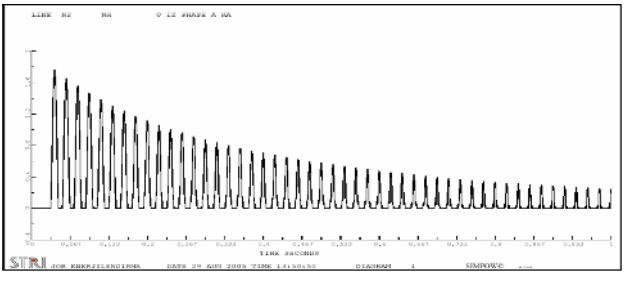


Fig. 2. The diagram of transformer inrush current

If sampled data sets that belong to diagram is analyzed it will seen that inrush current contain much harmonic component. According the analysis, inrush current contains the odd and even harmonic components. Diagram of harmonic analysis is shown in Fig.3a. Inrush current magnitude depends on several parameters. Some of them are: magnetic properties of the core material, remanence in the core, moment when a transformer is switched in etc. The most important one is point-on-voltage wave at the instant of energization. The change in magnitude of inrush current at different energization angles is shown in Fig.3b.

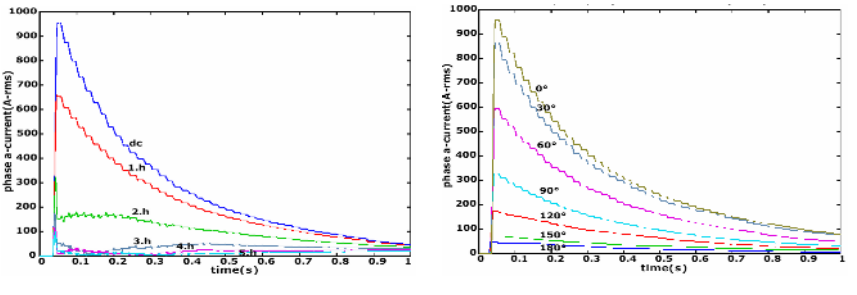
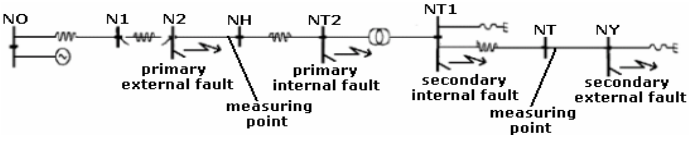


Fig. 3. (a)Harmonic components of inrush current.(b) Inrush current at different energization angles.

For getting training and testing data sets, the transformer that is fed from different transmission line lengths was energized different energization angles.

2.2 The Case of Internal and External Faults

The fault types generated in primary and secondary were single-phase to ground fault, double-phase to ground fault, phase to phase fault and three-phase fault. For each fault type, varying the fault start time varied the fault inception angle.



2.3 The Phenomenon of Magnetizing Inrush Current Caused by Voltage Sag

Voltage sags are the main cause of more than 80% of the problems experienced in power systems. It has been observed that voltage recovery after voltage sag can produce transformer saturation. This saturation produces an inrush current similar to that of the transformer energization.

Reference [6] describes the effect of symmetrical voltage sags on three-phase three legged transformers. It shows the effects, which are depth, duration and initial point-on-wave of sag duration, to the peak value of the inrush current.

2.3.1 Simulated System Results

Simulations were made based on proposed system that is explained in Section 2 in SIMPOW. In Fig.4 diagrams that were obtained using this program are shown. The simulation time is 1,5 sec. The transformer is energized at the instant 0,0128 sec., then it is loaded at the instant 0,6 sec. (120 MVA load with 0,6 inductive power factor) and voltage sag begins at the instant 0,8 sec. And ends at the instant 0,925 sec. Voltage sag magnitude is 50%.

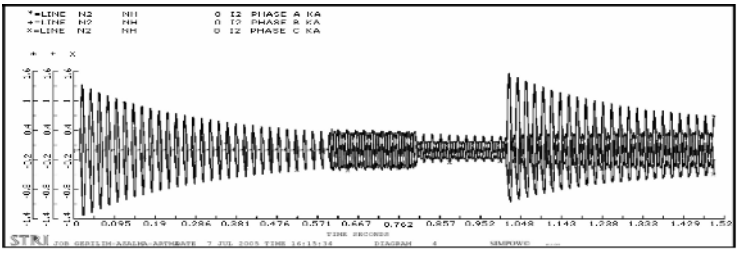


Fig. 4. Primary currents of the transformer in the case of voltage recovery after voltage sag

The effects of point-on voltage wave with changing of sag duration are examined for several voltage magnitudes. The start-point of voltage sag is taken, 0° of the sine wave and the duration of voltage sag is changed for one cycle. The effective values of current according to the duration of voltage sag are shown in Fig.5.

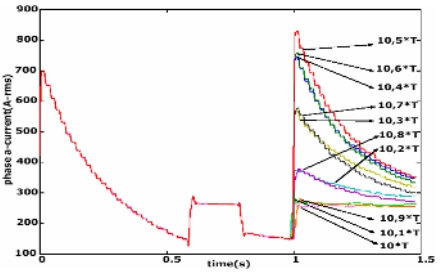


Fig. 5. Transformer phase-a current at different voltage sag duration (T, one period time)

It is seen that the highest value of inrush current occurs when the voltage sag duration is (n+1/2)T where n is real number. For getting training and testing data sets different voltage sag durations and depths are applied to the simulated system.

2.3.2 Test System Results

The transformer used for the test was 1kVA single-phase 240/120V. It is loaded resistively 30% of nominal power. Voltage sag is applied by a 4,5kVA programmable three-phase source with an integrated arbitrary waveform generator (4500LX California Instrument). The data sets are obtained by using a 200 MHz scope meter (Scope meter 199-C). It is seen in Fig.6.

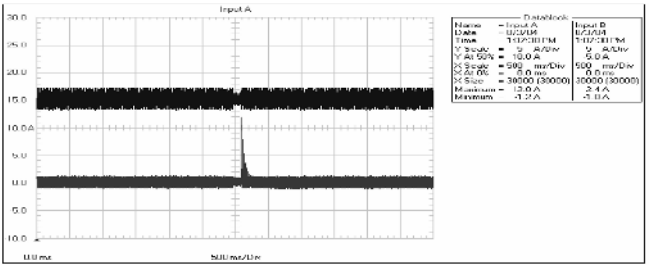


Fig. 6. Test transformer voltage and currents in the case of voltage recovery after voltage sag

3 Artificial Neural Network

The artificial neural network represents a parallel, multi-layer information processing structure that enables the inclusion of expert knowledge into processing, recognition and classification of signals.

Among the various artificial neural networks, the multilayer perception (MLP) can be considered as an information processing system whose function is defined from a set of examples describing both the inputs and the desired output. After a training step, the MLP is able to compute the right output not only from the input vectors of the examples set, but also from any unknown input vector [3,4].

We have used an improved method of error backpropagation (BP) learning algorithm. The learning process of BP neural network is a error correction learning

method. The process includes forward propagation and back propagation. In the forward propagating process, acted by node function, the input signals pass through the input layer and propagate to the hidden layer and output layer. The states of neurons at one layer only influence the next ones. If the expected results cannot get at the output layer, then the back propagation begins. The error signals are transmitted along with the coming paths for modifying the connecting weight coefficients between nodes to make the output error smallest.

3.1 Simulation Cases for Generating Training and Testing Data

The system simulated in order to get the pattern to train and test the ANN architecture was created in SIMPOW-STRI and MATLAB-SimPower System Toolbox. In both simulation programs saturable transformer was used. Since there is not saturable current transformer model in SIMPOW-STRI program, the internal faults followed by current transformer saturation condition was analyzed using MATLAB. Many simulation cases were investigated in this work: energization, overexcitation, normal, external fault, voltage recovery after voltage sag, parallel energization, internal fault, energization with internal fault, internal faults followed by current transformer saturation.

3.2 Network Architecture and Training

A set of 224 training cases (160 case of them for no-fault conditions and 64 of them fault-conditions) and a set of 159 testing cases (81 case of them for no-fault conditions and 78 of them fault-conditions) were used. Transformer primary and secondary current signals were sampled at 1kHz, which means 20 samples on 50 Hz power frequency. In each case the total of samples was limited to have 200 samples. By using sampled current signals, differential currents were calculated. By using full-cycle data window discrete fourier transform, differential currents were extracted to harmonic components. Inverse Discrete Fourier Transform was used to obtain the samples of harmonic components in time domain. Differential currents and the ratio of its second harmonic components to fundamental were taken as inputs of the ANN. All inputs were normalized to the input that had the highest magnitude. The output of the ANN is trained to respond “1” for no-fault current (no trip command) and “0” for fault current (trip command of the differential relay).

The MATLAB Neural Network Toolbox was used for generating network architecture. There is no particular formula to choose suitable network architecture for an application. The suitable network size is found by trial and error. Small sized network may not be enough to map the function, but bigger sized network may not be a better choice as well. By trial and error, it was found that the suitable network size for this system with 2 inputs and 1 output was a network with two hidden layers of size 5 and 3. In this work, tansig functions are used for both the hidden layers and linear function is used for output layer. The study was made using both sigmoid and linear transfer function in output layer. Since we used hardlim function in the output of the network there appear no differences on the results. It means that linear combination of hidden layer gives sufficient result for this work. The feed-forward backpropagation technique is used for training ANN. The criterion function for the sum square errors is minimised according to the gradient descent procedure. The proposed neural network architecture is shown in Fig.7.

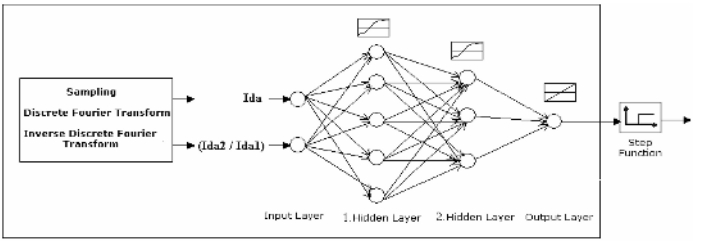


Fig. 7. Proposed neural network architecture

In Fig.8 the training inputs and target is given and the output shows that proposed network is well trained.

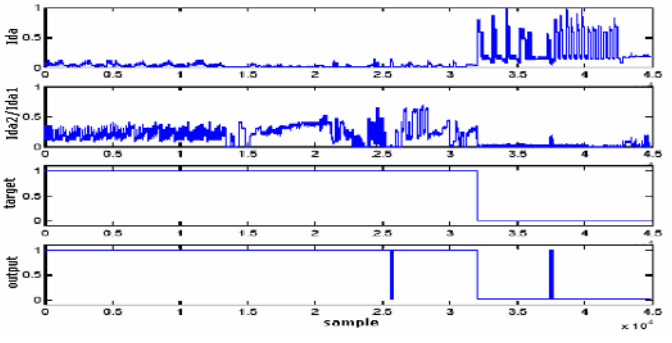


Fig. 8. Training inputs, target and output of the network

In Fig.9 the testing inputs is given and the output shows that proposed network is successful and it can follow the fault and no-fault cases very well.

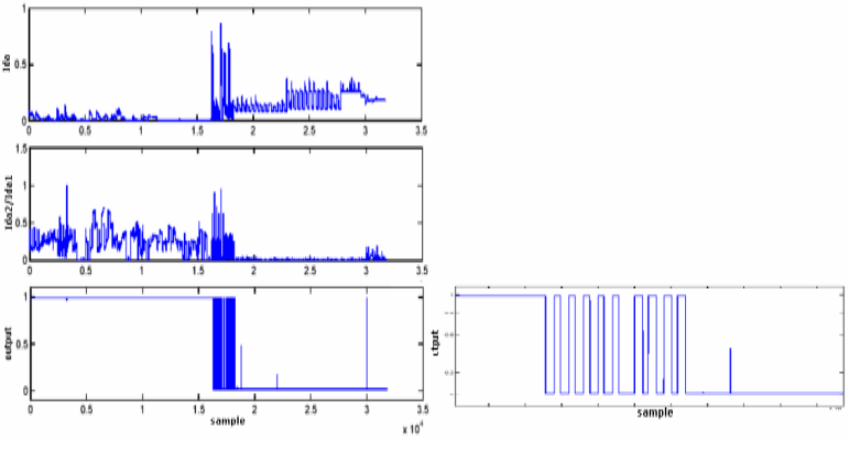


Fig. 9. Test inputs and output of the network

3.3 Laboratory Transformer Test Results and Network Response

In this section some practical simulation results and the response of proposed ANN are given. Showing the reliability of the proposed network and network adaptation to protect any transformer, laboratory tests were made. The system was described in section 2.3.2.

3.3.1 The Case of Primary Internal Fault

In this working condition internal short circuit between turns 1 and 104 was made in loaded test transformer. In Fig.10 the primary and secondary currents are shown.

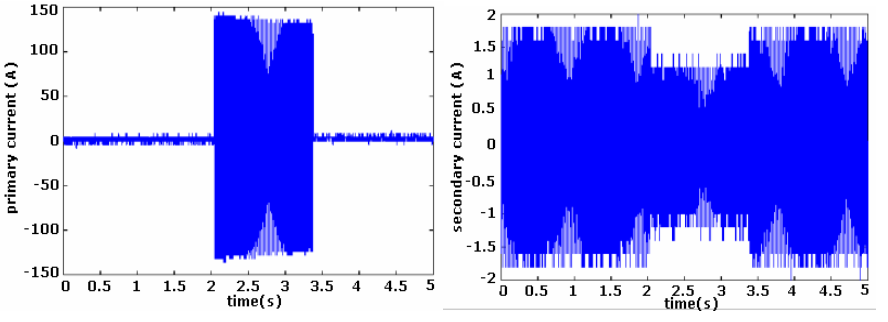


Fig. 10. Primary internal fault (between 1. and 104. turns)

In Fig.11 currents that belong to this condition and the response of the proposed network is shown. The proposed network operating time is about 0,002s.

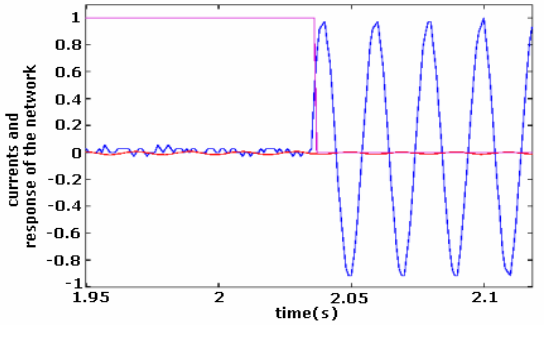


Fig. 11. Reduced primary and secondary currents and response of the network

3.3.2 The Case of Voltage Recovery After Voltage Sag

In this working condition 30% sag was created at the 0° of voltage and applied to loaded transformer. Voltage sag duration was applied during 5,5 periods. In Fig.12 the primary and secondary currents are shown.

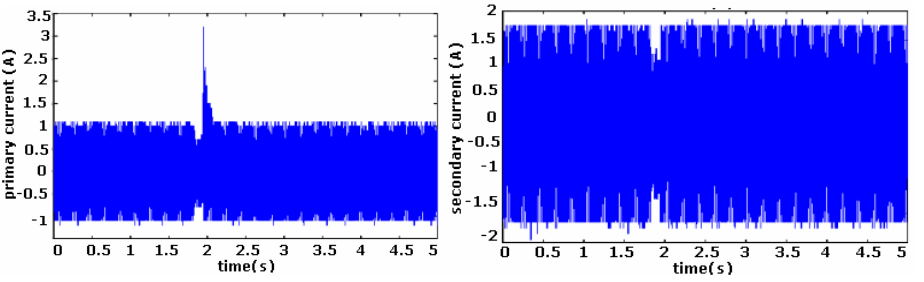


Fig. 12. Primary and Secondary Currents (voltage sag duration is 5.5T and depth is 30%)

In Fig.13 currents that belong to this condition and the response of the proposed network is shown. It is seen that the network can discriminate this condition from fault condition.

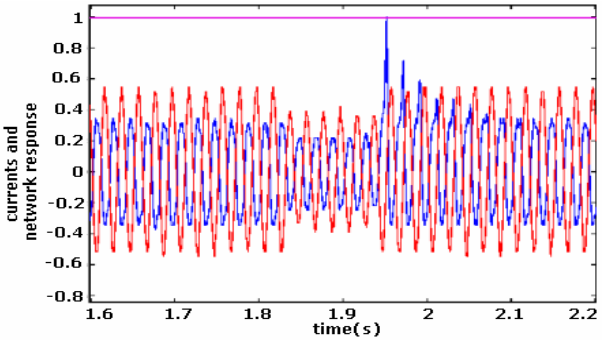


Fig. 13. Reduced primary and secondary currents and response of the network

4 Conclusions

A new approach of power transformer differential protection is proposed in this paper. The proposed relay can be suitable for all connection types of power transformers choosing a suitable current transformer normalization values and eliminating the phase shift between primary and secondary currents. According to the analysis of the training data set, the neural network was able to learn no-fault examples with accuracy 99% and fault examples with accuracy 98%. According to the analysis of the test data set, the neural network was able to discriminate no-fault examples with accuracy 100% and fault examples with accuracy 97%. The present work also includes effect of the voltage recovery after voltage dip. Important advantages of this protection are reliability in the case of voltage recovery after voltage dip and improved operating speed. The operating time of the relay is a half or less than a half cycle.

References

1. Kulidjian, A., Kasztenny, B., Campbell, B.: New Magnetizing Inrush Restraining Algorithm for Power Transformer Protection. IEEE Developments in Power Sys. Protec. Conf. 2001.
2. Zhigian, B., Geoff, W., Tom, L.: A New Technique for Transformer Protection Based on Transient Detection. IEEE Transactions on Power Delivery, Vol.15, No.3., July 2000.
3. Perez, G., Flechsig, A.J., Meador J. L., Obradovic, Z.: Training an Artificial Neural Network to Discriminate Between Magnetizing Inrush and Internal Faults. IEEE Transactions on Power Delivery, Vol.9, No.1, January 1994.
4. Pihler, J., Grcar, B., Dolinar, D.: Improved Operation of Power Transformer Protection Using Artificial Neural Network. IEEE Transac. on Power Delivery, Vol.2, No.3, July 1997.
5. Orille-Fernandez, A., Ghonaim, N.K.L., Valencia, J.A.: A FIRANN as a Differential Relay for Three Phase Power Transformer Protection. IEEE Transac. on Power Delivery, Vol.16, No.2, April 2001.
6. Guasch, L., Pedra, J.: Effects of Symmetrical Voltage Sags on Three-Phase Three-Legged Transformers. IEEE Transac. on Power Delivery, Vol.19, No.2, April 2004.

Neural Network Based Algorithm for Radiation Dose Evaluation in Heterogeneous Environments*

Jacques M. Bahi[1], Sylvain Contassot-Vivier[1], Libor Makovicka[2], Éric Martin[2], and Marc Sauget[1,**]

[1] University of Franche-Comté, Laboratoire d'Informatique de Franche-Comté, IUT Belfort-Montbéliard, Rue Engel Gros, 90016 Belfort, France
bahi@iut-bm.univ-fcomte.fr
http://info.iut-bm.univ-fcomte.fr/and
[2] University of Franche-Comté, CREST Femto-ST, IUT Belfort-Montbéliard, Portes du Jura, 4 Place Tharradin, 25200 Montbéliard, France

Abstract. An efficient and accurate algorithm for radiation dose evaluation is presented in this paper. Such computations are useful in the radiotherapic treatment planning of tumors. The originality of our approach is to use a neural network which has been trained with several homogeneous environments to deduce the doses in any kind of environment (possibly heterogeneous). Our algorithm is compared in several representative contexts to a reference simulation code in the domain.

1 Introduction

Among all the treatments of tumors, external radiotherapy is certainly one of lightest for the patient which gives good recovery results. However, such a treatment must be accurately planned in order to maximize the radiation dose received by the tumor while preserving the surrounding tissues.

In computer science, that problem corresponds to a global optimization problem whose objective is to find a series of particle beams which will produce the desired radiation dose distribution in the environment of the tumor. That process requires the possibility to evaluate the interest of a given beam and thus implies the ability to compute its impact on the treated environment, that is to say, to evaluate the radiation dose distribution resulting from that beam.

The standard method to evaluate the dose distribution in a given environment represented in a numerical form on a computer is to perform a simulation of the physical phenomenon. The most accurate and flexible simulations are those based on the Monte-Carlo algorithm such as in BEAM-nrc [1]. Unfortunately, they are very slow (several hours or days) and thus not usable in practice to compute treatment plannings in a medical environment. There exist other kinds

* Work supported by LCC,CAPM,Région Franche-Comté and Canceropôle Grand- Est.
** Authors are in alphabetic order and M.Sauget is the main contributor of this work.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 777–787, 2006.

of simulations based on approximated analytic formulations which are faster but which are not accurate enough to be used for treatment planning.

We propose in this paper an efficient and accurate algorithm to perform radiation dose evaluation in any environment. Our approach consists in using a particular algorithm together with a neural network previously trained with the doses obtained in several homogeneous (only one material) environments. With that method, it is possible to accurately evaluate the radiation dose distribution in any environment, possibly heterogeneous (containing several materials).

In that context, the neural network is used as a universal approximator. It cannot directly deduce the radiation doses in a heterogeneous environment but it can be used in a specific algorithm to fastly give the dose received at a given position in a given material if that position and material are in the range of the learned domain. Moreover, since the accuracy of its results directly depends on the accuracy of its training set, it suffices to build this set with accurate simulation codes to obtain accurate results. The computation times required to build the training set and to perform the learning itself do not reduce the interest of our approach because those two steps are made only once and not during the medical exploitation of the algorithm.

The following section describes the neural network used in our main algorithm. Section 3 details our radiation dose evaluation algorithm. The results of our algorithm are then qualitatively and quantitatively compared to those of a Monte-Carlo simulation code (BEAM-nrc) in Section 4.

2 Homogeneous Evaluation Dose Neural Network

We present in this section the different features of the neural network to be used in our main algorithm. Since this part of the problem has already been the subject of previous studies, particularly in [2], we present here the latest improvements brought to it. However, to clearly settle the problem, the global form of the function to approximate is firstly given. Then, the structure of our neural network is described followed by the learning method used.

2.1 Objective Function

Our objective function must give the radiation dose at a given point in a given material. Hence, its inputs consists in the spatial position and the density of a point and its output is the dose at that point. In order to simplify the problem and to provide intuitive graphical representations, we consider the dose distribution situated on a plane in the middle of the tridimensional environment and aligned with the axis of the accelerator as shown in Fig.1. That spatial restriction does not reduce the generality of our method since the transition from 2D to 3D environments does not imply any fundamental modification in our process. The data structure used to represent the plane of interest is a two dimensional discrete grid in which the absorbed dose is given at each discrete position in that grid. Then, using that representation, the dose distribution in any

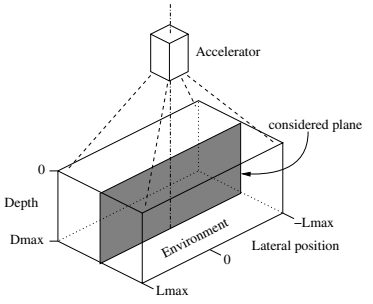


Fig. 1. Plane of interest in the environment

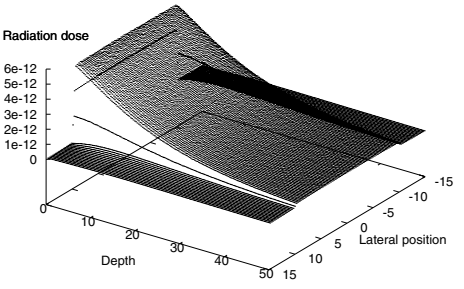


Fig. 2. Aspect of the radiation dose distribution in a homogeneous water environment for a discrete grid of 100×200 points

homogeneous environment always has the same global aspect shown in Fig.2.In the latter, sharp variations can be seen which are classically difficult cases in function approximation.

The main features of that distribution, which vary from a material to another, are the depth at which the dose distribution reaches is maximal value (usually near the surface) and the way the dose decreases after the maxima. The width of the function (around the axial position) is determined by the width of the beam and not by the material in the environment.

2.2 Structure

Many results have shown that a multi-layer neural network can be used as a universal approximator [3,4]. In our case, three layers (input, hidden and output) are sufficient to obtain the desired results. The number of neurons in the input layer is determined by the number of parameters of the objective function (spatial position and density). The number of neurons in the output layer is reduced to one neuron which delivers the dose. Finally, the number of neurons in the hidden layer is the most difficult one to determine. It does not directly depend on the number of inputs and outputs of the problem and there is no precise rule to compute it. In fact, this number of neurons rather influences the ability of the network to approximate high degree functions. However, it is not a good idea to coarsely overestimate that number since that sharply increases the learning time and may thus make the network unusable. Thus, to bypass that problem, we have designed an incremental learning method which automatically sets up the number of hidden neurons (see Section 2.3).

Also, it appears that the structure of the network has a direct impact over the learning time. Some slight modifications of the classical structure of the network can enhance its capacity to approximate high degree functions with fewer neurons and thus to be trained faster. In our context, the HPU (Higher-order Processing Unit) structure has obtained the best results among all the

tested ones. We recall that a HPU neural network has additional inputs which are polynomial combinations of the original inputs, see [5] for further details.

2.3 Learning

As mentioned above, the learning is also a critical step to obtain an accurate approximation of the function. The classical learning method used with this kind of multi-layer network is the back-propagation. Nevertheless, although this method gives good results, the learning process remains slow. Among all the tested optimizations of that process (see [6] for a survey of the existing optimizations), the Resilient back Propagation (RPROP) has obtained the best results.

Contrary to the classical back-propagation, the RPROP algorithm [7] only uses the sign of the error derivative to update the weights with an independent value. That modification value is respectively increased or decreased whether the error evolves in the same direction or not.

However, the efficient combination of the HPU structure and the RPROP learning does not avoid the problem of fixing the number of hidden neurons. Effectively, there is no a priori information which may indicate the best suited number of hidden neurons to approximate a given function. To solve that problem, incremental constructive algorithms have been proposed [8,9]. Nonetheless, due to the particular structure of our network, we had to design a quite different version from those previously proposed.

The principle of our algorithm is, as in the other incremental algorithms, to start with a given number of hidden neurons and to add new ones (one by one) during the learning process. However, in our case, a RPROP learning is performed over the initial HPU neural network until the error either reaches the required accuracy or does not sensibly evolve any more according to a given threshold. In the first case, the neural network has the desired accuracy and the learning process stops. In the second case, the learning limit of the current neural network is considered to be attained. Then, a neuron is added to the hidden layer without modifying the other neurons and links. The added neuron is initialized with null weights and threshold. After that, the learning process is resumed with that new configuration of neural network and so on until the desired accuracy is reached or the evolution of the error between two consecutive configurations of networks becomes too small (under another given threshold). In that last case, it is assumed that the overall limit of the network has been reached and that adding hidden neurons will not improve the results.

Additionally, the possibility to specify an upper bound to the number of hidden neurons has also been included in the learning process in order to limit the size of the network and, by the way, the learning time in cases of extremely slow convergences (mostly with very high desired accuracies).

Finally, the resulting neural network can learn and accurately approximate radiation doses in different homogeneous environments. However, a particular algorithm which uses that neural network is required to compute doses in heterogeneous environments in order to take into account the particular behaviors due to the material changes.

3 Radiation Dose Evaluation Algorithm

This section details our general dose evaluation algorithm. A first part is devoted to the presentation of the physical phenomenon and particularly to its behavior at the interfaces between material changes. The second part details how that behavior is taken into account in our algorithm.

3.1 Physical Phenomenon at the Interfaces Between Materials

As said in the introduction, the goal of the external radiotherapy is to expose some cells to a radiation beam. When the beam enters the cellular tissues, only a part of it is actually absorbed by the cells. Another part is deviated towards the surrounding cells, that is called the diffusion, and a last part directly goes through the cells without interacting with them. The resulting radiation dose distribution in the cells mainly depends on two parameters, the beam intensity and the density of the materials in the environment.

As we focus on the heterogeneity of the environment in the study, only one beam intensity has been considered. This restriction only implies one less input in the used neural network and does not modify the general process.

Concerning the density of each material in the environment, it sharply influences the dose distribution as can be seen in Fig.3 (left). The higher the density is, the shallower the maximal dose absorption in the material is and the sharper the dose decrease along the depth is. Moreover, the global dose distribution is always continuous, even in heterogeneous environments. So, when there is a longitudinal interface (perpendicular to the depth) between two materials there must be a continuity between the dose of the last point in the first material and the dose of the first point in the following material, as shown in Fig.3 (right). As can be seen in that figure, there might be small artefacts at the interfaces between materials. However, those artefacts are very localized and relatively negligible in a first approximation. Thus, they are not taken into account in the presented algorithm but will be the subject of a future work to obtain still more accurate results.

As our computational algorithm can only have access to the dose distributions in homogeneous environments via the neural network, it has to compute the depth shifting between the two materials corresponding to the dose at the interface, as exhibited in Fig.3 (left), in order to get the following doses.

The modification of dose distribution in the environment does not only come from the longitudinal interfaces but also from the lateral ones. Their influence is due to the diffusion of the beam in the environment and it results in soft dose transitions at the lateral interfaces. Here again, there are small artefacts at those transitions. Nevertheless, for the same reasons as for longitudinal interfaces, that local behavior is not taken into account in the presented algorithm and will be the subject of a future work. Lateral dose distributions are given at a same depth for two different materials in Fig.4 (left) and the resulting dose distribution at the same depth for an environment composed of those two materials with the lateral interface in the middle is shown in Fig.4 (right).

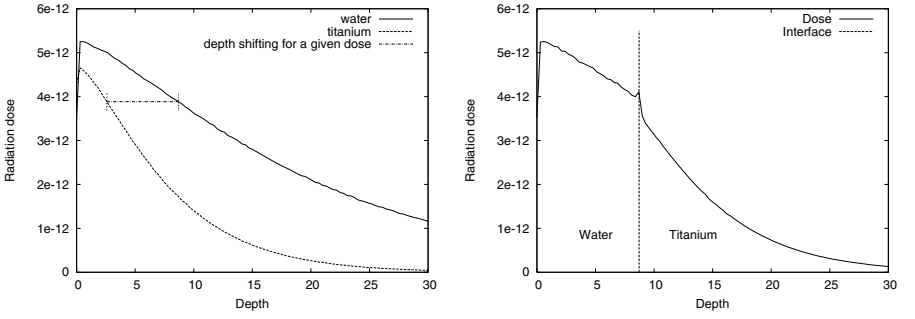


Fig. 3. Dose distributions along the depth in homogeneous environments of water and titanium (left) and dose distribution in a heterogeneous environment of water and titanium with a longitudinal interface at depth 8.7 (right)

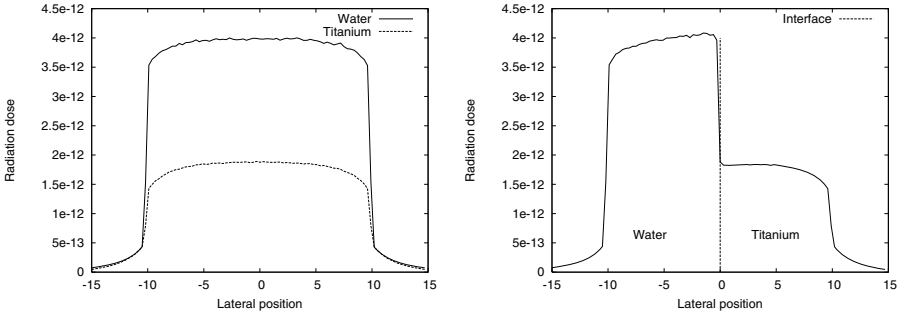


Fig. 4. Dose distributions along the lateral axis in homogeneous environments of water and titanium (left) and dose distribution in a heterogeneous environment with a lateral interface in the middle (right)

As already said, since the neural network has only been trained with homogeneous environments, it cannot directly manage the particular behaviors due to material changes. So, a particular algorithm is needed to appropriately control the use of the neural network and take into account those behaviors.

3.2 Description of the Algorithm

As said above, the dose distribution in a heterogeneous context nearly corresponds to a collection of dose distributions in homogeneous contexts which are depth shifted by the longitudinal interfaces and smoothed at the lateral ones.

Thus, the approximation of the radiation dose at each point in the environment can be made using the neural network trained with dose distributions in homogeneous environments. However, in order to get the correct answer from the neural network for a given point, it is needed to compute the inputs which actually correspond to the current context of that point according to the material changes in the environment.

As seen in the description of the physical phenomenon, two kinds of material changes must be taken into account in the dose evaluation. Since the computations related to each of those interfaces are different and can be performed independently, they are separately described in the two following parts.

Longitudinal interfaces. As mentioned above, the problem with longitudinal interfaces lies in the depth shifting implied in the dose distribution of a material which is behind another one. Thus, when computing the dose at a given point in the grid, our algorithm must take into account the dose and the density at the point at the previous depth.

According to that dependency, the computation of the doses in the environment must be performed from the shallowest depths to the deepest ones. So, the plane of interest in the environment is organized as a 2D grid whose lines are along the depths and whose columns are along the lateral axis. Then, the dose computations are performed line by line from the first line to the last one.

Since it is assumed that there is no material before the first line of the environment, each point in that line is in the same context as the homogeneous one and the doses can be directly computed by the neural network.

Concerning the following lines, the dose evaluation at each point depends on the respective densities of the current point and the previous one along the current column.

When those densities are different, our algorithm must initially find the depth shifting needed to retrieve the dose at the previous point in the homogeneous environment of the current material. Once that depth shifting is found, one just has to add to it the distance between two consecutive lines in the grid to obtain the depth in the homogeneous environment of the current material associated to the actual physical context of the current point.

When the densities are the same, there is no material change and the depth parameter of the neural network is directly deduced by adding the associated depth of the previous point and the depth step between the lines of the grid.

Finally, the neural network can be used with that associated depth together with the current density and lateral position to obtain the correct dose.

Lateral interfaces. As the lateral material interfaces produce soft dose transitions, they can be approximated by a local dose ponderation at those interfaces along the lateral axis. Thus, the process which takes into account that behavior performs a filtering of the dose distribution obtained by the previous dose estimation. The principle is to resample the doses around the interface by using a specified filter. The size of the filter (the number of neighbors taken into account) and its definition (the weights in the filter) can be adjusted to accurately conform to the experimental or simulated results.

4 Results

In this section, our algorithm, implemented in standard C++ on a classical workstation, is qualitatively and quantitatively compared to the standard Monte-Carlo

code BEAM-nrc. The following results are obtained with a network trained with two 100×200 sized homogeneous environments of water and titanium.

4.1 Qualitative Evaluation

In order to get a complete evaluation of our algorithm, we have tested it in several representative environments having the same grid size as the training. The simplest one is a homogeneous environment of water. That case allows us to verify the good learning of the neural network since there is no other treatment, due to material changes, interfering with the computation.

There are three other cases which deal with heterogeneous contexts. All those heterogeneous environments are composed of water and titanium. The water (electronic density=1) has been chosen since most of the densities in the human body are near it. The titanium has been chosen because it has a density which is far higher (ed=3.7) than what is normally in a human body, but it is used in some prosthesis. Thus, it represents quite an extreme case. Moreover, the important density difference between those two materials amplifies the dose changes at the interfaces, which represents a more difficult case for a dose evaluation algorithm.

The first heterogeneous environment only contains a longitudinal interface, as in Fig.3 (right), and allows us to test the part of our algorithm dealing with longitudinal interfaces. In a symmetrical way, the second environment only contains a lateral interface, as in Fig.4 (right), and allows us to test the part of the algorithm dealing with lateral interfaces. Finally, the last environment is more complex and contains a cylinder of titanium immersed in water, as shown in Fig.5. It permits to test the whole algorithm when there are combinations of longitudinal and lateral interfaces.

The qualitative results of our algorithm are presented in Table 1. For each test, two kinds of information are reported. The first one is the bias which corresponds to the average of the relative errors of the doses computed by our algorithm according to the doses computed by the Monte-Carlo code (used as the reference). The second one is the error which is computed in the same way except that the absolute values of the errors are used. It indicates the global accuracy

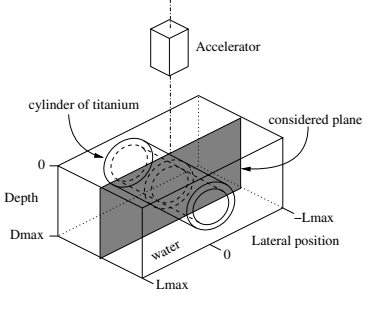


Fig. 5. A cylinder of titanium in a tank of water

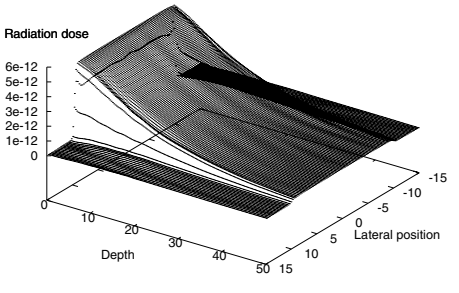


Fig. 6. Dose distribution in a homogeneous environment of water computed by our algorithm

Table 1. Bias and error of our algorithm in percentage of the reference doses

Tested environment	bias (%)	error (%)
homogeneous water	0.018	0.765
longitudinal interface	-1.459	1.855
lateral interface without ponderation	-0.154	1.479
lateral interface with ponderation	0.112	1.884
complex form	1.319	2.873

of our algorithm. Both information are computed on a representative area in the environment which corresponds to the width of the beam and to 30 cm in depth. Beyond that depth, the doses are, in most cases, no more representative.

First of all, it can be seen that our neural network is well trained since it has a very small bias and an error below 1% in the homogeneous case. It can be seen that the result of our algorithm presented in Fig.6 is very similar to the result of the Monte-Carlo code shown in Fig.2.

The results for the other tests show that our dose evaluation algorithm provides accurate results since the errors stay below 3%. We recall that the tolerated error for medical use is up to 5% and our results are quite far under this bound.

Concerning the longitudinal interface, it presents a slightly larger bias which comes from the current accuracy of our dose shifting computation. Nonetheless, since the error is quite close in absolute value to the bias, there can be expected a final error similar to the homogeneous case once that bias is removed.

Concerning the lateral interface, a distinction has been made depending on whether the ponderation is used or not. It can be seen that the ponderation tends to reduce the bias but also slightly increases the error. That behavior mainly comes from the inherent approximations in the ponderation method. However, that test case is quite extreme and better accuracies can be expected in practice.

Finally, the last test concerns the cylinder of titanium in water. The results of the Monte-Carlo code and of our algorithm are respectively presented in Fig.7. Here again, most of the error comes from the bias which is itself due to the shiftings errors. However, even if the error is larger than in the other cases, it remains quite small and completely acceptable in practical use.

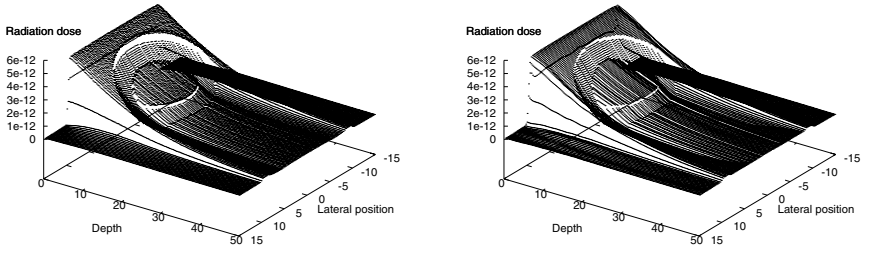


Fig. 7. Dose distribution in the complex environment respectively computed by the Monte-Carlo code (left) and our algorithm (right)

4.2 Quantitative Evaluation

As said in the introduction, the standard accurate techniques to evaluate dose distributions are highly time-consuming, typically several hours or days. In our algorithm, all the time-consuming parts of the process can be performed only once and before the use of the dose evaluation algorithm in medical applications. Hence, we obtain a largely faster algorithm than the standard Monte-Carlo techniques. In the previous tests, the computation times range from 1.002s for the homogeneous case to 1.409s for the complex one, on a classical Pentium IV 3,6Gz with 1Go of RAM.

5 Conclusion

An accurate and efficient algorithm for radiation dose evaluation in heterogeneous environments has been described. It uses a neural network trained with dose distributions from several homogeneous environments.

The neural network used in our algorithm is a three-layer HPU whose training is performed with the RPROP algorithm. Moreover, an incremental learning algorithm has been developed to automatically set the number of hidden neurons. That learning starts with a given number of hidden neurons and automatically adds new ones as required to reach the desired accuracy.

It has been pointed out that a particular algorithm is required to properly use the neural network to compute the doses in heterogeneous environments. Effectively, from the nature of its training set, the neural network cannot directly manage the particular behaviors at the interfaces between materials.

Experimental results show that the resulting process is accurate since it obtains less than 3% of error according to a standard Monte-Carlo code for representative test examples. Moreover, it has the great advantage to be largely faster since its computation times are of the order of the second for a complete 2D environment.

All those results are very promising and future works are already planned to enhance some parts of the process which let expect still better performances. With such a dose evaluation process, an efficient and accurate fully automatic system to plan radio-therapeutic treatments of cancerous tumors can be seriously envisaged in future developments.

References

1. BEAM-nrc: NRC of Canada. (http://www.irs.inms.nrc.ca/BEAM/beamhome.html)
2. Mathieu, R., Martin, E., Gschwind, R., Makovicka, L., Contassot-Vivier, S., Bahi, J.: Calculations of dose distributions using a neural network model. Physics in Medicine and Biology **50**(5) (2005) 1019–1028
3. Cybenko, G.: Approximations by superpositions of sigmoidal functions. Mathematics of Control, Signals, and Systems **2** (1989) 303–314
4. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks **2**(5) (1989) 359–366

5. Ghosh, J., Shin, Y.: Efficient higher-order neural networks for classification and function approximation. Int. J. Neural Systems **3**(4) (1992) 323–350
6. Tertois, S.: Réduction des effets des non-linéarités dans une modulation multiporteuse à l'aide de réseaux de neurones. PhD thesis, Rennes 1 (2003)
7. Riedmiller, M., Braun, H.: A direct adaptative method for faster backpropagation learning: The RPROP algorithm. In: Proceedings of the IEEE Internationnal Conference on Neural Networks (ICNN93), San Francisco (1993)
8. Fahlman, S.E., Lebiere, C.: The cascade-correlation learning architecture. In Touretzky, D.S., ed.: Advances in Neural Information Processing Systems. Volume 2., Denver 1989, Morgan Kaufmann, San Mateo (1990) 524–532
9. Dunkin, N., Shawe-Taylor, J., Koiran, P.: A new incremental learning technique. In Verlag, S., ed.: Neural Nets Wirn Vietri-96. Proceedings of the 8th Italian Workshop on Neural Nets. (1997) 112–118

Exploring the Intrinsic Structure of Magnetic Resonance Spectra Tumor Data Based on Independent Component Analysis and Correlation Analysis

Jian Ma and Zengqi Sun

Department of Computer Science, Tsinghua University,
100084 Beijing, China
Majian03@mails.tsinghua.edu.cn,
Szq-dcs@mail.tsinghua.edu.cn

Abstract. Analysis on magnetic resonance spectra (MRS) data gives a deep insight into pathology of many types of tumors. In this paper, a new method based on independent component analysis (ICA) and correlation analysis is proposed for MRS tumour data structure analysis. First, independent components and their coefficients are derived by ICA. Those components are interpreted in terms of metabolites, which interrelate with each other in tissues. Then correlation analysis is performed to reveal the interrelationship on coefficient of ICs, where residue dependence of components of metabolites remains. The method was performed on MRS data of hepatic encephalopathy. Experimental results reveal the intrinsic data structure and describe the pathological interrelation between parts of the structure successfully.

1 Introduction

Magnetic Resonance Spectra (MRS) has a strong ability of monitoring the metabolism of tissues for clinical purpose, in which chemical information can be obtained from a well-defined region of interest. In MR spectrum, peaks corresponding to different chemical substances can be observed. Particularly, in ^{1}H MR spectra of normal and pathological brain resonances from metabolites, such as N-Acetylaspartate (NAA), Choline (Cho), Creatine (Cr), Lactate and etc., are presented together. Thereby, MRS can provide much clear characteristic information of tissues than magnetic resonance imaging technique does. MRS patterns are much reliable in clinical tumor diagnosis and have been used extensively in identification of tumors, where standard MR imaging methods usually fail [3]. However, studies indicated that the densities of certain metabolites vary in pathological tissues with respect to tumor type and grade, which reflect as different structure of peaks corresponding to those metabolites in MRS. Complicated MRS structure makes a big obstacle for clinicians on MRS interpretation.

There has been much of research interest in the use of statistical methods, particularly pattern recognition techniques for analysis, interpretation and classification of MRS data [2,4-7]. The fundamental idea of previous research is that MRS data is a

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 788 – 797, 2006.

linear combination of a group of components corresponding to metabolites. Based on the idea, many works have been contributed to the decomposition of MRS and attempted to recover these individual components respectively [1]. Contrasted to statistical methods, model-based decomposition methods were usually used [14]. However, model-based methods require priori knowledge and human interferences. Furthermore, the accuracy of results from model-based methods is reduced in the situation where there are noise presented and peaks overlapped in MRS. Whereas, statistical methods, such as principal component analysis (PCA) and bayesian analysis, which have more flexibility to tackle complicated situations using statistical information of dataset, have been widely studied. However, clinical MRS usually has poor signal-to-noise-ratio and overlapping peaks, which contains circumstance artifacts and irregular baselines. Meanwhile, the dataset usually is small size. Hence, those statistical methods could not present good performance, and sometimes fail on MRS data analysis.

ICA is a new statistical method which has been widely used in the field of biomedical signal processing [1,8,9]. It deals with the problem that recovers the hidden constituent components from observed linearly mixing signals under the assumption of mutual independence of those components. As mentioned above, MRS is an integrating structure of a group of peaks corresponding to certain metabolites. Each underlying metabolite could be considered as a random variable. So based on this model ICA is considered as a suitable method for MRS structure analysis. There exist some successfully applications which decomposed MRS into biomedical interpretable independent components [1]. Unfortunately, metabolites in tissues reflected in MRS are not mutually independent. Actually, there is biological interrelation between their changes corresponding to pathological mechanism. So in spite of the assumption of independence, components derived by ICA have residue dependence. The problem here is that how to find the residue dependence and then to present a more accurate interpretation of MRS data.

In this paper, a new method to solve the above problem by combining ICA and correlation analysis is proposed. The main idea is that the interrelation between components derived by ICA was measured by correlation coefficient (CorCoef), through spearman correlation analysis. According to those coefficients, similar meaningful components should be combined and different kinds of components should be considered as interrelated with each other. Thus, a further insight into the mechanism of diseases can be gained.

2 Material and Methods

2.1 Data Acquisition

This research was performed on a clinical 1.5T GE Horizon MRI scanner (General Electric, Waukesha, WI) with high speed gradients. The dataset are acquired from 33 subjects, including 10 normal volunteers and 23 brain tumor (hepatic encephalopathy: HE) patients. All subjects authorized the collected MRS data using in our research. The data collected for the study is Single-voxel In Vivo Human HE MR spectra data. Two types of data acquisition (PRESS and STEAM) were used with an echo time (TE) at 35ms and a repetition time (TR) at 1500ms. Therefore, every subject could contribute more than one sample to dataset. The final dataset comprises a total of 80

spectra, including 23 normal samples and 57 tumor samples, which assigned classes by clinical examination. Some of data are with poor signal-to-noise-ratio. After post processing, spectra are still additionally corrupted by technical artifacts or uneven liquid baselines. All the spectra data are sampled into 512 dimensions, representing the ppm range from 4.295ppm to -0.571ppm.

2.2 Independent Component Analysis and FastICA

ICA extracts independent components from their mixing data. A good survey has been presented by Hyvarinen [9]. One of implementation of ICA is the FastICA by Hugo Gävert etc [13]. It takes mutual information as the measure of independence between components and furthermore introduces a new kind of contrast function for ICA which is approximated through negentropy. The approximation is of the form: $J(s) \approx c[\mathrm{E}\{G(s)\} - \mathrm{E}\{G(v)\}]^2$, where s denotes the random variable of zero mean and unit variance, Gaussian variable $v \sim N(0,1)$, and c is irrelevant constant. Non-quadratic function $G(\bullet)$ should be chosen with respect to the applications.

2.3 Spearman Correlation Analysis

There are two kinds of commonly used correlation analysis methods: Pearson correlation analysis and Spearman correlation analysis. The difference between them is that the former is a parametric method while the latter is a nonparametric one. In detail, the former assumes the distribution of correlated variances belong to a Gaussian family while the latter has no such assumptions on variances.

Let x, y represent two different rows of A, and x_i, y_i denote the observation of x, y respectively and s_i, t_i denote rank order of x_i, y_i. Then calculation of the spearman rank order CorCoef is as follows:

$$r_{xy} = \frac{\sum_{i=1}^{n}(s_i - \bar{s})(t_i - \bar{t})}{\sqrt{\sum_{i=1}^{n}(s_i - s)^2 \sum_{i=1}^{n}(t_i - t)^2}} \tag{1}$$

where $r_{xy} \in [-1,1]$.

The CorCoef r_{xy}, ranging from -1 to 1, can be interpreted in terms of the interrelation between *x* and *y* as listed in **Table 1**.

Table 1. Rules for the interpretation of CorCoef

Value of r_{xy}	Interpretation
r = 1.0	*x* and *y* vary in the exact same way.
0 < r < 1	*x* and *y* increase or decrease together.
r= 0	No correlation.
-1 < r < 0	One increases as the other decreases.
r = -1.0	*x* and y vary in the inverse way.

3 MRS Data Structure Analysis

As there are some abnormal spectra called outlier in the data, PCA was utilized to remove them from the dataset at first. Note that these outliers caused by unsuccessfully post processing contain abrupt peaks or heavily contained noise or uneven baselines, which should bias the sequential results seriously. First, PCA transforms the dataset into a new space coordinated by vectors which represents the main distributions of data. 512 PCs were derived from 512 dimension data. In this space, Outliers should have much bigger loadings at certain direction of vector than normal data, as illustrated in **Fig. 1**(b). Thus, they can be removed by thresholding on loadings of principal components (PCs). The value of thresholding should be chosen carefully in case that the samples which have the abnormal MRS indicating the difference between normal and patients were removed. Only the loadings of the front part of total 512 PCs are referenced because PCs are ordered and mainly represents the distribution of dataset. We calculated the mean and standard variance of loadings and then the threshold of loading was determined as three times the standard variance according to experience.

After outlier removed, the FastICA algorithm was performed on the remaining samples. First, the type of nonlinearity function $g(\bullet)$, the derivative of $G(\bullet)$, was chosen. The criteria to choose the nonlinearity is to investigate which one can make a better meaningful decomposition than others. There are four options for nonlinearity: 'pow3', 'tanh', 'gauss' and 'skew' listed in **Table 2**. The FastICA algorithm was performed on the dataset with every nonlinearity function four times to make sure good results because the FastICA initiate the separating matrix W randomly.

Table 2. Nonlinearity functions using in the FastICA package

nonlinearity	Function
pow3	$g(x) = x^3$
tanh	$g(x) = \tanh(x)$
gauss	$g(x) = \exp(x^2 / 2u)$
skew	$g(x) = x^2$

To evaluate the results, we proposed two criteria: 1) SNR of ICs and 2) nICR: ratio of first N IC on the whole dataset. SNR of ICs was defined as

$$I_{snr} = Max / V_{std} \tag{2}$$

where Max and V_{std} denotes the max absolute value and standard variation of IC respectively. nICR is calculated as follows: the first N IC with their coefficients was used to reconstruct a new dataset D, D_{sum} and S_{sum} is the sum vector of D with respect to rows (samples), and then nICR is defined as

$$nICR = D_{sum} \cdot / S_{sum} \tag{3}$$

I_{snr} describes the whole quality of IC set. *nICR* reflects the ability of nonlinearity extracting meaningful components because those meaningful components always list at front relatively.

The proper number of independent components should be decided. There are two reasons: first, the FastICA algorithm assumes that the number of underlying components is equal to that of observations. However, the intrinsic dimension of MRS is comparably low, as far as the number of observations is concerned. Therefore, not all IC are necessary. Second, ICA as a kind of data-driven methods is very sensitive to the quality of inputs data. Especially when there is insufficient data artifacts like bumps and spikes are generated in IC sets from dataset not as expected. They must be eliminated from the results. So it is necessary tuning the parameter to decide the number of independent components in order that all the meaningful components could be reserved and artifacts and noise elements could be excluded from the final component set. The meaningful components have interpretable peaks and artifacts are those that have a narrow sharp and uninterpretable peak. Each metabolite corresponds to a peak with a fixed position in MR spectrum. The meaningful component is labeled as the metabolite, peak of which is no more than 0.01ppm closer to the dominated peak of the component. The reference positions of some important metabolites are listed in **Table 3**. Thus the intrinsic dimension of MRS is about 8.

Table 3. Reference position of some metabolites

metabolites	NAA	Cho	Lac.	myo-In	Cr.	Taur.	liquids	Glx
position (*ppm*)	2.009	3.209	1.318	3.275	3.0	3.418	1.4-0.9	2.45-2.28

Correlation analysis was used to identify the correlationship of metabolites. As mentioned above, there are residue dependences on components in spite of the assumption of mutual independence of ICA on them. While a group of proper independent components was obtained, a mixing matrix *A* was generated which is composed of coefficient of the components. Each row of the matrix *A* corresponding to one component represents the density vector of metabolites or other elements. The nonparametric spearman CorCoefs were calculated on rows of matrix A. Thus a square correlation matrix is obtained, dimension of which are indexed by labeled components. Each element of the matrix corresponds to a pair of components. Obviously, the diagonal elements of matrix are '1'.

Based on correlation matrix, the relation between metabolites could be analyzed. In most cases, only a few pair of variables is interrelated and thus has a relatively high value of CorCoef. In this study, only if the absolute value of a matrix element is above 0.5 the corresponding two variables are considered as strongly correlated with each other. Depending on the positive or negative sign of the value, the correlationship is regarded as in the same or inverse way. The variables were grouped which mutually interrelated.

4 Experimental Results

PCA removed bad samples out of our dataset. 512 PCs are derived by PCA decomposition from 512 dimension dataset. First 20 PCs contributing 90% energy of the total dataset was selected as references of the removing methods. Each PC contains some dominated peaks, as shown in **Fig. 1**(a).Hence, a big loading on those PCs means that the sample has unusual big peaks. 36 samples are marked as outlier, loadings of which are bigger than 3 times than standard variance. One PC, its corresponding scores, and the corresponding removed samples are illustrated in **Fig. 1**.

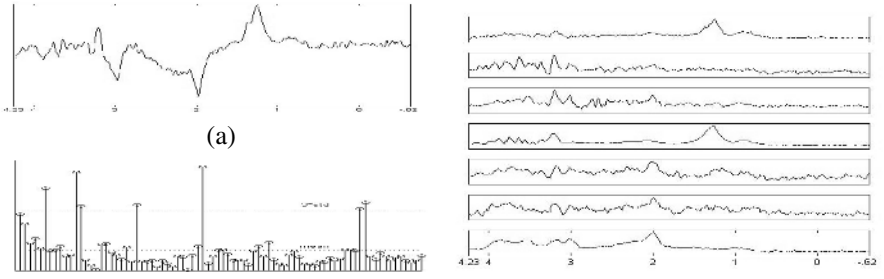


Fig. 1. Removing outlier by PCA. (a) the 2nd principal component. (b) The loading of the above PC of samples. (c) the removed samples according to (b).

All the samples were manually checked to guarantee that no bad sample was neglected. It could be learn from the results that those heavily contaminated by noise or corrupted by unsuppressed liquid signals peaks or distorted are eliminated.

ICA decomposed the remaining data into a group of interpretable 'independent' components. The FastICA package provides four types of nonlinearity. After comparing on IC results, we learned that the nonlinearity 'tanh' is a better option than others. The detail results are listed in **Table 4**. The default number of ICs in the experiments is 46, the size of the dataset. However, only *pow3* gave 46 ICs in four times most quickly, skew failed to converge before the default number of IC reaching, and the other two nonlinearities run in moderate time and did not always converge.

The two criteria proposed above show us an interesting result which was listed in **Table 4**. The average SNR of components range about 6. *Pow3* presents the highest value of SNR, followed by *skew*, *tanh* and *gauss* orderly. It seems that *pow3* gives the best ICs while *gauss* the worse. However, It can learn from every ICs that the ICs of *pow3* tends to be an over fitting result with sharp-peaked artifacts and that some ICs of *gauss* contained more than one peaks which means the failure of MRS decomposition to some extent. *nICR* of *pow3* (N=15) is too low in that overfitting caused too much meaningless ICs, while *tanh* and *gauss* are much better than *skew*. Notice that when all the ICs of *skew* were used to remix the dataset, the ratio of remixing data to

Table 4. Decomposition results derived from the FastICA algorithm in terms of nonlinearity

Experiments		# of IC set	SNR	*nICR*	
nonlinearity	#time			15	All
pow3	1	46	6.9357	0.3567	1.0
	2	46	6.8485	0.5288	1.0
	3	46	6.8426	0.5437	1.0
	4	46	6.8081	0.3567	1.0
tanh	1	46	6.0040	0.7402	1.0
	2	43	6.2982	0.7245	0.9993
	3	46	6.0632	0.6985	1.0
	4	42	6.2744	0.6473	0.9901
gauss	1	41	5.7593	0.7569	0.9931
	2	42	5.8079	0.6993	0.9421
	3	44	5.6556	0.8211	0.9746
	4	46	5.5588	0.6609	1.0
skew	1	40	6.7985	0.5911	0.6326
	2	39	6.8461	0.5775	0.7252
	3	43	6.6652	0.6601	0.8355
	4	41	6.7979	0.6527	0.8794

the original data is only about 0.6-0.88. There may be two reasons for this: the failure of convergence and the bad quality of ICs. So, *skew* is untrustworthy. Considering the above results, it could be concluded that *tanh* is much reliable and effective than the others and therefore its results were adopted for the sequent analysis.

A proper number of ICs is chosen while *tanh* used. It was observed that the meaningful components always ranked at front. Considering that the intrinsic dimension of MRS data is about 10, we choose the first 15 ICs which correspond to the most important metabolites related to tumors and thus all valuable components are included while others irrelevant elements were excluded. **Fig. 2** shows one of the 15 IC results extracting from the remaining dataset. Most of those ICs can be interpreted in terms of metabolites.

Spearman CorCoefs matrix was calculated on the above components. Every pairs of components with a strong correlation (CorCoef larger than 0.5) were illustrated in **Fig. 3**. As shown in **Fig. 3**, there are two kinds of correlation: Glux and Cho correlated negatively, that is to say, when one of them increasing, the other would decrease. All the other correlations are positive, which means correlated components vary in the same way. Study had shown that the MR spectrum of hepatic encephalopathy is significantly characterized by markedly reduced myo-Inositol, decreased Cho and Taurine, and elevated glutamine [10-12]. This relation is certified by our results that When tumour occurs in tissues, Glux will increase, Cho, Taurine and myo-In will decrease simultaneously. Twin components mentioned above, now connected with each other. It should be mentioned that CorCoef between Glux and *Glux, Cr. and

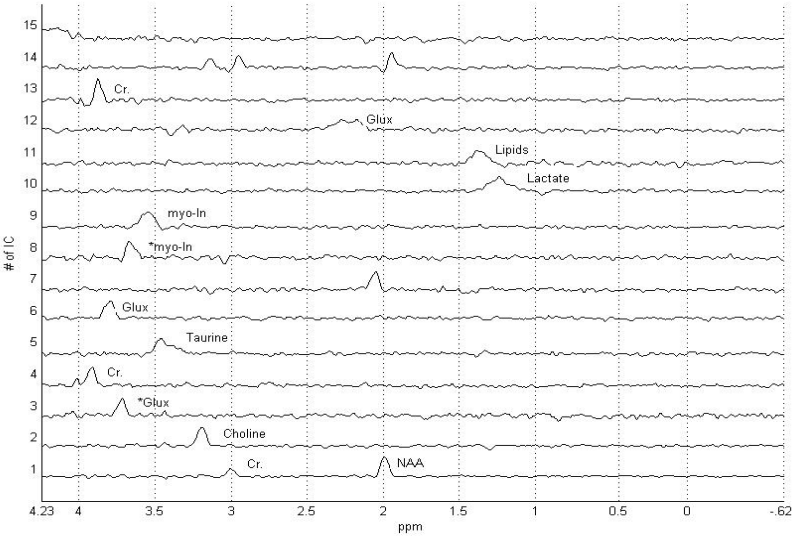


Fig. 2 The first 15 ICs obtained by the FastICA algorithm with nonlinearity tanh, are labeled with interpreting metabolites. The initial '*' in label means a components with a slight shift from a reference position.

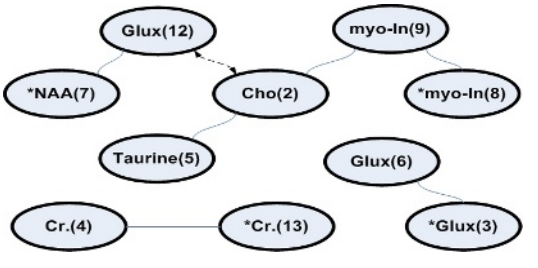


Fig. 3. Correlation between components. Correlation is significant at the 0.01 level. Each node represents a component, which labeled in the form of the abbreviation of interpreted metabolite and the rank in IC set. Each association line represents a big CorCoef. There are two styles of lines: solid line means positive CorCoef and dash line means negative CorCoef.

*Cr. are not above 0.5, but very close to it. Except the above CorCoef, the other CorCoefs are relatively or much lower than 0.5. Hence, it can be learn from above that correlation analysis on residue dependence could not only reveal the intrinsic structure of relationship between metabolites but also links the components which are from the same metabolite but split by variation.

5 Discussion

Based on ICA, the dependency between metabolites of HE in MRS was studied. In theory, components that derived from ICA are mutually independent. Hence, when

there is dependency between the underlying components of the real problem, the model of ICA is unfit. However, if a proper nonlinearity is chosen ICA could still separate those components appropriately from their mixing with dependency remaining in coefficients of ICs or with combining components from more than one component. Especially when data are insufficient for fully analysis, ICA may not decompose them thoroughly into mutually independent components and thus much dependency would remain. Residue dependency at coefficients of ICs provides a way of having an insight into the intrinsic structure of data.

There are some means for the study of dependence between components. In our study, correlation analysis was used to measure the dependence relation between metabolites in MR spectra. It is because that pathological study on metabolites in MRS only concerns the density of metabolites and their relative ratios. Though our method could not generate rules for diagnosis, it could mine the coherence of metabolites and thus furthers the study on MRS data analysis. The Spearman test in stead of the Pearson test in our study was used in that the components derived by ICA are assumed to be non-Gaussinity. The correlation between improperly separated components might be taken into account. Although when study was confined on carefully examined meaningful components, the result should be reliable.

6 Conclusions

In this paper, a new method using ICA and correlation analysis was presented for MRS data analysis. First, PCA was used to eliminate bad quality samples. Then ICA extracts interpretable components from remaining MRS data. Two criteria were proposed for evaluating ICs. They present much insight on ICs. Consequently, the Spearman correlation analysis was performed on the coefficient of ICs to investigate the relation between metabolites. Correlation coefficients indicate that when the density of Glux increases, that of myo-In, Cho, and Taurine will decrease accordingly. This coincides with the pathological analysis of hepatic encephalopathy. According to correlation coefficient, the split components caused by variations in data should be identified by the quantities of correlation.

Our study shows that residue dependence of ICA is helpful to analysis the intrinsic structure and interrelation of metabolites of MRS data for clinical purpose. The new method furthers the application of ICA on MRS data.

References

1. Ladroue, C., et al.: Independent component analysis for automated decomposition of in vivo magnetic resonance spectra. Magnet Reson Med (2003)50:697-703
2. Anthony, M.L., et al.: Classification of toxin-induced changes in 1H NMR spectra of urine using an artificial neural network. J Pharmaceut Biomed (1995)13:205-211
3. Arjan W. Simonetti, W.J.M., Fabien Szabo de Edelenyi, Jack J. A. van Asten, Arend Heerschap, Lutgarde M. C. Buydens: Combination of feature-reduced MR spectroscopic and MR imaging data for improved brain tumor classification. NMR Biomed (2005)18:34-43

4. Devos, A., et al.: Classification of brain tumours using short echo time H-1 MR spectra. J Magn Reson (2004)170:164-175
5. C. E. Mountford, R.L.S., P. Malycha, L. Gluch, C. Lean, P. Russell, B. Barraclough, D. Gillett, U. Himmelreich, B. Dolenko, A. E. Nikulin, I. C. P. Smith: Diagnosis and prognosis of breast cancer by magnetic resonance spectroscopy of fine-needle aspirates analysed using a statistical classification strategy. Brit J Surg (2002)88:1234-1240
6. Tate, A.R., et al.: Automated classification of short echo time in in vivo 1H brain tumor spectra: A multicenter study. Magnet Reson Med (2003)49:29-36
7. Howells, S.L., R.J. Maxwell, and J.R. Griffiths: Classification of tumour 1H NMR spectra by pattern recognition. NMR Biomed (1992)5:59-64
8. James, C.J. and C.W. Hesse: Independent component analysis for biomedical signals. Physiol Meas (2005)26:R15
9. Aapo Hyvarinen: Survey on independent component analysis. Neural Comput Surveys (1999)2:94-128
10. Häussinger, D.: Hepatic encephalopathy: clinical aspects and pathogenesis. Deutsche medizinische Wochenschrift (2004)129(Suppl 2): 66-7
11. Kreis, R., et al.: disorders of the brain in chronic hepatic encephalopathy detected with H-1 MR spectroscopy. Radiology (1992)182:19-27
12. Kreis R, Farrow N, Ross BD: Localized 1H NMR spectroscopy in patients with chronic hepatic encephalopathy. Analysis of changes in cerebral glutamine, choline and inositols. NMR Biomed (1991)4:109-16
13. H. Gavert, J. Hurri, J. Sarela, and A. Hyvrinen. Fast-ICA for matlab 5.x, 2001. http://www.cis.hut.fi/projects/ica/fastica/
14. Kreis, R.: Quantitative localized 1H MR spectroscopy for clinical use. Progress in Nuclear Magnetic Resonance Spectroscopy (1997)31:155-195

Fusing Biomedical Multi-modal Data for Exploratory Data Analysis

Christian Martin, Harmen grosse Deters, and Tim W. Nattkemper

Technical Faculty, Bielefeld University, Germany
christian.martin@uni-bielefeld.de,
tim.nattkemper@uni-bielefeld.de,
hgrosse@techfak.uni-bielefeld.de
www.techfak.uni-bielefeld.de/ags/ani

Abstract. Data analysis in modern biomedical research has to integrate data from different sources, like microarray, clinical and categorical data, so called multi-modal data. The reef SOM, a metaphoric display, is applied and further improved such that it allows the simultaneous display of biomedical multi-modal data for an exploratory analysis. Visualizations of microarray, clinical, and category data are combined in one informative and entertaining image. The U-matrix of the SOM trained on microarray data is visualized as an underwater sea bed using color and texture. The clinical data and category data are integrated in the form of fish shaped glyphs. The resulting images are intuitive, entertaining and can easily be interpreted by the biomedical collaborator, since specific knowledge about the SOM algorithm is not required. Visual inspection enables the detection of interesting structural patterns in the multi-modal data when browsing through and zooming into the image. Results of such an analysis are presented for the van't Veer data set.

Keywords: data mining, exploratory data analysis, semantic data integration, information visualization, self organizing maps, neural networks, multi-modal data, complex data.

1 Introduction

In modern biomedical research data from different sources, so called multi-modal data or complex data, is often linked together to allow a holistic analysis. Especially in clinical studies concerning cancer research, clinical and categorical data is completed by gene expression data from microarray experiments in the last decade [1,2]. The clinical data may contain age, weight or sex of the patient, the size or the grade of the tumor, information about the lymph nodes, or results from a histological analysis [3]. The categorical data contain a classification of the tumor regarding its malignancy. This might be the patients survival time after the analysis or the success of a chemotherapy. In the last years, the number of experiments and studies using microarray technology has increased considerably. Especially for breast cancer research there are at least 39 studies, and for about two third of them data is available on the internet [4]. Up to 25,000 genes

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 798–807, 2006.

can be analyzed simultaneously, even though often only a fraction of these genes is selected and used for further analysis [5]. The major challenge is how all this clinical, categorical and genomic data can be analyzed in an integrative manner. This problem is further increased by the high dimensionality of the data. Usually the number of available experiments (number of samples) is approximately of the same magnitude or even smaller than the number of genes (dimensionality of data space) analyzed, which makes the application of statistical test methods impracticable. Many machine learning methods can handle such difficulties, but most of them are based on pairwise similarities, which cannot be defined appropriately for multi-modal data. Considering all these aspects makes a more interactive, exploratory data analysis seem more reasonable. To allow an exploratory study, the multi-modal data, which is usually distributed in several media (tables, flat files) must be integrated into one representation, combining visualizations of all kinds of available data. A simultaneous visual inspection of all modalities enables the detection of patterns and structure in the data when browsing through and zooming into the image.

In this paper we propose an integrative multi-modal visualization approach based on the Self-Organizing Map (SOM) [6]. The basic idea is to render a multi-modal visualization to display multi-modal data. In this work we design a visualization consisting of dimension reduction and multivariate object display, i. e. data glyphs. The SOM algorithm comprises the aspects dimension reduction, clustering and visualization and is well suited for the analysis and visualization of the microarray data [7,8].

Thus the first data modality, the microarray data is fed into the SOM. Displaying the trained SOM with the U-matrix approach [9] visualizes structural features of the high dimensional microarray data space. We expand the visualized U-matrix by introducing multivariate data glyphs in order to display clinical and categorical data.

Using a metaphoric display approach [10] the SOMs U-matrix is rendered as an underwater sea bed with color and texture. In contrast to the reef SOM presented in [10] the underwater landscape can then be completed with glyphs generated from data from different sources, which was not used for training of the SOM. In this work we use a kind of metaphoric glyph, a so called fish glyph. The fish glyphs have two groups of parameters that describe shape or colors. We use this to display clinical data by shape and categorical features with color. The resulting images are both informative and entertaining and can easily be interpreted by the biomedical collaborator, since specific knowledge about the SOM algorithm is not required. Its visual inspection might reveal interesting structural pattern in microarray, clinical and categorical data.

2 SOM-Based Sea Bed Rendering

The self-organizing map as proposed in [11] provides an unsupervised learning algorithm for dimension reduction, clustering and visualization which is easy to implement [12]. To visualize the trained SOM, several approaches have been

proposed: The feature density of the trained SOM prototype vectors is displayed based on smoothed histograms [13], the U-matrix [9], or by clustering the prototype vectors [14,15]. For the special case of very large SOMs, fish eye view or fractal view have been proposed [16]. In addition, the SOM visualization can be augmented by text labels, as for instance the WEBSOM [17] or a single feature analysis with a component plane view [18]. Also automatic feature selection has been proposed to render icons for displaying the SOM prototype vectors on a grid [19].

The U-matrix as proposed by Ultsch [9] is probably the most applied visualization framework for SOM, especially for SOM with a large number of neurons. The U-matrix visualizes the data structure by a display of approximated data densities at the SOM grid nodes. For each node, the average distance to all its neighboring nodes is computed. These average distances are displayed by a height profile or by a colored plane. In this work we combine both techniques and visualize the U-matrix as a colored height profile. Since we consider an underwater scenario we visualize the U-matrix the other way round, i. e. we draw the *depths* of the sea bed proportional to the average distances. So in the display clusters of very different data are separated by valleys. However, it should be noticed that in the case of overlapped and interconnected clusters as they often occur when analyzing microarray data, the U-matrix approach might show some limitations.

3 The Fish Glyph

Glyphs (or icons) are parameterized geometrical models that are used for an integrated display of multivariate data items. The idea is to map the variables of one data item to the parameters of one glyph so that the visual appearance of the glyph encodes the data variables.

Glyph approaches can be classified as being *abstract* or *metaphoric*. Abstract glyphs are basic geometric models without direct symbolic or semantic interpretation like profiles [20], stars [21], boxes [22]. To display more variables or also data relations, abstract glyphs can get quite complex like the customized glyphs [23,24], shapes [25] or infochrystals [26]. Such glyphs can be powerful tools for a compact display of a large number of variables and relations. However, the user must spend considerable time for training to be able to use these tools effectively.

Since the idea of using metaphoric display is quite natural, metaphoric glyphs have been proposed in the earliest years of information visualization already. In 1970, the well known Chernoff faces [27] were introduced for multivariate data display. The idea of rendering data faces may get new stimuli from advances in computer graphics and animation [28] since a large range of algorithms exist to render faces in different emotional states. However, the successful application of Chernoff faces seems to be restricted to data with a one-dimensional substructure, like social and economic parameters as in [29,30,31]. Similar approaches use stick figures [32], a parameterized tree [33] or wheels [34]. To visualize the SOM in a metaphoric manner, we need to synchronize the designs of the U-matrix

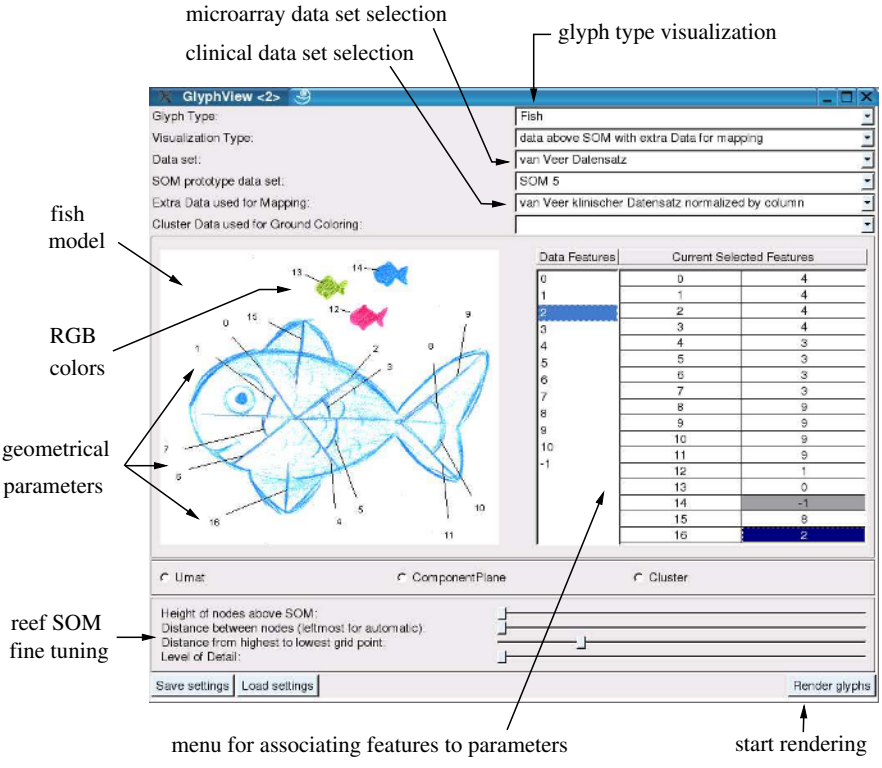


Fig. 1. The fish glyph GUI: In the left half a fish cartoon shows the geometrical parameters p_k of the fish glyph. This is used for associating the clinical variables and categories to the parameters $x_j^{(i)}$. Parameters $p_0, \ldots, p_{11}, p_{15}$ and p_{16} encode the geometrical properties of the fish and can be used to display different clinical variables. Parameters p_{12} to p_{14} encode the RGB color of the fish and can be used to display a category. The right side is used to map clinical variables and categories to the 17 fish glyph parameters. A value of -1 encodes, that no variable is associated to this parameter. In this case a default value is taken. In the lower part of the GUI, fine tuning can be applied, parameter settings can be stored and loaded and the rendering process of the reef SOM can be triggered.

landscape and the data glyphs. To this end we developed a fish shaped glyph. The fish glyph is used to display (i) the prototypes of the SOM or (ii) all the items of the data set or (iii) both. In mode (ii) and (iii) the data set items are to be visualized on top of the sea bed, i. e. the SOM. But, the computation of an appropriate two dimensional grid position for each data item on the SOM (relative to the SOM node coordinates) is a nontrivial problem. The most naive approach is to take the grid coordinates of the winner node This approach must fail, if the number of data items per winner node exceeds one, since in this case two fish must be rendered at the same position. A more advanced solution is to interpolate the two dimensional position from the grid node positions of several

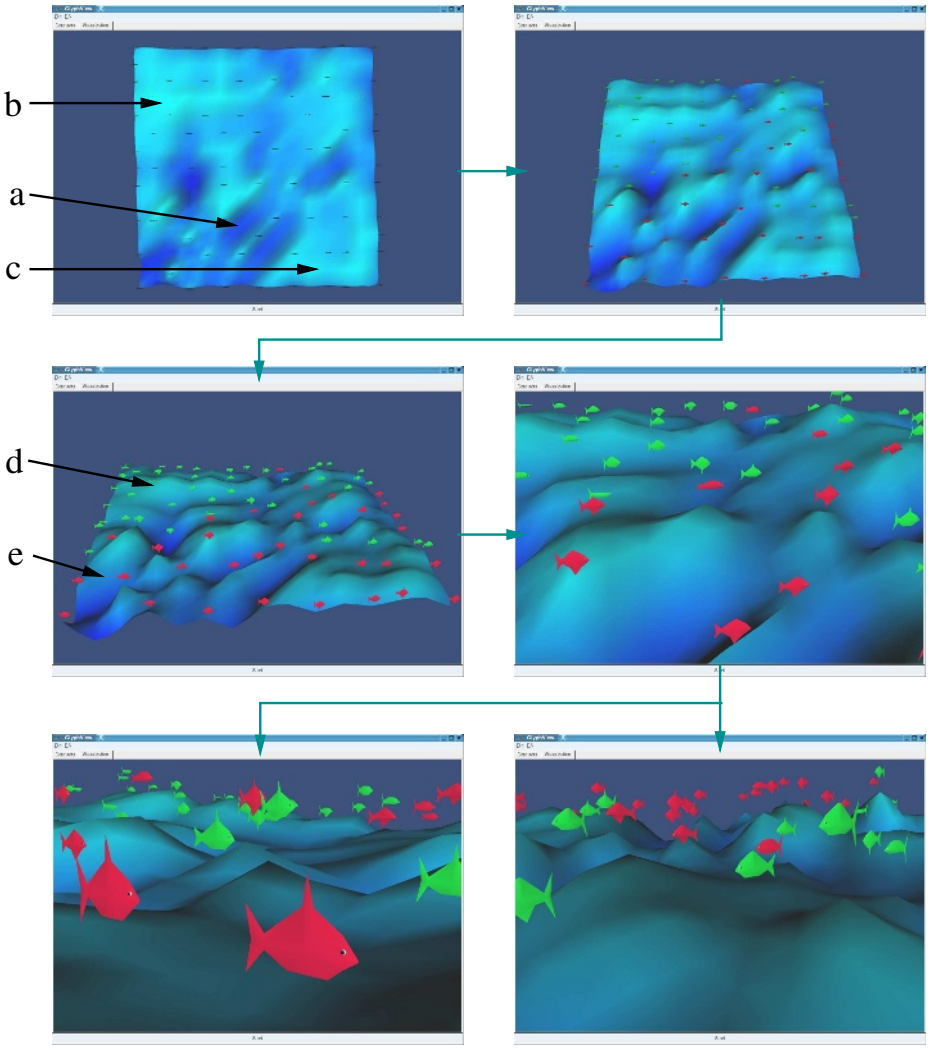


Fig. 2. A flight into the reef SOM of the microarray data set is shown. The sea bed is barely divided by one abyss (**a**) into two plateaus (**b** and **c**). An inspection of the fish glyphs reveals, that the top left plateau is dominated by green fish corresponding to patients who survived the next five years (**d**). The front is dominated by red fish (patients who died within the next five years) which are placed in the abyss as well as on the plateau (**e**).

nodes. In the literature, some approaches have been proposed, most of them applying advanced interpolation algorithms. In our first version of the software, we disclaim an exact positioning of the data items on the SOM and render each data item at a random position in the close vicinity of its winner node. On first sight, this strategy looks a bit crude, but it is motivated by several arguments.

First, several solutions to the interpolation problem have been proposed and there is not *one* solution which is accepted by the entire community. Second, one important feature of each data item is its cluster prototype, i. e. its nearest neighbor. If the interpolation leads to suboptimal results, the data item, or its glyph, is rendered at a position closer to another node , which makes it visually infeasible to identify the winner node correctly. Third, the random strategy is the computationally least expensive one.

The fish model consists of two kinds of parameters, 14 geometric parameters (six angles and eight arc lengths) and three color values (RGB). The mapping of the clinical values to the fish parameters can either be done automatically or can be defined by the user using the fish glyph GUI (Figure 1). Prior knowledge can be used to map similar clinical values to related fish parameters.

In Figure 2 a flight into the reef SOM illustrates, how the color and shape of fish, rendered on top of a U-matrix sea bed, varies.

4 Application

To illustrate the application and usefulness of the reef SOM for the exploratory analysis of biomedical multi-modal data, results are shown for the van't Veer breast cancer data set [3]. It consists of microarray, clinical and categorical data and is available via internet. The microarray data comprises the analysis of 25000 genes for 78 primary breast samples. For each gene and sample the logarithm of basis 10 of the intensity and the ratio ($[-2, 2]$) are provided. The original gene pool was reduced (mainly by using statistical methods) in three steps to 5000, 230 and finally 70 genes forming sets of marker genes that are somehow related with breast cancer outcome [3]. Since gene selection usually helps to improve the results in the domain of microarray data, we focus on the gene set with 70 genes to compute the SOM reef. More sophisticated gene selection techniques might further improve the SOM reef results but are not considered here. The clinical data contain the age of the subject (28 to 62 years), the grade (I to III) and diameter (2 to 55 mm) of the tumor, the oestrogen (0 to 100) and progesterone receptor status (0 to 100), angioinvasion (yes or no), metastasis (yes or no) and lymphocytic infiltrate (yes or no). A subset of these variables is used to render the shape of the fish glyphs. The categorical data consists in the subjects survival during the following five years (yes or no). This information is used to define the color of the fish.

4.1 Mapping

In Table 1 the mapping of the clinical and categorical features to the parameters of the fish glyph is summarized. In order to enhance the contrast between different fish glyphs the software allows to map a feature to more than one parameter of the fish glyph. Here this is done for the clinical features grade, diameter and age which are mapped to four parameters (two lengths and two arcs) each. Figure 3 shows four visualizations of fish glyphs and illustrates the significance of their shape and color.

Table 1. The mapping of the clinical and categorical features to the parameters of the fish glyph

type	feature	visualization	parameter	fish glyph parameter
clinical	grade	shape	$x_0 \dots x_3$	top
clinical	diameter	shape	$x_4 \dots x_7$	bottom
clinical	age	shape	$x_8 \dots x_{11}$	tail (caudal) fin
clinical	lymphocytic infiltrate	shape	x_{15}	top (dorsal) fin
clinical	angioinvasion	shape	x_{16}	lower (pectoral) fin
category	survival ≤ 5 years	color	x_{12}	red
category	survival > 5 years	color	x_{13}	green

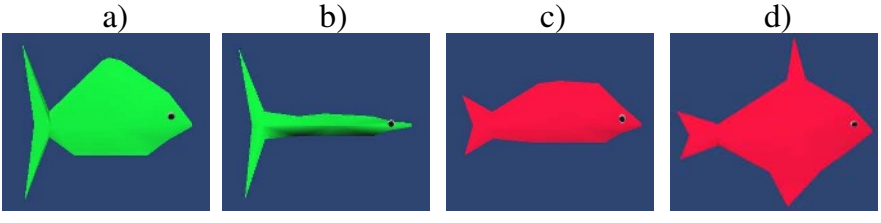


Fig. 3. Fish **a)** represents a subject with a high age (large tail fin), who survived the following five years (green color). There is no lymphocytic infiltrate (no top fin) and no angioinvasion (no lower fin). The tumor has grade III (huge top) and has a medium diameter (medium bottom). Fish **b)** represents a subject with a high age (large tail fin), who survived the following five years (green color). There is no lymphocytic infiltrate (no top fin) and no angioinvasion (no lower fin). The tumor has grade I (small top) and has a tiny diameter (tiny bottom). Fish **c)** represents a subject with a low age (small tail fin), who died within the following five years (red color). There is no lymphocytic infiltrate (no top fin) and no angioinvasion (no lower fin). The tumor has grade II (medium sized top) and has a small diameter (small bottom). Fish **d)** represents a subject with a low age (small tail fin), who died within the following five years (red color). There is a lymphocytic infiltrate (top fin) and angioinvasion (lower fin). The tumor has grade III (large top) and has a large diameter (large bottom).

4.2 Results

The SOM is trained with one million training steps whereas a linear decreasing neighborhood and learning rate are used. Preliminary experiments with SOMs of sizes between 5×5 and 100×100 trained on the microarray data set with 70 genes revealed the best results for a SOM of size 15×15 using visual inspection. The SOM result could probably be further improved by fine-tuning of the parameters and by using objective measures for the map organization and topology preservation [35]. Fish glyphs are integrated to represent the clinical and categorical data. The clinical data is used to render the shape of the fish and the categorical data specifies the color of the fish. A flight into the computed reef SOM is shown in Figure 2. In Figure 4 results from an exploratory data analysis are described.

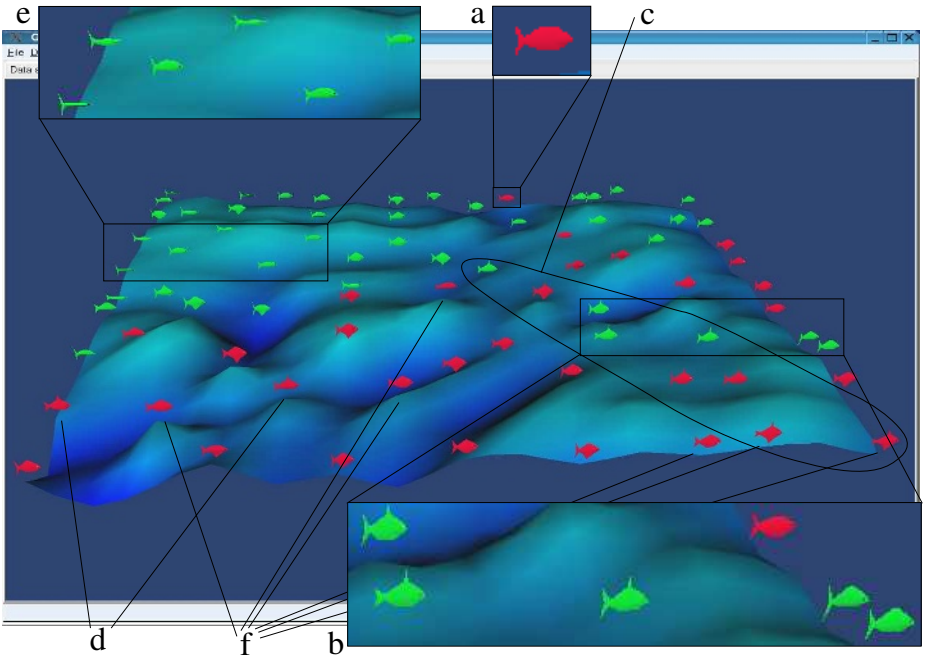


Fig. 4. Subjects who survived the following five years (green fish) are separated from those who died within the following five years (red fish), except a few outliers (**a** and **b**). The single red outlier (**a**) is placed in a region dominated by green fish indicating that its gene expression profile might be similar to those subjects who survived the following five years. Also its clinical features do not indicate any deviation from the surrounding green fish. The group of outliers (**b**) consists of five fish. All of them have tumor grade III (huge top) and three of them have lymphocytic infiltrate (top fin). Interestingly subjects with lymphocytic infiltrate (top fin) cluster together (**c**) except two outliers (**d**). This indicates that the gene expression profile of subjects with lymphocytic infiltrate might be similar. The upper left half is dominated by subjects who survived the following five years. Many of them have a tumor of grade I or II (small or middle size top with a small tumor diameter (small bottom) (**e**). In contrast to that most of the subjects in the front have tumor grade III (huge top). Many subjects who died within the following five 5 years (red fish) are still young (small tail fin) (**f**).

5 Summary and Discussion

The reef SOM [10], a metaphoric display, is applied and further improved such that it allows the simultaneous display of biomedical multi-modal data for an exploratory analysis. Visualizations of microarray, clinical, and category data are combined in one informative and entertaining image. The U-matrix of the SOM trained on microarray data is visualized as an underwater sea bed using color and texture. The clinical data and category data are integrated in the form of fish shaped glyphs. The color represents a category (the main information one is interested in) and the shape is modified according to selected clinical features.

In order to compare the reef SOM with other data analysis approaches a test scenario is imaginable where test persons are asked to detect structures and patterns in either artificial or real-life data sets.

The reef SOM has the fundamental advantage that it is multi-modal itself, and thus expecially well suited for the display of multi-modal data. The *geology modus* (U-matrix displayed as sea bed) is combined with a *fauna modus* (fish glyphs) or *fauna modi* (fish shape representing the clinical data and fish color representing the category). The user can direct his attention to the modus of his choice or to both. Additional modi allow the integration of further data sources, e.g a *flora modus* might be introduced for displaying features of biomedical images (X-ray, CT, MRI).

We believe that the reef SOM is well suited for the exploratory data analysis of multi-modal data since its ability to combine visualizations of microarray, clinical and category data. The resulting images are intuitive, entertaining and can easily be interpreted by the biomedical collaborator, since specific knowledge about the SOM algorithm is not required. Visual inspection enables the detection of interesting structural pattern in the multi-modal data when browsing through and zooming into the image.

Acknowledgments. A first prototype of the system has been presented on the 5th Workshop on Self-Organizing Maps (WSOM 2005). A detailed description of the visualization technique has been submitted to Brain Minds & Media online.

References

1. Quackenbush, J.: Computational analysis of microarray data. Nat Rev Genet **2**(6) (2001) 418–27
2. Ochs: Microarray in cancer: Research and applications. Biotech. **34** (2003) 4–15
3. van't Veer, L.J., Dai, H., van de Vijer, M.J., Y D He, A.A.M.H., Mao, M., Peterse, H.L., van der Kooy, K., Marton, M.J., Witteveen, A.T., et al.: Gene expression profiling predicts clinical outcome of breast cancer. Nature **415** (2002) 530–6
4. Brennan, D.J.: Application of DNA microarray technology in determining breast cancer prognosis and therapeutic response. Expert opinion on biological therapy **5**(8) (2005) 1069–83
5. Dettling, M., Buehlmann, P.: Finding predictive gene groups from microarray data. Journal of Multivariate Analysis **90**(1) (2004) 106–31
6. Kohonen, T.: The self-organizing map. Proc. of the IEEE **78**(9) (1990) 1464–80
7. Tamayo, P., Slonim, D., Medirov, J., et al.: Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. PNAS **96** (1999) 2907–12
8. Wang, J., et al.: Clustering of the SOM easily reveals distinct gene expression patterns: results of a reanalysis of lymphoma study. BMC Bioinf. **3**(36) (2002)
9. Ultsch, A.: Self organizing neural networks for visualization and classification. In et al., O., ed.: Information and Classification, Springer (1993) 307–13
10. Nattkemper, T.W.: The som reef - a new metaphoric visualization approach for self organizing maps. WSOM (2005)
11. Kohonen, T.: Self-Organization and Associative Memory. Springer (1989)

12. Kohonen, T.: Self Organizing Maps. Springer-Verlag, Berlin (2001)
13. Vesanto, J.: Som-based visualization methods. Intell. Data Anal. **3** (1999) 111–26
14. Vesanto, J., Alhoneimi, E.: Clustering of the self-organizing map. IEEE Transactions on Neural Networks **11** (2000) 586–600
15. Wu, S., Chow, T.W.S.: Clustering of the self-organizing map using a clustering validity index based on inter-cluster and intra-cluster density. Pattern Recognition **37** (2004) 175–188
16. Yang, C.C., Chen, H., Hong, K.K.: Visualization tools for self-organizing maps. In: Proc. of the 4th ACM conf. on Digital libraries. (1999) 258–9
17. Honkela, T., Kaski, S., Lagus, K., Kohonen, T.: Websom - self-organizing maps of document collections. In: Proc. of WSOM. (1997)
18. Kaski, S., Nikkilä, J., Kohonen, T.: Methods for interpreting a self-organized map in data analysis. In: Proc. of ESANN. (1998)
19. Rauber, A., Merkl, D.: Automatic labeling of self-organizing maps for information retrieval. JSRIS **10**(10) (2001) 23–45
20. du Toit, S., Steyn, A., et al.: Graphical exploratory data analysis. Springer (1986)
21. Siegel, J., Farrell, E., Goldwyn, R., Friedman, H.: The surgical implication of physiologic patterns in myocardial infarction shock. Surgery **72** (1972) 126–41
22. Hartigan, J.: Printergraphics for clustering. ournal. of Statistical Computing and Simulation **4** (1975) 187–213
23. Ribarsky, M., Ayers, E., Eble, J., Mukherjea, S.: Glyphmaker: Creating customized visualizations of complex data. IEEE Computer **27**(7) (1994) 57–64
24. Kraus, M., Ertl, T.: Interactive data exploration with customized glyphs. In Skala, V., ed.: WSCG 2001 Conference Proceedings. (2001)
25. Shaw, C.D., Hall, J.A., Blahut, C., Ebert, D.S., Roberts, D.A.: Using shape to visualize multivariate data. In: Workshop on New Paradigms in Information Visualization and Manipulation. (1999) 17–20
26. Spoerri, A.: Infocrystal: a visual tool for information retrieval & management. In: Proceedings of the second international conference on Information and knowledge management, Washington, D.C., United States, ACM Press (1993)
27. Chernoff, H.: The use of faces to represent points in n-dimensional pace graphically. Technical Report RN NR-042-993, Dept. of Stat., Stanford Univ. (1971)
28. Noh, J.y., Neumann, U.: A survey of facial modeling and animation techniques. Technical Report 99-705, USC Technical Report (1998)
29. Dorling, D.: Cartograms for visualizing human geography. In Hearnshaw, H.M., Unwin, D.J., eds.: Visualization in geographical Information Systems, Chichester, John Wiley & Sons (1994) 85–102
30. Alexa, M., Müller, W.: Visualization by metamorphosis. In Wittenbrink, C.M., Varshney, A., eds.: IEEE Visualization 1998 Late Breaking Hot Topics Proceedings. (1998) 33–36
31. Smith, M., Taffler, R., White, L.: Cartoon graphics in the communication of accounting information for management decision making. Journal of Applied Management Accounting Research **1**(1) (2002) 31–50
32. Pickett, R.M., Grinstein, G.G.: Iconographics displays for visualizing multidimensional data. In: Proc. IEEE Conf. on Systems, Man, and Cybernetics. (1988) 514–9
33. Kleiner, B., Hartigan, J.: Representing points in many dimension by trees and castles. J. Am. Stat. Ass. **76** (1981) 260–9
34. Chua, M., Eick, S.: Information rich glyphs for software management. IEEE Computer Graphics and Applications **18** (1998) 24–9
35. Venna, J., Kaski, S.: Local multidimensional scaling with controlled tradeoff between trustworthiness and continuity. WSOM (2005)

Semi-supervised Significance Score of Differential Gene Expressions

Shigeyuki Oba and Shin Ishii

Graduate School of Information Science, Nara Institute of Science and Technology
shige-o@is.naist.jp

Abstract. In gene expression analyses for DNA microarray data, various statistical scores have been proposed for evaluating significance of genes exhibiting differential expression between two or more controlled conditions. To consider an unsupervised case or a semi-supervised case rather than a well-studied supervised case, we assume a latent variable model and apply the optimal discovery procedure (ODP) proposed by Storey (2005) to the model. Theoretical consideration leads to two different interpretations of the hidden variable, i.e., it only implicitly affects the alternative model through the model parameters, or is explicitly included in the alternative model, so that they correspond to two different implementations of ODP. By comparing the two implementations through experiments with simulation data, we found that sharing the latent variable estimation as in the latter case is effective in increasing the detectability of truly active genes. We also propose unsupervised and semi-supervised rating of genes and show its effectiveness as a significance score.

1 Introduction

Selecting significant genes is an important task in gene expression analyses typically by means of DNA microarray technology. Significance of each gene is usually defined as differential expression between different conditions. Here, we call a gene 'active' when its expression has relationship to the biological conditions of interest, or 'inactive' when there is no relationship. Detecting active genes is a statistical decision problem to minimize risk to make errors, and the statistical significance test framework evaluates the risk to accept or reject the null hypothesis that the gene is inactive, namely, it has no relation to the difference between the conditions.

Considering multiplicity is crucial in statistical tests dealing with thousands of genes, and sharing commonality among multiple tests is an important issue in recent years, in order to increase the detectability from highly-correlatedly expressed genes. Significance analysis of microarray (SAM) [5] considered a shrinkage estimation of intra-class variance to improve stability, and derived such a simple score that it has become very popular today. Empirical Bayes score [1] assumed a hierarchical Bayes model in which priors of parameters for each gene were also estimated using expression of multiple genes. Storey (2005) [3,4] extended the Neyman-Pearson's lemma [2] to be applicable to multiple tests and

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 808–817, 2006.

proposed a new framework, optimal discovery procedure (ODP). When null and alternative hypotheses are not simple, i.e., they have their own parameters to be estimated by statistical inference like maximum likelihood, ODP can improve the gene discovery accuracy by sharing the estimated parameters among the multiple tests.

In this study, we apply the ODP framework to a Gaussian mixture model for the 'active' genes' expression, which includes a hidden variable that indicates a class label of conditions which each sample belongs to. We discuss two important but different ways to deal with the hidden variable in our model, i.e., the estimated hidden variable may or may not be shared among multiple tests as well as the estimated parameters. We compare these two variations through a simulation and show that sharing commonality in the hidden variable improves the detection accuracy of active genes. This model leads to new significance scores of genes; an unsupervised significance score which does not consider class labels, and a semi-supervised significance score which considers both types of samples, with and without class labels.

2 Neyman-Pearson's Lemma and Supervised Differential Gene Significance Score

Our objective is to decide accurately whether a gene i is active or inactive, according to expression data $\boldsymbol{x}_i = (x_{i1}, x_{i2}, ..., x_{iM})$ and label information $Y = (y_1, ..., y_M), y_j \in \{1, 2\}, j = 1, ..., M$, over M measurements (samples), where there are some possibly different conditions, typically two like in this setting.

According to the conventional framework of statistical testing, let the null hypothesis that the gene is actually inactive, be denoted as H_0, and the alternative hypothesis that the gene is actually active, as H_1. Our question is what kind of significance score S is the best to discriminate active genes from inactive genes based on a limited number of measurements like in usual microarray experiments.

When the null and the alternative models are represented as null and alternative probability density functions (pdf's), $f(X_i) = p(X_i|H_0)$ and $g(X_i) = p(X_i|H_1)$, respectively; namely, they are simple hypotheses with no variable parameter, the following likelihood ratio score is known as the most powerful score of significance:

$$S_{\mathrm{LR}}(X_i) = \frac{g(X_i)}{f(X_i)}, \tag{1}$$

which was stated and proven as Neyman-Pearson's lemma [2].

However, many useful statistical tests assume non-simple hypothetical models so that the null and alternative pdf's include variable parameters to be estimated statistically. For example, in a typical way for supervised differential gene discovery, the null and the alternative models are defined as

$$H_0 : f(X_i; \phi_i) = \prod_j N(x_{ij}|0, \sigma_{0i}^2), \quad H_1 : g(X_i; \theta_i) = \prod_j N(x_{ij}|\mu_i(y_j), \sigma_{1i}^2), \tag{2}$$

where $\phi_i = \{\sigma_{0i}^2\}$ and $\theta_i = \{\mu_i(1), \mu_i(2), \sigma_{1i}^2\}$ are the parameters of the null and alternative models, respectively. $N(x|\mu, \sigma^2)$ denotes a normal density function with a mean μ and a variance σ^2. $\mu_i(1)$ and $\mu_i(2)$ are centers of normal distributions for $y_j = 1$ (class 1) and $y_j = 2$ (class 2), respectively. σ_{0i}^2 and σ_{1i}^2 are intra-class variances under the null and alternative hypotheses, respectively. We assume that the expression data are normalized such that $(1/M)\sum_{j=1}^{M} x_{ij} = 0$ holds, for description simplicity.

In this case, the log likelihood ratio can define a significance score of a single gene i:

$$S_{LR-S}(X_i) = \sum_{j=1}^{M} \left(\ln N(x_{ij}|\mu_i(y_j), \sigma_{1i}^2) - \ln N(x_{ij}|0, \sigma_{0i}^2) \right), \tag{3}$$

where the maximum likelihood estimates (mle's) of the parameters are given as $\hat{\mu}_i(k) = \frac{\sum_{j=1}^{M} x_{ij} I(y_j=k)}{\sum_{j=1}^{M} I(y_j=k)}$, $(k = 1, 2)$, $\hat{\sigma}_{0i}^2 = \frac{1}{M}\sum_{j=1}^{M} x_{ij}^2$, and $\hat{\sigma}_{1i}^2 = \frac{1}{M}\sum_{j=1}^{M}(x_{ij} - \hat{\mu}_i(y_j))^2$. Here, $I(A)$ is an index function which outputs 1 when condition A is satisfied, otherwise 0. By assigning the mle into the significance score, the following estimated log likelihood ratio function is available as a score:

$$\hat{S}_{\mathrm{LR-S}}(X_i) = \sum_{j=1}^{M} \left(\ln N(x_{ij}|\hat{\mu}_i(y_j), \hat{\sigma}_{1i}^2) - \ln N(x_{ij}|0, \hat{\sigma}_{0i}^2) \right) = \frac{1}{2} \ln \left(\hat{\sigma}_{0i}^2 / \hat{\sigma}_{1i}^2 \right), \tag{4}$$

which is often called an S/N ratio.

3 Unsupervised and Semi-supervised Likelihood Model

3.1 Unsupervised Differential Gene

For an active gene, the gene expression is assumed to relate to the condition of each sample, namely, it is 'on' in samples in certain conditions and 'off' in the others. Let a hidden variable $Z_i = (z_{i1}, ..., z_{iM}), z_{ij} \in \{1, 2\}, j = 1, ..., M$ denote whether the gene i is 'on', $z_{ij} = 1$, or 'off', $z_{ij} = 2$, in a sample j. We in particular consider a binary categorization of conditions, but allow uncertainty in conditions, namely, the supervised label y takes either of 1 (class 1), 2 (class 2), or 0 (label unknown). An unsupervised case is defined as being $y = 0$ for all j, and a semi-supervised case is defined as the y values being zero for some j.

We assume a mixture of normal distributions for active gene i's expression x_{ij} in a sample j:

$$\begin{aligned} p(x_{ij}, |\theta_i) &= \sum_{k=1}^{2} p(z_{ij} = k|\theta_i) p(x_{ij}|z_{ij} = k, \theta_i \\ &= \sum_{k=1}^{2} \nu(k) N(x_{ij}|\mu(k), \sigma_1^2), \end{aligned} \tag{5}$$

where $\nu(k)$ is the prior probability that the latent variable z_{ij} takes $k \in \{1, 2\}$. We assume $\nu(k)$ is independent of i or j. Under the alternative hypothesis H_1, the likelihood function of the parameter $\theta_i = (\nu(1), \nu(2), \mu(1), \mu(2), \sigma_1^2)$, given the expression vector $X_i \equiv \{x_{i1}, \cdot, x_{iM}\}$ of gene i, is given by

$$g(X_i; \theta_i) = \prod_{j=1}^{M} \sum_{k=1}^{2} \nu_k N(x_{ij} | \mu(k), \sigma_1^2). \tag{6}$$

The null hypothesis is given as the same as in the supervised case, i.e., a normal distribution:

$$H_0 : \quad f(X_i; \phi_i) = \prod_j N(x_{ij} | 0, \sigma_0^2). \tag{7}$$

The log-likelihood ratio score is then defined as

$$\hat{S}_{\mathrm{LR-U}}(X_i) = \ln \frac{g(X_i; \hat{\phi}_i)}{f(X_i; \hat{\theta}_i)}, \tag{8}$$

where $\hat{\phi}_i = \arg\max_{\phi_i} f(X_i; \phi_i)$ and $\hat{\theta}_i = \arg\max_{\theta_i} g(X_i; \theta_i)$ are mle's for the two hypotheses.

When we have an mle $\hat{\theta}$ for the alternative hypothesis, a posteriori probability of the hidden variable, given a datum x_{ij}, is obtained as

$$\hat{z}_{ij}(k) \stackrel{\mathrm{def}}{=} P(z_{ij} = k | x_{ij}, \hat{\theta}_i) \frac{\nu(k) N(x_{ij} | \mu(k), \sigma_1^2)}{\sum_{k=1}^{2} \nu(k) N(x_{ij} | \mu(k), \sigma_1^2)}. \tag{9}$$

Let $\hat{Z}_i = (\hat{z}_{ij}(k); j = 1, ..., M,\ k = 1, 2)$ be the vector of the estimated posterior, where $\hat{z}_{ij}(k) \geq 0$ and $\sum_k \hat{z}_{ij}(k) = 1$ holds. It will be used for considering multiple testing later.

3.2 Semi-supervised Differential Gene

In a semi-supervised case, an active gene can be modeled by modifying the prior of the hidden variable into incorporating label-unobservability:

$$P(z_{ij} = k | y_j, \theta_i) = I(y_j = k) + I(y_j = 0)\nu(k),\ k = 1, 2. \tag{10}$$

This prior states the hidden label is the same as the observed one when it is observed, but is predominantly predicted by the prior knowledge $\nu(k)$ when it is unobserved; the point is that we regard the observation as a prior information. Accordingly, a semi-supervised case is dealt with by simply employing the same prior as in the unsupervised case, equation (10), but the first term vanishes because $I(y_j = 0) = 1$ for every j.

4 Optimal Discovery Procedure

4.1 ODP Lemma and Its Application

According to the Neyman-Pearson's lemma, a statistic is defined as most powerful when its detection probability β is the largest with a fixed significance rate α, and the likelihood ratio was proven to be most powerful when the null and alternative hypotheses are both simple. Storey [3,4] extended the Neyman-Pearson's framework to multiple testing and proposed a new general framework, ODP. They defined that a statistic is optimal if it maximizes the expected true positive (ETP) rate when fixing the expected false positive (EFP) rate. They also showed that an ODP function is available as in the following ODP lemma, which is not the same as the likelihood ratio.

ODP lemma. *Let S_{ODP} be a common statistic, called an ODP function, for all genes $i = 1, ..., M$:*

$$S_{\mathrm{ODP}}(X) = \frac{\sum_{i' \in G_1} g(X|\theta_{i'})}{\sum_{i' \in G_0} f(X|\phi_{i'})}, \tag{11}$$

where X is a gene expression vector (its dimensionality corresponds to the number of samples) of a gene. Then, the criterion that the gene is significant when $S_{\mathrm{ODP}}(X_i) > \lambda$ for any threshold $\lambda > 0$ is an ODP. In equation (11), G_0 and G_1 are index sets of inactive and active genes, respectively.

This lemma defines an ideal ODP but not a practical one, because it needs information not available in actual situations of multiple testing. True values of parameters θ and ϕ are not available and they are substituted by mle's estimated from the observed data. G_0 and G_1 are not available in a real situation either, because there is no need to calculate significance scores if we know G_0 and G_1. As a practical use, therefore, they proposed an approximate ODP criterion:

$$\hat{S}_{\mathrm{ODP}}(X_i) = \frac{\sum_{i'} g(X_i|\hat{\theta}_{i'})}{\sum_{i' \in \hat{G}_0} f(X_i|\hat{\phi}_{i'})}, \tag{12}$$

where $\hat{\theta}_i = \arg\max_{\theta_i} g(X_i|\theta_i)$ and $\hat{\phi}_i = \arg\max_{\phi_i} f(X_i|\phi_i)$ are the mle's. Note here that the summation in the numerator is taken over all genes, and that in the denominator is taken for a roughly estimated set of inactive genes, $\hat{G}_0$. As an example, they proposed $\hat{G}_0 = \{j | g(X_j|\hat{\theta}_i)/f(X_j|\hat{\phi}_i) > \epsilon\}$, i.e., a set of genes which are not significant by the standard gene-wise likelihood ratio test, where ϵ is an arbitrary positive threshold.

The key of the ODP, in contrast to the individual likelihood ratio, is that the ODP shares among all genes common information about distribution represented as null and alternative models, so that the common information is used when evaluating a single gene. They actually showed that when the hypothetical models have some global characters shared by the genes, such as asymmetric and/or cluster structure, the ODP can incorporate them so as to improve the performance of gene discovery, i.e., increasing the ETP for a fixed EFP.

4.2 ODP on Latent Variable Models

For parametric hypothetical models, hypotheses are distributed attributable to the distribution of model parameters. When there are hidden variables as in our model's case, however, another definition of what is the hypothesis is possible, namely, (a) unknown values of hidden variables are marginalized out in the hypotheses, or (b) the hidden variables are explicitly included like the model parameters in the hypothetical models. In the case (b), the distribution of hypotheses is attributed to the distribution of both model parameters and hidden variables. These two definitions of the hypothesis distribution lead to different results from each other. We formalize these two cases in this section and compare them in the next section.

In the likelihood ratio score, we evaluate the ratio of the two likelihood functions, $g(\cdot|\hat{\theta}_{i'})$ and $f(\cdot|\hat{\phi}_{i'})$, for each gene i', where we estimate the parameters of null and alternative models, $\hat{\theta}_{i'}$ and $\hat{\phi_{i'}}$, respectively, as mle's, and the hidden variable in the alternative model, as its posterior, $\hat{Z}_{i'}$. In the ODP, only a single significance function, (12), is constructed by using a set of hypotheses each corresponding to a single gene, and hence the likelihood functions, $g(\cdot|\hat{\theta_{i'}})$ and $f(\cdot|\hat{\phi_{i'}})$, for gene i' are shared by all genes. Namely, in the ODP, we should evaluate $g(X_i|\hat{\theta_{i'}})$ and $f(X_i|\hat{\phi_{i'}})$ for $i \neq i'$, and the key difference between the cases (a) and (b), appears in the way to do this process.

In case (a), i.e., when only the model parameters are shared by all genes, the likelihood function becomes

$$\hat{g}(X_i|\hat{\theta}_{i'}) = \prod_{j=1}^{M} p(x_{ij}|\hat{\theta}_{i'}) = \prod_{j=1}^{M} \sum_{k=1}^{2} \hat{\nu}(k) N(x_{ij}|\hat{\mu}_{i'}(k), \hat{\sigma}^2_{1i'}). \tag{13}$$

In case (b), i.e., when the estimated posterior of the hidden variable, $\hat{Z}_{i'}$, is also shared, another likelihood function is given as

$$\hat{g}(X_i|\hat{\theta}_{i'}, \hat{Z}_{i'}) = \prod_{j=1}^{M} p(x_{ij}|\hat{z}_{i'j}, \hat{\theta}_{i'}) = \prod_{j=1}^{M} \sum_{k=1}^{2} \hat{z}_{i'j}(k) N(x_{ij}|\hat{\mu}_{i'}(k), \hat{\sigma}^2_{1i'}). \tag{14}$$

As we pointed at the beginning of this section, the difference between the two cases above comes from the difference in the interpretation of what the model is and what the hypothesis is. In fact, if one assumes that the posterior of hidden variable $\hat{Z}_{i'}$ is a part of the unknown parameter $\theta_{i'}$, the second case is a special case of the original ODP. Namely, in case (b), the alternative hypothesis of a single gene is dependent on expression distributions of the two groups of samples, while that in case (b) is dependent not only on expression distributions but also on class labels of the samples. When we ignore the multiplicity of statistical testing, these two cases are identical, because the hidden variable is uniquely determined, even probabilistically, from the estimated parameter of the single gene model.

The model in case (b) may have a biological meaning. The hidden variable vector Z_i can be regarded as an on/off (binary) pattern vector of gene i over the

samples. Since some biology such as gene regulatory factors may be represented as characteristic distribution of binary pattern vectors, the ODP framework will be able to utilize such characteristic and possibly biological information by making it shared by multiple tests.

5 Numerical Experiment

5.1 Unsupervised Differential Gene Discovery

First of all, we compare various unsupervised scores devised for detection of significant genes.

Figure 1(A) shows a structure of an artificial data set consisting of expression of 6400 genes for 16 samples. The 6400 genes are made up of 3200 inactive ((1) and (2)) and 3200 active ((3) and (4)) genes. Expressions of inactive genes are generated from normal distribution with mean 0 and variance 1.2^2 or 0.8^2 for gene group (1) or (2), respectively. Since they are generated from a single distribution regardless of the sample index, they are in fact inactive. Expressions of active genes are generated from normal distribution with mean μ or $-\mu$ for highly or lowly expressed genes in each sample, respectively, and a common variance 1.0. The expression pattern for each gene is different between groups (3) and (4); in group (3), there are eight subgroups of 200 genes and each subgroup has a high/low pattern different from the other subgroups, and in group (4), 1600 genes have the same high/low pattern. These situations represent (4) all the genes reflect a common biology leading to similar expressions over all samples, and (3) there are eight gene clusters each of which reflects its own biology.

We compared three scores listed below:

- $\hat{S}_{\mathrm{LR-U}}$, a gene-wise likelihood ratio, which is independent of the other genes;
- $\hat{S}_{\mathrm{ODPp-U}}$, an ODP based on the case (a) model which shares estimation of model parameters; and,
- $\hat{S}_{\mathrm{ODP-U}}$, an ODP based on the case (b) model which shares estimation of both model parameters and hidden variables.

Figures 1(B) and 1(C) show the results. In Fig. 1(C), we can see $\hat{S}_{\mathrm{ODP-U}}$ outperformed the others, i.e., it achieved the best sensitivity for each specificity, and $\hat{S}_{\mathrm{LR-U}}$ was the worst. From the histogram in Fig. 1(B), the $\hat{S}_{\mathrm{LR-U}}$ score is found to be insensitive to the difference between the groups (1) and (2), or between (3) and (4). On the other hand, the score $\hat{S}_{\mathrm{ODPp-U}}$ behaves differently between (1) and (2); namely, genes in group (1), which exhibit expressions with larger variance, tended to be evaluated as more significant, because a larger variance compared to a typical variance of inactive genes likely causes a high chance to detect active (possibly false) genes. However, the $\hat{S}_{\mathrm{ODPp-U}}$ was also insensitive to the difference between (3) and (4) which have the same expression distribution. The principal character of the $\hat{S}_{\mathrm{ODP-U}}$ score is that it extracted a larger number of active genes in group (4). Since the 1600 genes in group (4) have the identical pattern of true hidden variables, the commonality boosted the

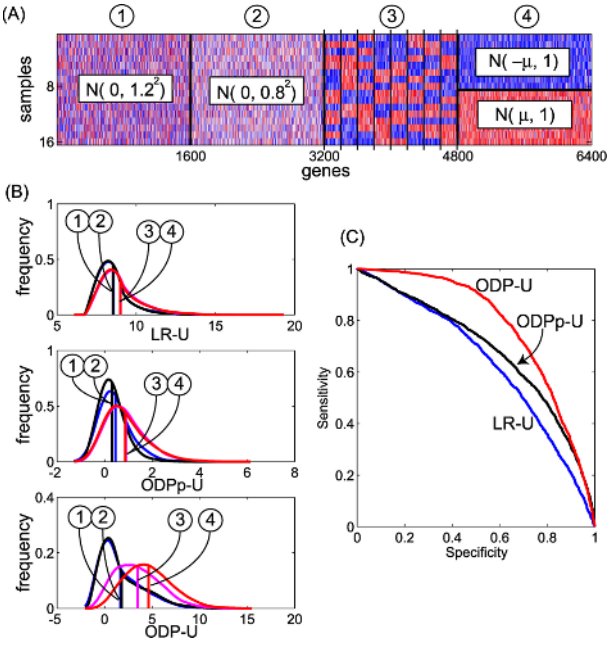


Fig. 1. Unsupervised extraction of significant genes. (A) An artificial data set and its generative models. See text for detail. (B) Histogram of three unsupervised significance scores: LR-U, ODPp-U, and ODP-U, which are separately described for the four gene groups. A vertical line denotes the mean of distribution. (C) ROC curves of active gene detection by changing the threshold for each score; the horizontal and vertical axes denote specificity, (true negative/(true negative + false positive)), and sensitivity, (true positive/(true positive + false negative)), respectively.

significance scores of genes in the same group. In group (3), the number of genes sharing the true hidden variables was 200, which led to weaker boosting of the significance scores than in group (4).

These results show that sharing the estimation of both the parameter and the hidden variable is effective in gene discovery, when they have some structures that can be embossed by making multiple tests cooperative. In addition, we have found that the more genes there are in the same group, the more effective sharing of hidden variable estimation becomes.

5.2 Semi-supervised Differential Gene Discovery

Next, we consider a semi-supervised case. We assumed eight samples (index: 1,2,...,8) have the true class label 1 and the other eight samples (index: 9,10,...,16) have the class label 2, in the 16 samples above. For each class, five samples were correctly labeled but the others' labels were unknown. We compared the four scores listed below:

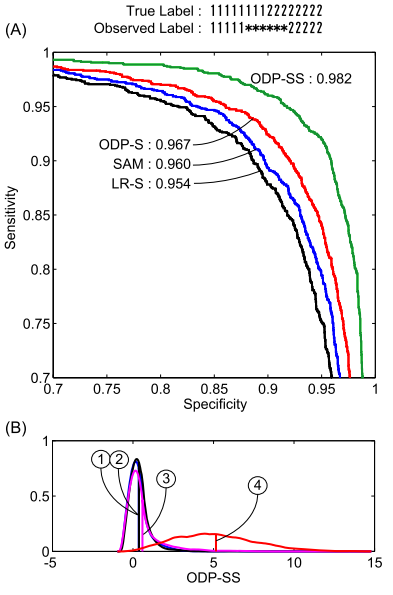


Fig. 2. Semi-supervised extraction of significant genes from the same artificial data as in Figure 1(A). The true labels of the 16 samples are 1 for the earlier 8 samples and 2 for the rest, but for three samples of the both classes are assumed to be unknown (labeled as * in the title of the figure). This experiment evaluates how many genes in the truly active 1600 genes of group (4) can be detected by four significance scores: LR-S, SAM, ODP-S, and ODP-SS. Note that the genes in group (3) are regarded as inactive in this experiment because the high/low patterns are different from the true label pattern. Therefore, there are 4800 inactive genes (groups (1), (2), and (3)) and 1600 active genes (group (4)). (A) ROC curves of the four scores. The number for each score denotes area under the curve (AUC). (B) Distributions of the ODP-SS scores for the four gene groups, (1), (2), (3), and (4).

- S_{SAM}, SAM statistics;
- $\hat{S}_{\mathrm{LR-S}}$, likelihood ratio score;
- $\hat{S}_{\mathrm{ODP-S}}$, an ODP using only the samples with labels; and,
- $\hat{S}_{\mathrm{ODP-SS}}$, an ODP using all samples with the labels being known or unknown.

The three supervised scores, S_{SAM}, $\hat{S}_{\mathrm{LR-S}}$, and $\hat{S}_{\mathrm{ODP-S}}$, used the ten samples with labels to calculate their gene scores, while $\hat{S}_{\mathrm{ODP-SS}}$ used, in addition, the remaining six unlabeled samples to estimate the unknowns in the hypothesis models.

The ROC curves in Figure 2(A) clearly show that $\hat{S}_{\mathrm{ODP-SS}}$ achieved the best specificity for every sensitivity. Although S_{SAM} showed a better detectability than the conventional likelihood ratio, $\hat{S}_{\mathrm{ODP-S}}$ exhibited further better. Figure 2(B) shows the distributions of $\hat{S}_{\mathrm{ODP-SS}}$ for the four gene groups, (1), (2), (3), and (4). The active genes in group (4) were evaluated clearly as more significant than in the others by our $\hat{S}_{\mathrm{ODP-SS}}$ score. The score distribution of gene group

(3) is interesting. Although the distribution of expressions of inactive group (3) was identical to that of active group (4), the score distribution of group (3) is quite different from (4), but instead, similar to those of (1) and (2), which are also inactive with respect to the biology represented as the true label pattern. Even when the labeling includes uncertainty, the information of labels affected the significance of genes, which is stronger than the information of the expression distribution.

6 Concluding Remarks

In this study, we showed an improvement of gene significance criteria in unsupervised and semi-supervised cases. Both of gathering samples and labeling samples by clinical researchers require large amount of time, money, and other costs. A semi-supervised case, which incorporates unlabeled samples, will be useful for gene selection in such a situation. Unsupervised score may be used as an alternative way of filtering insignificant genes out as a pre-process of gene expression analyses.

Our experiments are currently made using artificial data set. For real data sets, however, it is not easy to show clearly the effectiveness of gene ranking, because, in many actual cases, we have no mean to know whether a single gene is truly active or not. We have made an experiment to compare various scores and found that semi-supervised score can improve the concordance between rankings calculated from a large real data set and its smaller data subset, by using unlabeled data in addition to labeled data; the results will be shown elsewhere.

Various latent variable models such as mixture of t-distribution model, Cox's proportional hazards model, and so on, may lead to useful gene ranking within our framework, and such an application would be our future work.

References

1. Bradley Efron and Robert Tibshirani. Empirical bayes methods and false discovery rates for microarrays. *Genet Epidemiol*, 23(1):70–86, Jun 2002.
2. J. Neyman and E. S. Pearson. On the problem of the most efficient test of statistical hypotheses. *Philosophical Transactions of the Royal Society*, 231:289–337, 1933.
3. J.D. Storey. The optimal discovery procedure: A new approach to simultaneous significance testing. *UW Biostatistics Working Paper Series*, Working Paper 259, 2005.
4. J.D. Storey, J.Y. Dai, and J.T. Leek. The optimal discovery procedure for large-scale significance testing, with applications to comparative microarray experiments. *UW Biostatistics Working Paper Series*, Working Paper 260, 2005.
5. V G Tusher, R Tibshirani, and G Chu. Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Natl. Acad. Sci. USA*, 98(9):5116–5121, Apr 2001.

Semi Supervised Fuzzy Clustering Networks for Constrained Analysis of Time-Series Gene Expression Data

Ioannis A. Maraziotis, Andrei Dragomir, and Anastasios Bezerianos

Department of Medical Physics, Medical School, University of Patras,
26500 Rio, Greece
{imarazi, adragomir}@heart.med.upatras.gr,
bezer@patreas.upatras.gr

Abstract. Clustering analysis of time series data from DNA microarray hybridization studies is essential for identifying biological relevant groups of genes. Microarrrays provide large datasets that are currently primarily analyzed using crisp clustering techniques. Crisp clustering methods such as K-means or self organizing maps assign each gene to one cluster, thus omitting information concerning the multiple roles of genes. One of the major advantages of fuzzy clustering is that genes can belong to more than one group, revealing this way more profound information concerning the function and regulation of each gene. Additionally, recent studies have proven that integrating a small amount of information in purely unsupervised algorithms leads to much better performance. In this paper we propose a new semi-supervised fuzzy clustering algorithm which we apply in time series gene expression data. The clustering that was performed on simulated as well as experimental microarray data proved that the proposed method outperformed other clustering techniques.

1 Introduction

DNA microarray technology enables the rapid and efficient measurement of expression levels of a large number of genes in a simultaneous manner, thus providing a means of detecting specific patterns of gene expression in a cell, as well as enabling the extraction of crucial information regarding the conjunctive functioning of genes [1]. Initial computational efforts employed classical clustering techniques (see ref [2] for an extensive overview) for grouping genes according to their expression patterns, based on the biologically validated assumption that genes involved in the same biological process exhibit similar patterns of variation. The approach is meant to infer functional category for genes of unknown functionality, based on the labels of already annotated genes. A recent direction is to employ expression clustering as an important step in extracting cluster-representative genes that are further used to reconstruct gene regulatory networks [3].

The present study proposes a novel approach that is able to overcome specific clustering techniques drawbacks. These include the incorporation of supervised information (whenever available) regarding the genes functional category, in order to

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 818 – 826, 2006.

improve the biological accuracy of the derived clusters, as well as the capability of the method to assign genes to multiple clusters (since it is well known that certain genes might be involved in several biological pathways and hence belong to more than one functional category). A third issue that is of high interest is the one of automatically determining the number of clusters in the data (especially in the scenario of identifying cluster-representative genes for regulatory networks reconstruction).

Having in mind the above considerations, we propose a fuzzy partitioning framework based on the Fuzzy Kohonen Clustering Network [4]. The approach allows us to profit from the advantages of fuzzy clustering [5,6], which facilitates the identification of overlapping clusters, thus allowing genes to belong to more than one group. Therefore, each gene is considered to be a member of various clusters, with a variable degree of membership. Additionally, fuzzy logic methods inherently account for noise in the data because they extract trends rather than the precise values. The incorporation of supervised information is accomplished by considering sets of constraints either forcing genes to cluster together or assigning them to different clusters, according to available biological knowledge, fact which significantly improves the accuracy of clustering even when given a relatively small amount of supervision.

2 Methods

2.1 Fuzzy Kohonen Clustering Network

Fuzzy clustering is a partition – optimization technique that aims to group data based on their similarity in a non-exclusive manner by permitting each sample to belong to more than one cluster. Considering a finite set of elements $X = \{x_1, x_{2,\ldots,} x_n\}$, the problem is to perform a partition on this set into c fuzzy sets with respect to a given criterion.

One of the most widely used fuzzy clustering methods is Fuzzy c-means (FCM) due to its efficacy and simplicity [7]. The FCM algorithm partitions a collection of n data points into c fuzzy clusters in such a way as to minimize the following objective function:

$$J\left(u_{ij}, \mathbf{v}_k\right) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m} \left\|\mathbf{x}_j - \mathbf{v}_i\right\|^2, \quad m > 1 \tag{1}$$

where $\mathbf{v}_i$ is the prototype of the i^{th} cluster generated by fuzzy clustering, u_{ij} is the membership degree of the j^{th} data point belonging to the to the i^{th} cluster represented by $\mathbf{u}_i$, $u_{ik} \in U$, U is a *c x n* fuzzy partition matrix which satisfies the constraints:

$$0 < \sum_{j=1}^{n} u_{ij} < n, \ \text{i=1, 2, ..., } c \text{ and } \sum_{i=1}^{c} u_{ij} = 1, \quad j = 1, 2, \ldots, n \tag{2}$$

In (1), *m* is called exponential weight and controls the fuzziness degree of the membership matrix, in [8] there is a study on the value of m and how it influences clustering of microarray data.

On the other hand one of the most common crisp techniques is the well known self organizing map SOM [9], which has been used in numerous fields, the structure of SOM consists of two layers an input layer and an output (competitive) layer. When a new input vector arrives the nodes in the output layer compete with each other and the winner (whose weight has the minimum distance from the input) updates its weights and those of some predefined neighbors. This process is repeated until the weight vector stabilizes.

SOM in its classical form suffer from a number of disadvantages. Firstly, being heuristic procedure its convergence is not based on optimizing any model of the process or its data. Another issue is that usually the final weight vectors depend on the input sequence. Finally, several parameters of the SOM algorithms, such as the learning rate, the size of the update neighborhood function, and the strategy for altering these two parameters during learning must be varied from one data set to another in order to achieve the desirable results.

The Fuzzy Kohonen Clustering Network (FKCN) is an integration of the FCM algorithm into the learning rate as well as the updating strategies of the SOM. Combining FCM, which is an optimization procedure with SOM, which is not, is a way to address several of the aforementioned problems of the classic SOM structure. Additionally FKCN enables SOM to generate continues-valued outputs for fuzzy clustering instead of hard clustering. Another advantage of FKCN is that it gains in computational efficiency over the FCM model by using the linear weight update rule in SOM.

The complete integration of FCM and SOM is done by defining the learning rate a_{ij} for Kohonen updating as follows:

$$a_{ij}(t) = \left(u_{ij}(t)\right)^{m_t}; \quad m_t = m_0 - t\Delta m, \quad \Delta m = \frac{m_0 - 1}{t_{\max}} \tag{3}$$

where m_0 is a positive constant grater than 1, and t_{max} is the maximum number of iterations.

After the computation of the learning rates, the learning algorithm of the FKCN updates the weight vectors following the:

$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) + \frac{\sum_{j=1}^{n} a_{ij}(t)\left[\mathbf{x}_j - \mathbf{v}_i(t)\right]}{\sum_{j=1}^{n} a_{ij}(t)} \tag{4}$$

For large values of m_t all c weight vectors are updated with lower individual learning rates, but as $m_t \rightarrow 1$ an always increasing quantity of the unit sum is given to the winning node. Consequently, FKCN is a non-sequential unsupervised learning algorithm that uses fuzzy membership values from the FCM algorithm as learning rates and therefore realizes control of the learning rate distribution as well as achieving neighborhood updating. In the section that follows we will present a methodology for transforming FKCN to adapt the advantages of a semi-supervised algorithm, while at the same time keep all the characteristics reported so far.

2.2 Semi Supervised FKCN

We consider a framework where prior knowledge on a specific domain is given on sets of either must-link or cannot-link constraints or both. Let **E** be the set of must-link constraints to be given in pairs $(\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{E}$ where the instances $\mathbf{x}_i$ and $\mathbf{x}_j$ should be assigned to the same cluster, while cannot-link constraints in pairs $(\mathbf{x}_i, \mathbf{x}_j) \in \Delta$ where Δ is the set of cannot-link constraints and $\mathbf{x}_i$, $\mathbf{x}_j$ should be assigned to different clusters.

In SS-FKCN, we have modified the objective function by including a cost term for constraint violation, in order to guide the algorithm towards an appropriate partitioning, taking in this way advantage of the given sets of paired constraints. Thus following (1) the objective function of SS-FKCN has the form of:

$$J\left(u_{ij}, \mathbf{v}_k\right) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m} \left\|\mathbf{x}_j - \mathbf{v}_i\right\|^2 + a\left(\sum_{\left(x_j, x_p\right)\in E}\sum_{i=1}^{c}\sum_{k=1, k\neq i}^{c} u_{ij}u_{kp} + \sum_{\left(x_j, x_p\right)\in \Delta}\sum_{i=1}^{c} u_{ij}u_{kp}\right) \tag{5}$$

where $\mathbf{x}_p$ is a point with which $\mathbf{x}_j$ must be linked together, while $\mathbf{x}_q$ is a point with which $\mathbf{x}_j$ cannot be linked. The membership matrix obeys the constraints described in (2).

The first term of (5) is the sum of squared distances to the prototypes weighted by constrained memberships and follows the objective function that is minimized by the FCM algorithm. The second term of (5) is composed of two parts. The first deals with the cost of violating the given pair-wise must-link constraints, while the second with the cost of violating the known pair-wise cannot-link constraints. Factor α, with which the second term is weighted by, serves the relative importance of the supervision.

We solve the problem of minimizing J in (5) subject to (2) by applying the method of Lagrange multiplier, thus we have:

$$u_{ij} = u_{ij}^{(1)} + u_{ij}^{(2)} \tag{6}$$

where

$$u_{ij}^{(1)} = \frac{\left(1 \Big/ \left\|\mathbf{x}_j - \mathbf{v}_i\right\|^2\right)^{1/m-1}}{\sum_{k=1}^{C}\left(1 \Big/ \left\|\mathbf{x}_j - \mathbf{v}_k\right\|^2\right)^{1/m-1}} \tag{7}$$

and

$$u_{ij}^{(2)} = \frac{\lambda}{2\left\|\mathbf{x}_j - \mathbf{v}_i\right\|^2}\left(\Phi_j - \Phi_i\right) \tag{8}$$

Algorithm Ss-Fkcn (Data matrix, Must Link Vector, Must Not Link Vector)

Step 1: Randomly initialize the weights $\mathbf{v}_i$

Step 2: For t=1,2,…,$t_{max.}$

a. Update all learning rates according to Eqs. (3) and (6)

b. Update all c weight vectors following Eq. (4)

c. Compute $E_{t+1} = \|\mathbf{v}(t+1) - \mathbf{v}(t)\|^2$

d. If $E_{t+1} \leq \varepsilon$ then stop; else goto Step 2

e. End for

End Ss-FKCN

Fig. 1. Pseudo code listing for the SS-FKCN algorithm

as well as

$$\Phi_i = \sum_{(\mathbf{x}_j, \mathbf{x}_p) \in \mathrm{E}} \sum_{k=1, k \neq i}^{C} u_{kp} + \sum_{(\mathbf{x}_j, \mathbf{x}_q) \in \Delta} u_{iq} \tag{9}$$

and

$$\Phi_j = \frac{\sum_{t=1}^{C} \frac{\left(\sum_{(\mathbf{x}_j, \mathbf{x}_p) \in \mathrm{E}} \sum_{k=1, k \neq t}^{C} u_{kp} + \sum_{(\mathbf{x}_j, \mathbf{x}_q) \in \Delta} u_{tq} \right)}{\left\| \mathbf{x}_j - \mathbf{v}_t \right\|^2}}{\sum_{t=1}^{C} \left(1 \Big/ \left\| \mathbf{x}_j - \mathbf{v}_t \right\|^2 \right)} \tag{10}$$

The first term of (6) is the same as the one used by the FCM algorithm for the determination of the membership matrix. As it can be depicted by equation (7) it considers distances among weight vectors and input samples. The $\mathbf{u}_{ij}$ term as we can see from equations (9) and (10), increase or decrease the values of the membership matrix as created by (7), using the presented supervision data given in the form of pair-wise constraints. Specifically Φ_i represents the cost for violating a constraint, stating that sample *j* should belong to cluster *i*, and Φ_j represents the weighted average violation cost for sample *j*. The pseudo-code listing of the SS-FKCN algorithm is depicted in Fig. 1.

2.3 Determining the Number of Clusters

One of the main disadvantages of various clustering algorithms is that most of them need an initial guess for the number of clusters. Many times in real problems this is

not known in advance, and further to that, there are problems like the one under study in the current paper, where we need the algorithm to find interesting group of genes guiding this way their further biological study.

The optimal number of partitions is evaluated using the Xie-Beni (XB) validity index [10]. The choice of the specific validity index is based on the fact that it has shown a more stable behavior with respect to other indexes in finding the near-best fuzzy partition. XB is defined as:

$$XB(f) = \frac{\sum\sum u_{ij}^{m} \left\| \mathbf{x}_j - \mathbf{v}_i \right\|^2}{n \min_{ij} \left\| \mathbf{v}_j - \mathbf{v}_i \right\|^2} \tag{11}$$

The procedure followed, for the determination of the cluster number can be described like:

1. Initialize and run SS-FKCN with two weight vectors $n_w = 2$
2. Evaluate XB index, as XB_1 according to (11)
3. Execute SS-FKCN using $n_w = n_w + 1$
4. Evaluate XB index, as XB_2 according to (11)
5. If $XB_2 > XB_1$ then the new value of n_w corresponds to the number of weight vectors for the optimal partition, else goto step 3.

The better separated the clusters are the larger the denominator of (11) and the smaller becomes the value of XB. Thus a valid optimal partition with well distinguishable clusters is indicated by the smallest XB value.

3 Results

In this part of the paper we describe the data sets we have used and the experiments conducted for the validation of the proposed algorithm. The results of SS-FKCN are compared with the ones acquired by two other fuzzy approaches FCM, FKCN, as well as a well known crisp method, the K-means algorithm.

3.1 Artificial Data

In order to test our method under more controlled conditions we resorted to artificial data. Following a standard methodology (e.g. [11]) for the creation of artificial time series data we have created nine groups each one consisting of five time series with the same parameters of linear transformation between time points. Thus

$$x_j(t+1) = \alpha_{it} x_{jt} + \beta_{it} \tag{12}$$

where i represents the number of groups $1 \le i \le 9$, j the number of time series in each group $1 \le j \le 5$, and t the number of time points $1 \le t \le 20$, the parameters α, β for each group where chosen randomly from normal distributions.

3.2 Yeast Cell Cycle – Y5

In this data set first published by [13] we have the expression level of more than 6000 genes measured in 17 time points during two cell cycles of Yeast. We have used a subset five phase criterion abbreviated Y5, of 384 genes visually identified as five distinct time series, each one representing a distinct phase of the cell cycle (Early G1, Late G1, S, G2, and M). All of the Y5 set member genes are annotated. The expression levels of each gene were normalized, which can enhance the performance of clustering under noisy environments like microarrays experiments.

3.3 Simulation Results

In the following we check the validity of the algorithms under test, using the average errors of misclassification. Two other criteria that will be used and which have been adapted by other approaches [12] for evaluating the clustering of time-series data are specificity and sensitivity:

$$Sensitivity = \frac{|TP|}{|TP \cup FN|} \tag{13}$$

$$Specificity = \frac{|TP|}{|TP \cup FP|} \tag{14}$$

where TP denotes the number of pair of objects in the same cluster U and same class V. FP are the pairs with the same cluster but distinct class, TN distinct both class and cluster, and FN pairs distinct cluster but same class. Clustering is repeated 50 times for the data sets under consideration, by all the algorithms checked and the mean values of the validity indexes are calculated.

On the first dataset all algorithms had very good outcomes, given the number of clusters. As we can observe in Fig. 1, following the methodology we have described SS-FKCN was successful in finding the correct number of clusters for the simulation data set. In a second test we performed, the original simulation data was resampled by selecting 10 time points randomly out of the 20 original time points. In this case the best results were acquired by SS-FKCN as it can be shown in Table 2. In both case described above only two pairs were given to the algorithm (i.e. 4.4% of the total knowledge base) in order to have the semi-supervised scheme.

For the second data set (yeast) a total of 25 percent prior knowledge was given on the SS-FKCN, in the form of must and cannot-link pairs. Specifically, we have randomly selected 96 genes and based on their functional labels we have determined the sets of must-link constraints (for genes that are supposed to be part of the same cluster) and cannot-link constraints (for genes that should belong to different clusters). The algorithm as can be seen in Table 1, had a very good apodosis on all validation indexes. Except SS-FKCN, all other methods were unable to separate some genes form groups Late G1 and S or M and Early G1. These phases are separated by a phase shift and their genes have very similar time courses. We should point out that the specific dataset does not represent the ideal data for benchmarking since it almost

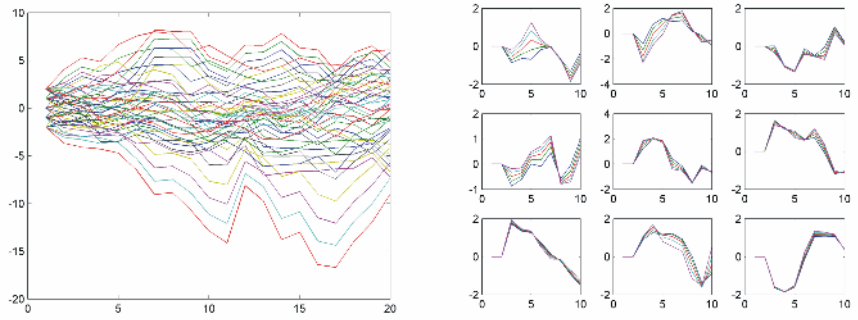


Fig. 1. On the left of the image we depict the whole data set of the artificial data, while on the right part we can see the nine different classes forming it, as they were depicted by SS-FKCN. On all the graphs appearing in this figure, the horizontal axis represents time while the vertical represents gene expression.

Table 1. Misclassification Error, sensitivity and specificity values for the two data sets used, compared for several clustering methods

	Method	Errors	Sensitivity	Specificity
Artificial Data	FCM	6.48	0.93	0.79
	FKCN	5.74	0.94	0.81
	K-means	20.56	0.88	0.68
	SS-FKCN	1.7	0.97	0.89
Yeast	FCM	43.1	0.46	0.53
	FKCN	41.4	0.75	0.36
	K-means	34.8	0.79	0.38
	SS-FKCN	21.7	0.87	0.79

exclusively contains cell cycle genes. Nevertheless the selection of the specific dataset was aiming in pointing out the efficiency of the proposed algorithm, which indeed managed to produce adequate results.

4 Conclusions and Future Work

The paper presents a novel algorithm for semi supervised clustering of gene expression time series data that achieves the incorporation of supervised information by means of pairwise constraints, while benefiting from the advantages of fuzzy, non-crisp, clustering. The experimental results as they were presented for both artificial as well as real data proved that SS-FKCN performs considerably better than other well established clustering techniques, both crisp like Kmeans as well as fuzzy like FCM, given only a small fraction of background knowledge. As a future expansion of the algorithm we are planning to integrate a metaheuristic method like variable neighborhood search in order to avoid local minima entrapment to which the algorithm is sensitive, due to the objective function of the FCM. Another extension which we are planning to work on is to incorporate a different metric more suitable

for time-series classification problem like the one presented in [11] or [14] instead of the Euclidean distance which is used currently by the algorithm.

Acknowledgements

The work conducted in our laboratory was supported by a grant from the General Secretariat of Research and Technology, Ministry of Development of Greece ((013/PENED03) to A.B.

References

1. Eisen M., Spellman P., Brown P., Botstein D. (1998) Cluster analysis and display of genome-wide expression patterns. Proc. Nat. Acad. Sci. **95**, 14863–14868.
2. Liew A.W-C., Yan H., Yang M. (2005) Pattern recognition techniques for the emerging field of bioinformatics. Pattern Recognition **38**, 2055-2073.
3. Guthke R., Moller U., Hoffmann M. et al. (2005) Dynamic network reconstruction from gene expression data applied to immune response during bacteria infection. Bioinformatics **21**, 1626-1634.
4. Tsao E., Bezdek J., Pal N. (1994) Fuzzy Kohonen clustering networks. Pattern Recognition **27**, 757-764.
5. Asyali, M., H., Alci, M. (2005) Reliability analysis of microarray data using fuzzy c-means and normal mixture modeling based classification methods. Bioinformatics, **21**, 644-649
6. Belacel, N., et al (2004) Fuzzy J-Means and VNS methods for clustering genes from microarray data. Bioinformatics, 20**,** 1690-1701.
7. Dembele, D., Kastner, P. (2003) Fuzzy C-means method for clustering microarray data. Bioinformatics, **19**, 973-980.
8. Bezdek, J. C. (1981) Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press.
9. Kohonen, T. (1997) Self-Organized Maps, Springer-Verlag, Second Edition.
10. Pal, N. R., Bezdek, J. C.: (1995) On Cluster Validity for the Fuzzy C-means model. IEEE Transactions on Fuzzy Systems, **3**, pp. 370-379.
11. Möller-Levet, C. S., Klawonn, F., Cho, K.-H., Yin, H., and Wolkenhauer, O, (2005) Clustering of unevenly sampled Gene Expression Time-Series Data. Fuzzy Sets and Systems, **152 pp.** 49-66.
12. Schliep, A., Costa, I., G., Schonhuth, A. (2005) Analyzing Gene Expression Time-Courses. IEEE/ACM Transactions on Computational Biology and Bioinformatics, **2**, 179-193.
13. Cho, R.J., Campbell, M.J., Winzeler, EA., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, TG., Gabrielian, AE, Landsman, D., Lockhart, DJ, Davis, RW (1998) A genome-wide transcriptional analysis of the mitotic cell cycle.Molecular Cell, **2**, pp. 65-73
14. Filkov, V., Skiena, S., Zhi, I. (2002) Analysis Techniques for Microarray Time-Series Data. Journal of Computational Biology, **9**, pp. 317-330.

Evolutionary Optimization of Sequence Kernels for Detection of Bacterial Gene Starts

Britta Mersch[1], Tobias Glasmachers[2], Peter Meinicke[3], and Christian Igel[2]

[1] German Cancer Research Center, 69120 Heidelberg, Germany
b.mersch@dkfz.de
[2] Institut für Neuroinformatik, Ruhr-Universität Bochum, 44780 Bochum, Germany
{Tobias.Glasmachers, Christian.Igel}@neuroinformatik.rub.de
[3] Institut für Mikrobiologie und Genetik, Abteilung für Bioinformatik,
Georg-August-Universität Göttingen, 37077 Göttingen, Germany
pmeinic@gwdg.de

Abstract. Oligo kernels for biological sequence classification have a high discriminative power. A new parameterization for the K-mer oligo kernel is presented, where all oligomers of length K are weighted individually. The task specific choice of these parameters increases the classification performance and reveals information about discriminative features. For adapting the multiple kernel parameters based on cross-validation the covariance matrix adaptation evolution strategy is proposed. It is applied to optimize the trimer oligo kernel for the detection of prokaryotic translation initiation sites. The resulting kernel leads to higher classification rates, and the adapted parameters reveal the importance for classification of particular triplets, for example of those occurring in the Shine-Dalgarno sequence.

1 Introduction

Kernel-based learning algorithms have been successfully applied to a variety of sequence classification tasks within the field of bioinformatics [1]. Recently, *oligo kernels* were proposed [2] for the analysis of biological sequences. Here the term oligo (-mer) refers to short, single stranded DNA/RNA fragments. Oligo kernels compare sequences by looking for matching fragments. They allow for gradually controlling the level of position-dependency of the representation, that is, how important the exact position of an oligomer is. In addition, decision functions based on oligo kernels are easy to interpret and to visualize and can therefore be used to infer characteristic sequence features.

In the standard oligo kernel, all oligomers are weighted equally. Thus, all oligomers are considered to have the same importance for classification. In general this assumption is not reasonable. In this study, we therefore propose the K-weighted oligo kernel considering all oligomers of length K (K-mers), in which the relative importance of all K-mers can be controlled individually. A task specific choice of the weighting parameters can potentially increase the classification performance. Moreover, appropriate weights for a particular classification task may reveal sequence characteristics with high discriminative power and biological importance.

The question arises how to adjust the weighting parameters for the K-mers for a given task. In practice, appropriate hyperparameter combinations are usually determined by

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 827–836, 2006.

grid search. This means that the hyperparameters are varied with a fixed step size through a wide range of values and the performance of every combination is assessed using some performance measure. Because of the computational complexity, grid search is only suitable for the adjustment of very few parameters. Hence, it is not applicable for the adjustment of the 4^K weights of the K-weighted oligo kernel. Perhaps the most elaborated systematic technique for choosing multiple hyperparameters are gradient descent methods [3, 4, 5]. If applicable, these methods are highly efficient. However, they have significant drawbacks. In particular, the score function for assessing the performance of the hyperparameters (or at least an accurate approximation of this function) has to be differentiable with respect to all hyperparameters. This excludes reasonable measures such as the (exact) cross-validation error. Further, the considered space of kernels has to have an appropriate differentiable structure.

We propose a method for hyperparameter selection that does not suffer from the limitations described above, namely using the covariance matrix adaptation evolution strategy (CMA-ES, [6]) to search for appropriate hyperparameter vectors [7, 8].

As an application of our approach to kernel optimization we consider the prediction of bacterial gene starts in genomic sequences. Although exact localization of gene starts is crucial for correct annotation of bacterial genomes, it is difficult to achieve with conventional gene finders, which are usually restricted to the identification of long coding regions. The prediction of gene starts therefore provides a biologically relevant signal detection task, well-suited for the evaluation of our kernel optimization scheme.

We therefore apply the CMA-ES to the tuning of weighted oligo kernels for detecting prokaryotic translation initiation sites, that is, for classifying putative gene starts in bacterial RNA. The performance measure for the hyperparameter optimization is based on the mean classification rate of five-fold cross-validation.

In the following we introduce the oligo kernel and our new parameterization. Section 3 deals with the adaptation of kernel parameters using evolutionary optimization methods. Section 4 presents the experiments demonstrating the performance of the kernel and the optimization of the hyperparameters.

2 Oligo Kernels

The basic idea of kernel methods for classification is to map the input data, here biological sequences, to a feature space endowed with a dot product. Then the data is processed using a learning algorithm in which all operations in feature space can be expressed by dot products. The trick is to compute these inner products efficiently in input space using a kernel function (e.g., see [9]). Here the feature space can be described in terms of *oligo functions* [2]. These functions encode occurrences of oligomers in sequences with an adjustable degree of positional uncertainty. This is in contrast to existing methods, which provide either position-dependent [10] or completely position-independent representations [11]. For an alphabet $\mathscr{A}$ and a sequence $\mathbf{s}$, which contains K-mer $\omega \in \mathscr{A}^K$ at positions $S^{\mathbf{s}}_{\omega} = \{p_1, p_2, \dots\}$, the oligo function is given by

$$\mu^{\mathbf{s}}_{\omega}(t) = \sum_{p \in S^{\mathbf{s}}_{\omega}} \exp\left(-\frac{1}{2\sigma_K^2}(t-p)^2\right)$$

for $t \in \mathbb{R}$. The smoothing parameter σ_K adjusts the width of the Gaussians centered on the observed oligomer positions and determines the degree of position-dependency of the function-based feature space representation. While small values for σ_K imply peaky functions, large values imply flatter functions. For a sequence $\mathbf{s}$ the occurrences of all K-mers contained in $\mathscr{A}^K = \{\omega_1, \omega_2, \ldots, \omega_m\}$ can be represented by a vector of m oligo functions. This yields the final feature space representation $\Phi^K(\mathbf{s}) = [\mu^{\mathbf{s}}_{\omega_1}, \mu^{\mathbf{s}}_{\omega_2}, \ldots, \mu^{\mathbf{s}}_{\omega_m}]'$ of that sequence. The feature space objects are vector-valued functions. This can be stressed using the notation

$$\phi^K_{\mathbf{s}}(t) = [\mu^{\mathbf{s}}_{\omega_1}(t), \mu^{\mathbf{s}}_{\omega_2}(t), \ldots, \mu^{\mathbf{s}}_{\omega_m}(t)]' \ .$$

This representation is well-suited for the interpretation of discriminant functions and visualization [2]. To make it practical for learning, we construct a kernel function to compute the dot product in the feature space efficiently. The inner product of two sequence representations ϕ^K_i and ϕ^K_j, corresponding to the oligo kernel $k_K(\mathbf{s}_i, \mathbf{s}_j)$, can be defined as

$$\left\langle \phi^K_i, \phi^K_j \right\rangle \equiv \int \phi^K_i(t) \cdot \phi^K_j(t) \mathrm{d}t \propto \sum_{\omega \in \mathscr{A}^K} \sum_{p \in S^i_\omega} \sum_{q \in S^j_\omega} \exp\left(-\frac{1}{4\sigma_K^2}(p-q)^2 \right) \equiv k_K(\mathbf{s}_i, \mathbf{s}_j)$$

using $\phi_i \equiv \phi_{\mathbf{s}_i}$. The feature space representations of two sequences may have different norms. In order to improve comparability between sequences of different lengths, we compute the normalized oligo kernel

$$\tilde{k}_K(\mathbf{s}_i, \mathbf{s}_j) = \frac{k_K(\mathbf{s}_i, \mathbf{s}_j)}{\sqrt{k_K(\mathbf{s}_i, \mathbf{s}_i) k_K(\mathbf{s}_j, \mathbf{s}_j)}} \ . \tag{1}$$

From the above definition of the oligo kernel, the effect of the smoothing parameter σ_K becomes obvious. For the limiting case $\sigma_K \to 0$ with no positional uncertainty, only oligomers which occur at the same positions in both sequences contribute to the sum. In general it is not appropriate to represent oligomer occurrences without positional uncertainty. This would imply zero similarity between two sequences if no K-mer appears at *exactly* the same position in both sequences. For $\sigma_K \to \infty$ position-dependency of the kernel completely vanishes. In this case, all terms of oligomers occurring in both sequences contribute equally to the sum, regardless of their distance and the oligo kernel becomes identical to the spectrum kernel [11].

2.1 Weighted Oligo Kernel

So far, the different K-mers are weighted equally in the K-mer oligo kernel. However, some K-mers may be more discriminative than others. Therefore, we introduce new parameters w_i, $i = 1, \ldots, 4^K$, for their weighting and define the K-weighted oligo kernel $\tilde{k}_{K\text{-weighted}}$ in analogy to equation (1) with

$$k_{K\text{-weighted}}(\mathbf{s}_i, \mathbf{s}_j) = \sum_{\omega \in \mathcal{A}^K} |w_i| \sum_{p \in S^{\mathbf{s}_i}_\omega} \sum_{q \in S^{\mathbf{s}_j}_\omega} \exp\left(-\frac{1}{4\sigma_K^2}(p-q)^2 \right) \ .$$

The parameterization ensures a valid oligo kernel for $w_1, \ldots, w_{4^K}, \sigma \in \mathbb{R}$. This makes unconstrained optimization methods directly applicable to the $1 + 4^K$ kernel parameters.

3 Evolutionary Model Selection

Evolutionary algorithms are iterative, direct, randomized optimization methods inspired by principles of neo-Darwinian evolution theory. They have proven to be suitable for hyperparameter and feature selection for kernel-based learning algorithms [7,8,12,13,14,15,16,17,18,19]. Evolution strategies (ES, [20]) are one of the main branches of evolutionary algorithms. Here the highly efficient covariance matrix ES (CMA-ES, [6,21]) for real-valued optimization is applied, which learns and employs a variable metric by means of a covariance matrix for the search distribution. The CMA-ES has successfully been applied to tune Gaussian kernels for SVMs considering a cross-validation error as optimization criterion [7, 8]. The visualization of the objective function in [7] depicts an error surface that shows a global trend superimposed by local minima, and ES are usually a good choice for such kind of problems.

In the CMA-ES, a set of μ individuals forming the parent population is maintained. Each individual has a genotype that encodes a candidate solution for the optimization problem at hand, here a real-valued vector containing the hyperparameter combination of the kernel parameters to be optimized. The fitness of an individual is equal to the objective function value—here the five-fold cross-validation error—at the point in the search space it represents. In each iteration of the algorithm, $\lambda > \mu$ new individuals, the offspring, are generated by partially stochastic variations of parent individuals. The fitness of the offspring is computed and the μ best of the offspring form the next parent population. This loop of variation and selection is repeated until a termination criterion is met. The object variables are altered by global intermediate recombination and Gaussian mutation. That is, the genotypes $\mathbf{g}_k^{(t)}$ of the offspring $k = 1,\ldots,\mu$ created in iteration t are given by $\mathbf{g}_k^{(t)} = \langle\tilde{\mathbf{g}}\rangle^{(t)} + \xi_k^{(t)}$, where $\langle\tilde{\mathbf{g}}\rangle^{(t)}$ is the center of mass of the parent population in iteration t, and the $\xi_k^{(t)} \sim \mathcal{N}(0, \mathbf{C}^{(t)})$ are independent realizations of an m-dimensional normally distributed random vector with zero mean and covariance matrix $\mathbf{C}^{(t)}$. The matrix $\mathbf{C}^{(t)}$ is updated online using the covariance matrix adaptation method (CMA). Roughly speaking, the key idea of the CMA is to alter the mutation distribution in a deterministic way such that the probability to reproduce steps in the search space that led to the actual population—i.e., produced offspring that were selected—is increased. The search path of the population over the past generations is taken into account, where the influence of previous steps decays exponentially. The CMA does not only adjust the mutation strengths in m directions, but also detects correlations between object variables. The CMA-ES is invariant against order-preserving transformations of the fitness function and in particular against rotation and translation of the search space—apart from the initialization. If either the strategy parameters are initialized accordingly or the time needed to adapt the strategy parameters is neglected, any affine transformation of the search space does not affect the performance of the CMA-ES. For details of the CMA-ES algorithm, we refer to the literature [6, 21].

4 Detection of Prokaryotic Translation Initiation Sites

We apply 1-norm soft margin SVMs with 3-mer weighted oligo kernels to the detection of prokaryotic translation initiation sites [22]. We first introduce the problem and then

the locality improved kernel, which we consider for comparison. Then the experimental setup is described. Finally the results are presented.

4.1 Problem Description

To extract protein-encoding sequences from nucleotide sequences is an important task in bioinformatics. For this purpose it is necessary to detect locations at which coding regions start. These locations are called translation initiation sites (TIS). A TIS contains the start codon ATG or rarely GTG or TTG (there is one known case where also ATT serves as a start codon). The start codon marks the position at which the translation starts. The codon ATG codes for the amino acid methionine, and not every ATG triplet is a start codon. Therefore it must be decided whether a particular ATG corresponds to a start codon or not. This classification problem can be solved automatically using machine learning techniques, in which the neighborhood of nucleotides observed around potential TISs is used as input pattern to a classifier.

In contrast to prediction of eukaryotic TIS (e.g., see [23]) there is no biological justification for using a general learning machine across different species for prediction of prokaryotic TIS. For this reason, learning of prokaryotic TISs is always restricted to a limited amount of species-specific examples and model selection methods have to cope with small data sets.

As in previous studies, we tested our approach on *E. coli* genes from the EcoGene database [24]. Only those entries with biochemically verified N-terminus were considered and the neighboring nucleotides were looked up in the GenBank file U00096.gbk [25]. From the 730 positive examples we created associated negative examples. For the negative examples we extracted sequences centered around a codon from the set $\{\mathsf{ATG}, \mathsf{GTG}, \mathsf{TTG}\}$. Such a sequence is used as a negative example if the codon is in-frame with one of the correct start sites used as a positive case, its distance from a real TIS is less than 80 nucleotides, and no in-frame stop codon occurs in between. This procedure generates a difficult benchmark data set, because the potential TISs in the neighborhood of the real start codon are the most difficult candidates in TIS discrimination. We created 1243 negative examples. The length of each sequence is 50 nucleotides, with 32 located upstream and 15 downstream with respect to the potential start codon.

To minimize random effects, we generated 40 different partitionings of the data into training and test sets. Each training set contained 400 sequences plus the associated negatives, the corresponding test set 330 sequences plus the associated negatives.

Measuring the performance of a TIS classifier by the standard classification rate on test sets leads to over-optimistic results. In a process of annotation, one normally obtains a window with several possible TISs. The goal is to detect the position of a real TIS—if there is one—within this window. If there are several positions marked as TISs, one has to select one of them. In practice, the position with the highest score (i.e., decision function value) is chosen. Thus, although a real TISs was classified as a TIS, the classification can be overruled by a wrong classification in the neighborhood. Therefore, when the SVM categorizes a location with corresponding sequence $\mathbf{s}$ as being a TIS, we consider a frame of 160 nucleotides centered at that position. The score of every potential TIS within this frame is computed. Only if $\mathbf{s}$ corresponds to a real TIS and

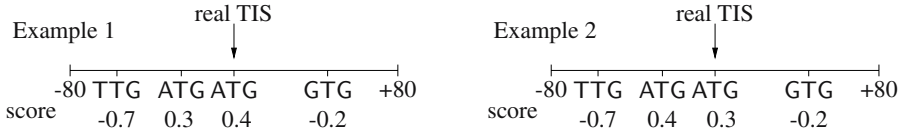


Fig. 1. Performance assessment: Example 1 shows a correct positive classification of a TIS. In example 2, the classification is not correct: The real TIS is classified as a TIS, but its score is not the largest in the neighborhood.

the score for $\mathbf{s}$ is the largest of all potential TIS locations, the pattern $\mathbf{s}$ is considered to be classified correctly, see Figure 1.

4.2 Locality Improved Kernel

For comparison, we consider the locality improved kernel [1,23]. It counts matching nucleotides and considers local correlations within local windows of length $2l+1$. Given two sequences $\mathbf{s}_i$, $\mathbf{s}_j$ of length L the locality improved kernel is given by

$$k_{\text{locality}}(\mathbf{s}_i,\mathbf{s}_j) = \sum_{p=l+1}^{L-l} \left(\sum_{t=-l}^{+l} v_{t+l} \cdot \text{match}_{p+t}(\mathbf{s}_i,\mathbf{s}_j) \right)^d$$

with $\text{match}_t(\mathbf{s}_i,\mathbf{s}_j)$ equal to one if $\mathbf{s}_i$ and $\mathbf{s}_j$ have the same nucleotide at position t and zero otherwise. The weights v_t allow to emphasize regions of the window which are of special importance. They were fixed to $v_t = 0.5 - 0.4|l-t|/l$ [1]. The hyperparameter d determines the order to which local correlations are considered.

4.3 Experiments

In our experiments, we considered trimer oligo kernels with hyperparameter σ, locality improved kernels with hyperparameters l and d, and weighted trimer oligo kernels with adjustable σ and 64 weights. For each of the 40 partitionings into training and test data and each sequence kernel independent optimizations of the kernel parameters were conducted. In the end, we evaluate the median of the 40 trials.

For the SVM using the oligo kernel without individually weighting of the K-mers we adjusted the smoothing parameter σ by one-dimensional grid-search. After narrowing the possible values down, the grid search varied $\sigma \in \{0.1 + 0.2 \cdot k \mid 0 \leq k < 10\}$. The parameters l and d of the locality improved kernel were also optimized using two-dimensional grid-search. After determining an interval of parameters leading to well generalizing classifiers, the grid-search varied $l,d \in \{2,3,4\}$ [23]. For both kernels, independent grid-searches were performed for each of the 40 partitionings.

The $1+4^3 = 65$ parameters of the weighted trimer oligo kernels were optimized using the CMA-ES with randomly chosen starting points in the interval $[0,1]$. For each of the 40 partitionings an independent optimization trial was started. The offspring population size was $\lambda = 16$ (e.g., a default choice for this dimensionality, see [21]) and each trial lasted 100 generations.

The optimization criterion in the grid-searches and the evolutionary optimization was the five-fold cross-validation error based on the error measure described above. The training data set is partitioned into five disjoint subsets. For each of the subsets, the classifier is trained using the union of the four other sets and a test error is computed on the left-out subset. The final cross-validation error is the average of the five test errors.

4.4 Results

We first interpret the outcome of the optimization of the parameters of the weighted oligo kernel. Then we compare the classification performance of the weighted oligo kernel, the trimer oligo kernel with equal weights, and the locality improved kernel.

The results of the optimization of the smoothing parameter σ are shown in Table 1. The optimized values are rather small, that is, the position of the triplets is very important. However, the smoothing parameter for the oligo and the weighted oligo kernel do not differ much.

To analyze the relevance of particular oligomers, the 64 triplets were sorted according to the median of the corresponding evolved weighting parameters. The weight values indeed vary, and a group of a few oligomers with comparatively high weight values can be identified. These triplets on the first 10 ranks are given in Table 2. Additionally to the start codon ATG the triplets GAG, AGG, and GGA were assigned the largest weight values. These triplets are all contained in the sequence TAAGGAGGT, which is known to be of importance for translation initiation sites because it is the sequence that will bind to the 16S rRNA 3' terminal sequence of the ribosome. This sequence is called Shine-Dalgarno Sequence [26, 27]. Obviously the kernel uses the presence of triplets occurring in the Shine-Dalgarno sequence for discrimination.

The medians of the weights for the potential start codons were 5.6 for ATG, 3.58 for TTG, and 2.45 for GTG. That is, the presence of ATG appears to be a relevant feature, whereas GTG and TTG are not as important as ATG. In all positive as well as negative sequence patterns there is a potential start codon at the positions 33–35. Still, the frequency of ATG at this position is considerably higher in positive than in negative

Table 1. Optimized smoothing parameter for the oligo and the weighted oligo kernel

	oligo kernel (grid search)	weighted oligo kernel (CMA-ES)
Mean	1.86	1.71
Median	1.9	1.83
0.25 quantile	1.5	1.34
0.75 quantile	2.3	2.12

Table 2. The 3-mers of major importance for classification

3-mer	GAG	ATG	AGG	GGA	GGC	GCT	CAA	TTG	TCC	GGG
weight	6.23	5.6	5.36	5.29	5	4.81	4.05	3.58	3.45	3.41

Table 3. Classification performance in percent for 40 trials with different partitionings of the data

	oligo kernel (grid search)	locality improved kernel (grid search)	weighted oligo kernel (CMA-ES)
Mean	84.86	85.60	86.02
Median	85.01	86.01	86.41
0.25 quantile	83.90	84.63	84.79
0.75 quantile	86.42	86.78	87.08

examples. The initiation codon of more than 90 % of prokaryotic genes is ATG [22]. The rule of thumb "a pattern is positive if the start codon is ATG and negative otherwise", which would lead to a classification accuracy of about 72% when applied to our data, can be implemented with the evolved kernel weights. However, more sophisticated features based on the triplets with large weights in Table 2 can overrule the presence or absence of ATG.

The classification results are given in Table 3. The median of the classification performance of the 3-mer oligo kernel with equal weighting is 85.01%. Introducing the weights for the individual 3-mers in the oligo kernel and optimizing them using CMA-ES leads to an increase of the classification performance to 86.41%. The results achieved by the weighted oligo kernel are significantly better than those of the oligo kernel with equal weights and the smoothing parameter as only adjustable variable (Wilcoxon rank-sum test, $p < 0.01$).

The median of the locality improved kernel parameters adjusted by grid search was two for both l and d. That is, the nucleotides were only compared within a small window. This is in accordance with the results for σ in the oligo kernels. The median of the classification performance reached by the locality improved kernel is 86.01%, that is, between the 3-mer oligo kernel with equal weights and the evolutionary optimized 3-weighted oligo kernel. However, the differences are not statistically significant (Wilcoxon rank-sum test, $p > 0.05$).

5 Conclusion and Outlook

A task specific choice of the kernel can significantly improve kernel-based machine learning. Often a parameterized family of kernel functions is considered so that the kernel adaptation reduces to real-valued optimization. Still, the adaptation of complex kernels requires powerful optimization methods that can adapt multiple parameters efficiently. When the considered space of kernel functions lacks a differentiable structure or the model selection criterion is non-differentiable, a direct search method is needed. The covariance matrix adaptation evolution strategy (CMA-ES) is such a powerful, direct algorithm for real-valued hyperparameter selection.

In biological sequence analysis, the CMA-ES allows for a more task specific adaptation of sequence kernels. Because multiple parameters can be adapted, it is possible to adjust new weighting variables in the oligo kernel to control the influence of every oligomer individually. Further, the cross-validation error can directly be optimized (i.e., without smoothening).

We demonstrated the discriminative power of the oligo kernel and the benefits of the evolutionary model selection approach by applying them to prediction of prokaryotic translation initiation sites (TISs). The adapted weighted oligo kernel leads to improved results compared to kernel functions with less adaptable parameters, which were optimized by grid-search. Furthermore, it is possible to reveal biologically relevant information from analyzing the evolved weighting parameters. For the prediction of TISs, for example, triplets referring to the Shine-Dalgarno sequence are used for discrimination.

Acknowledgments

The authors thank Nico Pfeifer for his work on the original implementation of the oligo kernel and the TIS classification.

References

1. Schölkopf, B., Tsuda, K., Vert, J.P., eds.: Kernel Methods in Computational Biology. Computational Molecular Biology. MIT Press (2004)
2. Meinicke, P., Tech, M., Morgenstern, B., Merkl, R.: Oligo kernels for datamining on biological sequences: A case study on prokaryotic translation initiation sites. BMC Bioinformatics **5** (2004)
3. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. Machine Learning **46** (2002) 131–159
4. Glasmachers, T., Igel, C.: Gradient-based adaptation of general Gaussian kernels. Neural Computation **17** (2005) 2099–2105
5. Keerthi, S.S.: Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms. IEEE Transactions on Neural Networks **13** (2002) 1225–1229
6. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation **9** (2001) 159–195
7. Friedrichs, F., Igel, C.: Evolutionary tuning of multiple SVM parameters. Neurocomputing **64** (2005) 107–117
8. Igel, C., Wiegand, S., Friedrichs, F.: Evolutionary optimization of neural systems: The use of self-adaptation. In: Trends and Applications in Constructive Approximation. Number 151 in International Series of Numerical Mathematics. Birkhäuser Verlag (2005) 103–123
9. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press (2002)
10. Degroeve, S., Beats, B.D., de Peer, Y.V., Rouzé, P.: Feature subset selection for splice site prediction. Bioinformatics **18** (2002) 75–83
11. Leslie, C., Eskin, E., Noble, W.S.: The spectrum kernel: A string kernel for SVM protein classification. In Altman, R.B., et al., eds.: Proceedings of the Pacific Symposium on Biocomputing, World Scientific (2002) 564–575
12. Eads, D.R., et al.: Genetic algorithms and support vector machines for time series classification. In Bosacchi, B., Fogel, D.B., Bezdek, J.C., eds.: Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation V. Volume 4787 of Proceedings of the SPIE. (2002) 74–85
13. Fröhlich, H., Chapelle, O., Schölkopf, B.: Feature selection for support vector machines using genetic algorithms. International Journal on Artificial Intelligence Tools **13** (2004) 791–800

14. Igel, C.: Multi-objective model selection for support vector machines. In Coello, C.A.C., Zitzler, E., Aguirre, A.H., eds.: Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005). Volume 3410 of LNAI., Springer-Verlag (2005) 534–546
15. Jong, K., Marchiori, E., van der Vaart, A.: Analysis of proteomic pattern data for cancer detection. In Raidl, G.R., et al., eds.: Applications of Evolutionary Computing. Number 3005 in LNCS, Springer-Verlag (2004) 41–51
16. Miller, M.T., Jerebko, A.K., Malley, J.D., Summers, R.M.: Feature selection for computer-aided polyp detection using genetic algorithms. In Clough, A.V., Amini, A.A., eds.: Medical Imaging 2003: Physiology and Function: Methods, Systems, and Applications. Volume 5031 of Proceedings of the SPIE. (2003) 102–110
17. Pang, S., Kasabov, N.: Inductive vs. transductive inference, global vs. local models: SVM, TSVM, and SVMT for gene expression classification problems. In: International Joint Conference on Neural Networks (IJCNN). Volume 2., IEEE Press (2004) 1197–1202
18. Runarsson, T.P., Sigurdsson, S.: Asynchronous parallel evolutionary model selection for support vector machines. Neural Information Processing – Letters and Reviews **3** (2004) 59–68
19. Shi, S.Y.M., Suganthan, P.N., Deb, K.: Multi-class protein fold recognition using multi-objective evolutionary algorithms. In: IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, IEEE Press (2004) 61–66
20. Beyer, H.G., Schwefel, H.P.: Evolution strategies: A comprehensive introduction. Natural Computing **1** (2002) 3–52
21. Hansen, N., Kern, S.: Evaluating the CMA evolution strategy on multimodal test functions. In Yao, X., et al., eds.: Parallel Problem Solving from Nature (PPSN VIII). Volume 3242 of LNCS., Springer-Verlag (2004) 282–291
22. Gualerzi, C.O., Pon, C.L.: Initiation of mRNA translation in procaryotes. Biochemistry **29** (1990) 5881–5889
23. Zien, A., et al.: Engineering support vector machine kernels that recognize translation initiation sites. Bioinformatics **16** (2000) 799–807
24. Rudd, K.E.: Ecogene: a genome sequence database for *Escherichia coli* K-12. Nucleic Acids Research **28** (2000) 60–64
25. Blattner, F.R., et al.: The complete genome sequence of Escherichia coli K-12. Science **277** (1997) 1453–1462
26. Kozak, M.: Initiation of translation in prokaryotes and eukaryotes. Gene **234** (1999) 187–208
27. Shine, J., Dalgarno, L.: The 3'-terminal sequence of *Escherichia coli* 16S ribosomal RNA: Complementarity to nonsense triplets and ribosome binding sites. PNAS **71** (1974) 1342–1346

Tree-Dependent Components of Gene Expression Data for Clustering

Jong Kyoung Kim and Seungjin Choi

Department of Computer Science
Pohang University of Science and Technology
San 31 Hyoja-dong, Nam-gu
Pohang 790-784, Korea
{blkimjk, seungjin}@postech.ac.kr

Abstract. Tree-dependent component analysis (TCA) is a generalization of independent component analysis (ICA), the goal of which is to model the multivariate data by a linear transformation of latent variables, while latent variables fit by a tree-structured graphical model. In contrast to ICA, TCA allows dependent structure of latent variables and also consider non-spanning trees (forests). In this paper, we present a TCA-based method of clustering gene expression data. Empirical study with yeast cell cycle-related data, yeast metabolic shift data, and yeast sporulation data, shows that TCA is more suitable for gene clustering, compared to principal component analysis (PCA) as well as ICA.

1 Introduction

Clustering genes from expression data into biologically relevant groups, is a valuable tool for finding characteristic expression patterns of a cell and for inferring functions of unknown genes. Clustering is also widely used in modelling transcriptional regulatory networks, since it reduces the data complexity [1]. On one hand, classical clustering methods such as k-means, hierarchical clustering and self-organizing map (SOM), have widely been used in bioinformatics. On the other hand, linear latent variables models were recently used in the task of gene clustering. This includes principal component analysis (PCA) [2], factor analysis [3], independent component analysis (ICA) [4,5,6], independent subspace analysis (ISA) [7,8], and topographic ICA [9].

The underlying assumption in linear latent variable models, is that gene expression profiles (measured by microarray experiments) are generated by a linear combination of linear modes (corresponding to prototype biological processes) with weights (encoding variables or factors) determined by latent variables. In such a case, latent variables indicates the portion of contributions of each linear mode to a specific gene profile. Clustering gene profiles can be carried out by investigating the significance of latent variables and representative biological functions directly come from linear modes of latent variable models. It was shown that clustering by latent variable models outperforms classical clustering algorithms (e.g., k-means) [5].

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 837–846, 2006.

Tree-dependent component analysis (TCA) is a generalization of ICA, the goal of which is to seek a linear transform with latent variables well-fitting by a tree-structured graphical model, in contrast to ICA which restricts latent variable to be statistically independent [10]. TCA allows the dependent structure of latent variables and also incorporates with non-spanning trees (forests). In this paper, we present a method of gene clustering based on TCA. We compare the performance of TCA to PCA and ICA, for three yeast data sets, evaluating the enrichment of clusters through the statistical significance of *Gene Ontology* (GO) annotations [11].

2 Linear Latent Variable Models

Gene expression patterns measured in microarray experiments, result from unknown generative processes contributed by diverse biological processes such as the binding of transcription factors and environmental change outside a cell [4]. Genome-wide gene expression involves a very complex biological system and the characteristics of biological processes is hidden to us. A promising way to model such a generative process, is to consider a linear latent variable model such as PCA and ICA.

The linear generative model assumes that a gene profile $\boldsymbol{x}_t \in \mathbb{R}^m$ (the elements of $\boldsymbol{x}_t$ represent the expression levels of gene t at m samples or m time points) is assumed to be generated by

$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{s}_t + \boldsymbol{\epsilon}_t, \quad t = 1, \ldots, N, \tag{1}$$

where $\boldsymbol{A} = [\boldsymbol{a}_1 \cdots \boldsymbol{a}_n] \in \mathbb{R}^{m \times n}$ contains linear modes in its columns and $\boldsymbol{s}_t \in \mathbb{R}^n$ is a latent variable vector with each element s_{it} associated with the contribution of the linear mode $\boldsymbol{a}_i$ to the gene profile $\boldsymbol{x}_t$. The noise vector $\boldsymbol{\epsilon}_t \in \mathbb{R}^m$ takes the uncertainty in the model into account and it is assumed to be statistically independent of $\boldsymbol{s}_t$. For the sake of simplicity, we neglect the noise vector $\boldsymbol{\epsilon}_t$. Then the linear generative model (1) can be written in a compact form:

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}, \tag{2}$$

where $\boldsymbol{X} = [X_{it}] \in \mathbb{R}^{m \times N}$ is the data matrix with each element X_{it} associated with the expression level of gene t at sample i (or time i). The latent variable matrix $\boldsymbol{S} \in \mathbb{R}^{n \times N}$ contains $\boldsymbol{s}_t$ for $t = 1, \ldots, N$.

Given a data matrix $\boldsymbol{X}$, latent variables $\boldsymbol{S}$ are determined by $\boldsymbol{S} = \boldsymbol{W}\boldsymbol{X}$, where the linear transformation $\boldsymbol{W}$ is estimated by a certain optimization method. Depending on restrictions or assumptions on $\boldsymbol{A}$ and $\boldsymbol{S}$, various methods including PCA, ICA, and TCA have been developed. A brief overview of those methods is given below.

2.1 PCA

PCA is a widely-used linear dimensionality reduction technique which decomposes high-dimensional data into low-dimensional subspace components. PCA

is illustrated as a linear orthogonal transformation which captures maximal variations in data. Various algorithms for PCA have been developed [12,13,14]. Singular value decomposition (SVD) is an easy way to determine principal components.

The SVD of the data matrix $\boldsymbol{X} \in \mathbb{R}^{m \times N}$ is given by

$$\boldsymbol{X} \approx \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\top}, \tag{3}$$

where $\boldsymbol{U} \in \mathbb{R}^{m \times n}$ $(n \leq m)$ contains n principal left singular vectors (eigenvectors) in its columns, $\boldsymbol{D} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with eigenvalues on diagonal entries, and $\boldsymbol{V} \in \mathbb{R}^{N \times n}$ contains n right singular vectors in its columns.

In the framework of gene expression data analysis, the n column vectors of $\boldsymbol{U}$ correspond to *eigengenes* and the n column vectors of $\boldsymbol{V}$ are associated with *eigenarrays*. Exemplary applications of SVD or PCA to gene expression data, can be found in [15,2].

2.2 ICA

ICA is a statistical method which model the observed data $\{\boldsymbol{x}_t\}$ by a linear model $\{\boldsymbol{A}\boldsymbol{s}_t\}$ with restricting non-Gaussian latent variables $\boldsymbol{s}_t$ to have statistically independent components. In contrast to PCA where the multivariate data is modelled by an orthogonal transformation of independent (or uncorrelated) Gaussian latent variables, ICA seeks a non-orthogonal transformation that makes non-Gaussian components to be as independent as possible. Refer to [16,17,18] for details and recent review of ICA.

The non-Gaussianity constraint for independent components, is very useful in the gene expression data analysis. Hidden biological processes affect only a few relevant genes and a large portion of genes remains unaffected. Gaussian distribution does not model this encoding process correctly. In fact, heavy-tailed distributions are more suitable for encoding variables $\{\boldsymbol{s}_t\}$ in gene expression data [5,4]. The independence assumption on hidden variables $\{\boldsymbol{s}_t\}$ was shown to be an effective hypothesis for separating linearly-mixed biological signals in gene expression data. Despite of this effectiveness of the independence assumption, it is not realistic since biological systems are known to be highly inter-connected networks.

2.3 TCA

For the sake of simplicity, we omit the index t in both $\boldsymbol{x}_t$ and $\boldsymbol{s}_t$, unless it is necessary. As in ICA, we also assume that the data is pre-processed by PCA such that its dimension is reduced down to n. TCA is a generalization of ICA, where instead of seeking a linear transformation $\boldsymbol{W}$ that makes components $\{s_i\}$ independent (s_i is the ith-element of $\boldsymbol{s} = \boldsymbol{W}\boldsymbol{x}$), it searches for a linear transform $\boldsymbol{W}$ such that components (latent variables) $\{s_i\}$ well-fit by a tree-structured graphical model [10]. In TCA, s_i are referred to as *tree-dependent components*. In contrast to ICA, TCA allows the components s_i to be dependent and its dependency is captured by a tree-structured graphical model. Thus, it is

expected that TCA will be more suitable for gene clustering than ICA, since it is more realistic in seeking hidden biological processes. A brief overview of TCA is given below, and see [10] for more details.

Let us denote by $T(\mathcal{V}, \mathcal{E})$ an undirected tree, where $\mathcal{V}$ and $\mathcal{E}$ represent a set of nodes and a set of edges, respectively. The objective function considered in TCA model, involves the *T-mutual information* $I_T(\boldsymbol{s})$:

$$\begin{aligned} \mathcal{J}(\boldsymbol{x}, \boldsymbol{W}, T) &= I_T(\boldsymbol{s}) \\ &= I(s_1, \ldots, s_n) - \sum_{(i,j)\in\mathcal{E}} I(s_i, s_j), \end{aligned} \tag{4}$$

where $I(\cdot)$ is the mutual information. Note that in the case of ICA, only the mutual information $I(s_1, \ldots, s_n)$ serves as the objective function. The objective function (4) results from the minimal KL-divergence between the empirical distribution $p(\boldsymbol{x})$ and the model distribution $q(\boldsymbol{x})$ where the linear model $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$ is considered and $\boldsymbol{s}$ is assumed to factorize in a tree T.

In terms of entropies (denoted by $H(\cdot)$), the objective function (4) can be written as

$$\begin{aligned} \mathcal{J}(\boldsymbol{x}, \boldsymbol{W}, T) = \sum_j H(s_j) - \sum_{(i,j)\in\mathcal{E}} [H(s_i) + H(s_j) - H(s_i, s_j)] \\ - \log|\det \boldsymbol{W}|, \end{aligned} \tag{5}$$

where $H(\boldsymbol{x})$ is omitted since it is constant. The objective function (5) involves the calculation of entropy, which requires the probability distribution of $\boldsymbol{s}$ that is not available in advance. Several empirical contrast functions were considered in [10]. These include: (1) kernel density estimation (KDE); (2) Gram-Charlier expansion; (3) kernel generalized variance; (4) multivariate Gaussian stationary process-based entropy rate. In the case of ICA, Gaussian latent variables are not interesting. In such a case, the transformation $\boldsymbol{W}$ is defined up to an orthogonal matrix. On the other hand, TCA imposes a tree-structured dependency on latent variables, hence, this indeterminacy disappears and the transformation $\boldsymbol{W}$ can be estimated with a fixed tree T.

Incorporating with a non-spanning tree in TCA allows us to model inter-cluster independence, while providing a rich but tractable model for intra-cluster dependence. This is desirable for clustering since an exact graphical model for clusters of variables would have no edges between nodes that belong to different clusters and would be fully connected within a cluster. In order for non-spanning trees to be allowed, the following prior term (penalty term), $\zeta(T) = \log p(T)$, was considered in [10]:

$$\zeta(T) = \log p(T) = \sum_{(i,j)\in\mathcal{E}} \zeta_{ij}^0 + f(\#(T)), \tag{6}$$

where ζ_{ij}^0 is a fixed weight of (i, j), f is a concave function, and $\#(T)$ is the number of edges in T.

Model parameters $\boldsymbol{W}$ and non-spanning trees T in TCA are determined by alternatively minimizing the objective function $\tilde{\mathcal{J}} = \mathcal{J}(\boldsymbol{x}, \boldsymbol{W}, T) - \zeta(T)$ [1]. Minimization of the objective function with respect to the discrete variable T, is solved by a greedy algorithm involving the maximum weight forest problem. The second minimization with respect to $\boldsymbol{W}$, is done by the gradient descent method. More details on TCA are found in [10].

3 Proposed Method for Clustering

ICA has been successfully applied to clustering genes from expression data in a non-mutually exclusive manner [5,6]. Each independent component is assumed to be a numerical realization of a biological process relevant to gene expression. The genes having extremely large or small values of the independent component can be regarded as significantly up-regulated or down-regulated genes. However, the assumption that the hidden variables are mutually independent is too strong to model the real biological processes of gene expression properly. This limitation of ICA-based method of clustering can be solved by using TCA. The tree-structured graphical model of TCA is enough rich to model the real biological processes. The procedures of TCA-based clustering are summarized below.

Algorithm Outline: TCA-Based Clustering

Step 1 [Preprocessing]. The gene expression data matrix $\boldsymbol{X}$ is preprocessed such that each element is associated with $X_{it} = \log_2 R_{it} - \log_2 G_{it}$ where R_{it} and G_{it} represent the red and green intensity of cDNA microarray, respectively. Genes whose profiles have missing values more than 10% are discarded. Missing values in $\boldsymbol{X}$ are filled in by applying the *KNNimpute*, a method based on k-nearest neighbors [19]. The data matrix is centered such that each row vector has zero mean. In the case of high-dimensional data, PCA could be applied to reduce the dimension, but it is not always necessary.

Step 2 [Decomposition]. We apply the TCA algorithm to the preprocessed data matrix to estimate the demixing matrix $\boldsymbol{W}$ and the encoding variable matrix $\boldsymbol{S}$.

Step 3 [Gene clustering]. In the case of ICA, the row vectors of $\boldsymbol{S}$ are statistically independent. Thus clustering is carried out for each row vector (associated with each linear mode that is the column vector of $\boldsymbol{A}$). In other words, for each row vector of $\boldsymbol{S}$, genes with strong positive and negative values of associated independent components, are grouped into two clusters,

[1] This objective function is the case where whitening constraints are imposed. In such a case, the minimization is carried out subject to $\boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{W}^\top = \boldsymbol{I}$ where $\boldsymbol{\Sigma}$ is the covariance matrix of $\boldsymbol{x}$.

each of which is related to induced and repressed genes, respectively. On the other hand, TCA reveals a dependency structure in the row vectors of $\boldsymbol{S}$. Hence, the row vectors of $\boldsymbol{S}$ associated with a spanning tree undergo a weighted sum. These resulting row vectors (the number of these row vectors is equal to the number of spanning trees in the forest) are used for grouping genes into up-regulated and down-regulated genes. Denote by $\mathcal{C}_i$ the cluster associated with an isolated spanning tree determined by TCA. The up-regulated ($\mathcal{C}_i^u$) and down-regulated ($\mathcal{C}_i^d$) genes are grouped by the following rule:

$$\mathcal{C}_i^u = \left\{ \text{gene } j \;\middle|\; \sum_{k \in \mathcal{C}_i} \|\boldsymbol{a}_k\|_2^2 \, \text{sign}(\overline{\boldsymbol{a}_k}) \, S_{kj} \geq c\,\sigma \right\},$$
$$\mathcal{C}_i^d = \left\{ \text{gene } j \;\middle|\; \sum_{k \in \mathcal{C}_i} \|\boldsymbol{a}_k\|_2^2 \, \text{sign}(\overline{\boldsymbol{a}_k}) \, S_{kj} \leq -c\,\sigma \right\}, \tag{7}$$

where σ denotes the standard deviation of $\sum_{k \in \mathcal{C}_i} \|\boldsymbol{a}_k\|_2^2 \, \text{sign}(\overline{\boldsymbol{a}_k}) \, S_{k,:}$, where $\overline{\boldsymbol{a}_k}$ is the average of $\boldsymbol{a}_k$ and $S_{k,:}$ is the kth row vector of $\boldsymbol{S}$. In our experiment, we chose $c = 1.5$.

4 Numerical Experiments

4.1 Datasets

We used three publicly available gene expression time series data sets, including yeast sporulation, metabolic shift, and cell cycle-related data. The details on these data sets are described in Table 1.

4.2 Performance Evaluation

Evaluating statistical significance of clustering is one of the most important and difficult steps in clustering gene expression data [1]. For biologists, the contents of a cluster should be correctly interpreted in order to extract biologically valuable

Table 1. The three data sets are summarized. The number of open reading frames (ORF) represents the total number of genes which are not discarded in the preprocessing step. The number of time points is equal to the dimension of the observation vector $\mathbf{x}$. We chose the number of clusters of hidden variables by using the TCA algorithm.

No.	Dataset	# of ORFs	# of time points	# of clusters	Reference
D1	sporulation	6118	7	2	[20]
D2	metabolic	6314	7	3	[21]
D3	cdc28	5574	17	9	[22]

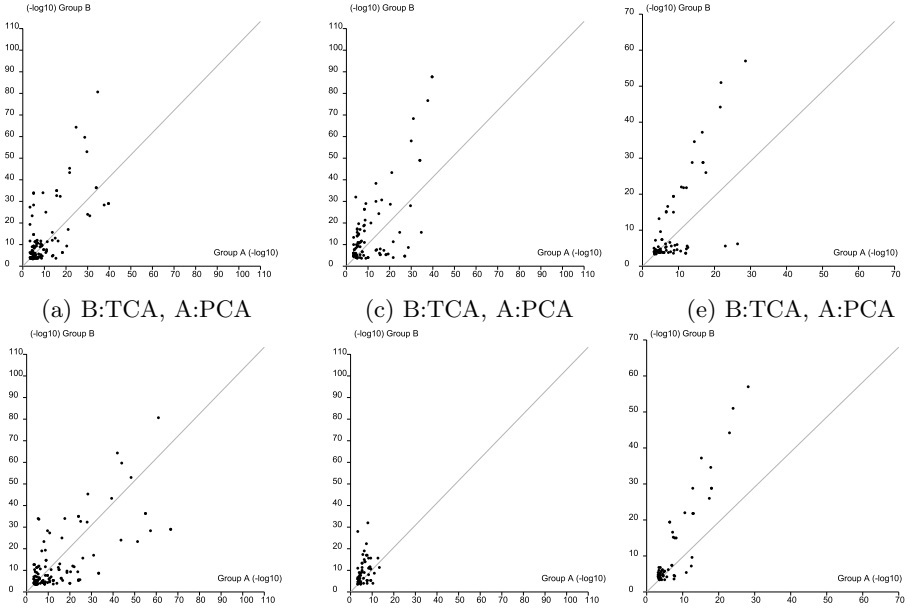
information from the results of clustering. The correct interpretation is guided by the analysis of statistical significance of clustering. In statistics, statistical significance is usually determined in the framework of hypothesis testing considering the null and alternative hypotheses. To apply the hypothesis testing framework to this work, we use the *Gene Ontology* (GO) database annotating gene products of many well-known genomes in terms of their associated biological processes, cellular components, and molecular functions [11]. From the gene list of a cluster, we obtain several annotation categories in which some genes of the cluster are contained. If the genes contained in a certain annotation category are observed within the cluster by chance, the number of genes follows the hypergeometric distribution. This is the null hypothesis H_0 and the opposite one is called the alternative hypothesis H_1. Under the null hypothesis H_0, the p-value of the probability to observe the number of genes as large or larger than k from an annotation category within a cluster of size n is given by

$$p = 1 - \sum_{i=0}^{k-1} \frac{\binom{f}{i}\binom{g-f}{n-i}}{\binom{g}{n}} \tag{8}$$

where f is the total number of genes within an annotation category of the GO database and g is the total number of genes within the genome. If the p-value is smaller than a fixed significance level α, we reject the null hypothesis H_0 and conclude that the genes contained in the annotation category are statistically significant [1]. To compare the statistical significance of two clustering results, we collect the minimum p-value smaller than α for each annotation category observed in both clustering results. A scatter plot of the negative logarithm of the collected p-values are finally drawn for visual comparison [5]. In the experiments, we set $\alpha = 0.005$ for the significance level. We have developed a software called *GOComparator* which calculates p values of GO annotations and compares the two clustering results visually by plotting the minimum p-values shared in both. It is freely available at *http://home.postech.ac.kr/∼blkimjk/software.html*.

4.3 Results

We compared the performance of TCA-based clustering with PCA and ICA by using the three yeast datasets. The method of clustering with the two algorithms is very similar to TCA except that decomposition is performed by PCA and ICA, respectively. In addition, the weighted summation of tree-dependent components in the gene clustering step is not done as there are no clusters of hidden variables in the two algorithms. We compared three different ICA algorithms to choose one showing the best clustering performance in ICA-based clustering. The used ICA algorithms are Self Adaptive Natural Gradient algorithm with nonholonomic constraints (SANG), Joint Approximate Diagonalization of Eigenmatrices (JADE), and Fixed-Point ICA (FPICA) [23]. Among the three ICA algorithms, SANG shows the best performance in terms of statistical significance of GO annotations for each dataset. We also compared TCA algorithms



(a) B:TCA, A:PCA (c) B:TCA, A:PCA (e) B:TCA, A:PCA

(b) B:TCA, A:ICA (d) B:TCA, A:ICA (f) B:TCA, A:ICA

Fig. 1. Comparison of TCA based clustering to PCA and ICA on three yeast datasets. For each dataset, TCA has more points above the diagonal, which indicates that TCA has more significant GO annotations. (a), (b): D1, (c), (d): D2, (e), (f): D3.

with different empirical contrast functions: CUM, KGV, KDE, and STAT. The TCA algorithm based on Gaussian stationary process (STAT) outperforms the others for each dataset. The performance of TCA with a non-spanning tree was better than that of a spanning tree. The comparison results of three datasets are shown in Fig. 1. It confirms that TCA-based clustering outperforms PCA- and ICA-based clustering. The number of clusters of tree-dependent components chosen by TCA is given in Table 1. By applying PCA, we reduced the number of hidden variables in PCA- and ICA-based clustering to the chosen number of clusters of TCA-based clustering. Because of the computational cost of TCA, we reduced the dimension of the data vector to 10 by applying PCA for the dataset D3. For each dataset, the edge prior,ζ_{ij}^0, in (6) was chosen to $\frac{8\log(N)}{N}$, where N is the total number of genes.

The clustering based on the linear latent variable models can reveal hidden biological processes determining gene expression patterns. In the case of TCA-based clustering, each non-spanning tree corresponds to an unknown biological process. The characteristics of the unknown biological processes can be revealed by referring to the most significant GO annotations. The most significant GO annotations of the dataset D2 selected by TCA are given in Table 2. The dataset D2 shows the diauxic shift which is a switch from anaerobic growth to aerobic respiration upon depletion of glucose [21]. The selected significant GO

Table 2. The most significant GO annotations of the dataset D2 selected by TCA. The results of cluster 2 are not shown since it did not contain any significant GO annotations.

Cluster	Induced functions	Repressed functions
1	sporulation, spore wall assembly	structural molecule activity, macromolucule biosynthesis
3	aerobic respiration, cellular respiration, carbohydrate metabolism	ribosome biogenesis and assembly, cytoplasm organization and biogenesis

annotations of the cluster 3 represent the unknown biological processing related with the diauxic shift of yeast.

5 Conclusions

In this paper, we have presented a method of TCA-based clustering for gene expression data. Empirical comparison to PCA and ICA, with three different yeast data sets, has shown that the TCA-based clustering is more useful for grouping genes into biologically relevant clusters and for finding underlying biological processes. The success of TCA-based clustering has confirmed that a tree-structured graph (a forest consisting of Chow-Liu trees) for latent variables is a more realistic and richer model for modelling hidden biological processes.

Acknowledgments. This work was supported by National Core Research Center for Systems Bio-Dynamics and POSTECH Basic Research Fund.

References

1. Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., Church, G.M.: Systematic determination of genetic network architecture. Nature Genetics **22** (1999) 281–285
2. Raychaudhuri, S., Stuart, J.M., Altman, R.B.: Principal components analysis to summarize microarray experiments: Application to sporulation time series. In: Proc. Pacific Symp. Biocomputing. (2000) 452–463
3. Girolami, M., Breitling, R.: Biologically valid linear factor models of gene expression. Bioinformatics **20** (2004) 3021–3033
4. Liebermeister, W.: Linear modes of gene expression determined by independent component analysis. Bioinformatics **18** (2002) 51–60
5. Lee, S., Batzoglou, S.: ICA-based clustering of genes from microarray expression data. In: Advances in Neural Information Processing Systems. Volume 16., MIT Press (2004)
6. Kim, S., Choi, S.: Independent arrays or independent time courese for gene expression data. In: Proc. IEEE Int'l Symp. Circuits and Systems, Kobe, Japan (2005)
7. Kim, H., Choi, S., Bang, S.Y.: Membership scoring via independent feature subspace analysis for grouping co-expressed genes. In: Proc. Int'l Joint Conf. Neural Networks, Portland, Oregon (2003)

8. Kim, H., Choi, S.: Independent subspaces of gene expression data. In: Proc. IASTED Int'l Conf. Artificial Intelligence and Applications, Innsbruck, Austria (2005)
9. Kim, S., Choi, S.: Topographic independent component analysis of gene expression time series data. In: Proc. Int'l Conf. Independent Component Analysis and Blind Signal Separation, Charleston, South Carolina (2006) 462–469
10. Bach, F.R., Jordan, M.I.: Beyond independent components: Trees and clusters. Journal of Machine Learning Research **4** (2003) 1205–1233
11. Ashburner, M., Ball, C.A., *et al.*: Gene ontology: Tool for the unification of biology. Nature Genetics **25** (2000) 25–29
12. Diamantaras, K.I., Kung, S.Y.: Principal Component Neural Networks: Theory and Applications. John Wiely & Sons, INC (1996)
13. Jolliffe, I.T.: Principal Component Analysis, 2nd Edition. Spinger (2002)
14. Choi, S.: On variations of power iteration. In: Proc. Int'l Conf. Artificial Neural Networks. Volume 2., Warsaw, Poland (2005) 145–150
15. Alter, O., Brown, P.O., Botstein, D.: Singular value decomposition for genome-wide expression data processing and modeling. Proceedings of the National Academy of Sciences, USA **97** (2000) 10101–10106
16. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley & Sons, Inc. (2001)
17. Cichocki, A., Amari, S.: Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications. John Wiley & Sons, Inc. (2002)
18. Choi, S., Cichocki, A., Park, H.M., Lee, S.Y.: Blind source separation and independent component analysis: A review. Neural Information Processing - Letters and Review **6** (2005) 1–57
19. Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., Altman, R.B.: Missing value estimation methods for DNA microarrays. Bioinformatics **17** (2001) 520–525
20. Chu, S., DeRisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P.O., Herskowitz, I.: The transcriptional program of sporulation in budding yeast. Science **282** (1998) 699–705
21. DeRisi, J.L., Iyer, V.R., Brown, P.O.: Exploring the metabolic and genetic control of gene expression on a genomic scale. Science **278** (1997) 680–686
22. Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. Molecular Biology of the Cell **9** (1998) 3273–3297
23. Cichocki, A., Amari, S., Siwek, K., Tanaka, T.: ICALAB Toolboxes (2002)

A Neural Model in Anti-spam Systems

Otávio A.S. Carpinteiro, Isaías Lima, João M.C. Assis,
Antonio C. Zambroni de Souza, Edmilson M. Moreira,
and Carlos A.M. Pinheiro

Research Group on Computer Networks and Software Engineering
Federal University of Itajubá
37500–903, Itajubá, MG, Brazil
{otavio, isaias, jmcassis, zambroni, edmarmo, pinheiro}@unifei.edu.br

Abstract. The paper proposes the use of the multilayer perceptron model to the problem of detecting ham and spam e-mail patterns. It also proposes an intensive use of data pre-processing and feature selection methods to simplify the task of the multilayer perceptron in classifying ham and spam e-mails. The multilayer perceptron is trained and assessed on patterns extracted from the SpamAssassin Public Corpus. It is required to classify novel types of ham and spam patterns. The results are presented and evaluated in the paper.

1 Introduction

Neural networks (NNs) have been widely employed in pattern recognition problems [1]. In this class of problems, they present several advantages over other mathematical models. For instance, they make use of parallel processing and present graceful degradation. Moreover, by inductive learning, they establish themselves the function which maps the set of inputs on the set of outputs [2].

The main advantage, however, is the fact that NNs are capable of generalization. They can produce correct outputs even on inputs that were never presented to them during training [2].

The generalization property is particularly interesting in the domain of anti-spam systems, i.e., systems which filter spam electronic mails (e-mails). Spam e-mails are unsolicited electronic messages posted blindly to many recipients usually for commercial advertisement.

Spam is a costly problem, and it is getting worse, for the number of spam e-mails circulating in computer networks is increasing rapidly [3,4,5]. Anti-spam systems are becoming increasingly complex [6]. Nonetheless, they commonly fail on blocking both known forms and novel forms of spam e-mails [5].

There are some works in the literature suggesting the application either of statistical models [7,8,9] or of neural network models [10,8,11] to anti-spam systems. Yet, whether they be statistical-based or NN-based, anti-spam systems share the same failing, which is, the number of false positives and false negatives generated is too high[1] [12].

[1] The classification of a normal mail as spam, and of spam as a normal mail is referred to as *false positive*, and *false negative*, respectively.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 847–855, 2006.

The objective of this paper is twofold. First, it is proposed the use of the multilayer perceptron (MLP) to approach the problem of detecting ham[2] and spam e-mails. MLP has been extensively applied to pattern recognition, as well as to several other categories of problems [1]. Second, it is proposed an intensive use of data pre-processing and feature selection methods. The use of such methods simplifies the task of the MLP in classifying ham and spam e-mails.

The paper is divided as follows. The second and third sections present the data pre-processing and feature selection methods, respectively. MLP is detailed in the fourth section. The fifth section describes the experiments. The sixth one discusses the results on the recognition of ham and spam e-mails. The last section presents the main conclusions of the paper, and indicates some directions for future work.

2 Data Pre-processing

The data consist of ham and spam e-mails extracted from the SpamAssassin Public Corpus [13]. This public corpus was chosen because its e-mails are complete, and kept in their original form.

The e-mails were pre-processed to make them simpler, more uniform, and to eliminate unnecessary parts. Many pre-processing operations were realized on the text, images, and on HTML tags.

2.1 Pre-processing on Text and Images

Some of the pre-processing operations realized on text and images are described below.

1. Letters are converted into lower case.
2. Images are removed. A tag is included to replace each image.
3. Only subject and body are considered. Attachments are removed. A tag is included to replace each attachment. All other parts of the e-mail are discarded.
4. Contents of e-mails may be presented in both HTML and text forms. Whenever both forms occur, the text form is discarded.
5. Hyphenated words are removed. A tag is included to replace each one of them.
6. Spaces and other delimiting characters between letters are removed.
7. Accent marks are removed from the accented letters.
8. One- or two-letter words are ignored.
9. Long words are discarded. A tag is included to replace each one of them.
10. Links, e-mail addresses, currency, and percentage are converted into special tags.
11. Numbers in e-mail subjects are removed. A tag is included to replace each one of them.

[2] Normal mails are referred to as *ham mails*.

2.2 Pre-processing on HTML Tags

HTML tags are divided into three categories. They are processed according to the category which they belong to. Table 1 presents some of the tags processed, and their respective categories.

Table 1. Some HTML tags, and their respective categories

Tag	Category	Tag	Category
a	3	html	2
abbr	2	i	2
acronym	2	img	3
b	2	input	3
base	3	ins	2
blockquote	3	kbd	2
body	2	label	2
br	2	li	2
button	3	map	3
caption	2	marquee	1
col	2	ol	2
comment	1	option	2
del	2	p	2
em	2	select	2
font	3	style	1
form	3	table	2
frame	2	textarea	2
h1–h6	2	title	1
head	2	tr	2
hr	2	var	2

Tags in the first category are totally discarded, that is, the tags, their attributes, and the contents they enclose are completely removed. For instance, the block "*<style> anything inside </style>*" is totally discarded during pre-processing.

Tags in the second category have their attributes removed during pre-processing. The tag itself is replaced by another special one. For instance, the block "* anything inside *" is converted into "*!_in_p anything inside*" during pre-processing.

Tags in the third category are processed in their entirety. Nonetheless, the contents of the attributes are removed, and the tag itself is replaced by another special one. For instance, the block "*<form action="results.php"> anything inside </form>*" is converted into "*!_in_form action anything inside*" during pre-processing.

3 Feature Selection Methods

Feature selection consists in extracting the most relevant features from a information set. Considering the information set as being the whole set of e-mails, the features consist then of the e-mail words, images, HTML tags and attributes which were pre-processed.

Feature selection methods are widely employed in text categorization [14]. In particular, those methods can also be employed in the categorization of e-mails into two classes — ham and spam.

Two feature selection methods were employed in the experiments — frequency distribution (FD) and chi-square (χ^2) distribution. These methods are described below.

3.1 Frequency Distribution

Frequency distribution (FD) measures the degree of occurrence of an element w in a set C. If w is a feature, the frequency distribution of the feature w is given by

$$FD(w) = \frac{N[w \in \{spam, ham\}]}{T} \tag{1}$$

where $N[w \in \{spam, ham\}]$ is the number of occurrences of feature w in the classes $\{spam, ham\}$, and T the total number of features in those classes. The features with the highest values of FD are then selected. Each selected feature is represented by one input unit of the neural model.

3.2 Chi-square Distribution

Chi-square (χ^2) distribution measures the degree of dependence between an element e and a set S [15]. If w is a feature, and C a set of two classes — spam and ham —, the chi-square distribution of the feature w is given by

$$\chi^2(w) = P(spam) \cdot \chi^2(w, spam) + P(ham) \cdot \chi^2(w, ham) \tag{2}$$

where $P(spam)$ and $P(ham)$ are the probabilities of occurrence of spam and ham e-mails, respectively. The chi-square distribution for the feature w and class c is given by

$$\chi^2(w, c) = \frac{N(kn - ml)^2}{(k + m)(l + n)(k + l)(m + n)} \tag{3}$$

where k is the number of e-mails, within class c, which contain the feature w; l is the number of e-mails, within class $\overline{c}$, which contain the feature w; m is the number of e-mails, within class c, which do not contain the feature w; n is the number of e-mails, within class $\overline{c}$, which do not contain the feature w; and N is the total number of e-mails within class c.

The features with the highest values of chi-square distribution are then selected. Each selected feature is represented by one input unit of the neural model.

4 Multilayer Perceptron

The multilayer perceptron (MLP) holds either six, twelve, or twenty-five input units. Several architectures including from three up to twenty hidden units are tested.

Activation a_i of each hidden unit i is given by the sigmoid function

$$a_i = \frac{1}{1 + e^{-net_i}} \tag{4}$$

net_i is given by

$$net_i = \sum_j w_{ij} a_j + bias_i \tag{5}$$

where w_{ij} is the weight from input unit j to hidden unit i, a_j is the activation of input unit j, and $bias_i$ is a special weight which adjusts values of net_i to make an efficient use of threshold of the sigmoid.

The output layer holds linear units to avoid *flat spots*[3] [16]. Activation a_i of each output unit i is thus given by

$$a_i = net_i = \sum_j w_{ij} a_j + bias_i \tag{6}$$

where w_{ij} is the weight from hidden unit j to output unit i, a_j is the activation of hidden unit j, and $bias_i$ is again a special weight[4].

Weights are updated according to generalized delta rule [17],

$$\Delta w_{ij}(p) = \alpha \delta_i a_j + \beta \Delta w_{ij}(p-1) \tag{7}$$

where $\alpha, \beta \in (0, 1)$ are the learning rate and momentum respectively. Subscript p indexes pattern number.

Training takes place on an epoch-by-epoch basis. At the end of each epoch, both learning rate and momentum are modified, and total error is calculated.

Training is performed through cross validation. Therefore, it is halted whenever the total error increases on the testing set.

Learning rate is reduced by 50% when total error increases, and increased by 2% when error decreases. Momentum is disabled until the end of training if total error increases. Total error E is given by

$$E = \sum_p \sum_i \delta_i^2(p) \tag{8}$$

[3] *Flat spots* are points in which the derivative of the sigmoid function approaches zero. The recovery of a non-linear output unit becomes extremely slow when it displays an incorrect output value on a flat spot.

[4] The existence of bias is not necessary, for the output units are linear. However, they were kept.

where subscript p indexes pattern number, and δ_i is the error signal for output unit i.

Error signal δ_i, for an output unit i, is given by

$$\delta_i = t_i - a_i \tag{9}$$

where t_i is the desired activation value and a_i is the activation obtained. For a hidden unit i, δ_i is given by

$$\delta_i = a_i(1 - a_i) \sum_k \delta_k w_{ki} \tag{10}$$

where w_{ki} is the weight from hidden unit i to output unit k.

Two output units were used in all experiments. The model was trained to display activation values (01) in these units when the units in the input layer are representing a *negative pattern*, that means, a ham e-mail. It was also trained to display values (10) when the input units are representing a *positive pattern*, i.e., a spam e-mail.

The initial weights are given randomly in the range [-0.5,0.5].

5 Experiments

Six experiments are carried out. They employ either different feature selection methods or different number of units in the input layer of the MLP.

Three sets — training set, testing set, and validation set — are prepared for each experiment. Training and testing sets are employed during training. The validation set contains totally novel patterns, i.e., novel types of ham and spam e-mails which are neither present in the training set nor in the testing set. The generalization property of the MLP is thus put to the test.

The first and second experiments make use of a MLP model with six units in its input layer. In the first experiment, the feature selection method employed is frequency distribution. In the second, chi-square distribution is the feature selection method employed.

The third and fourth experiments employ a MLP model with twelve units in its input layer. Frequency distribution and chi-square distribution are the feature selection methods employed in the third and fourth experiments, respectively.

The fifth and sixth experiments make use of a MLP model with twenty-five units in its input layer. In the fifth experiment, the feature selection method employed is frequency distribution. In the sixth, chi-square distribution.

In all experiments, from the first to the sixth, the training sets contain 2484 ham and 2484 spam patterns. The testing sets contain 832 ham and 832 spam patterns, and the validation sets 826 ham and 826 spam patterns.

6 Results

Table 2 presents the best results achieved by MLP on the validation sets. It shows the percentage of correct classifications both of ham and of spam patterns in the six experiments.

Table 2. Results of the experiments — FS Method: feature selection method; FD: frequency distribution; χ^2: chi-square distribution

Experiment	No. Inputs	FS Method	Correct Classifications (%)	
			Ham Pattern	Spam Pattern
1	6	FD	86.44	91.16
2	6	χ^2	93.58	96.49
3	12	FD	91.16	96.00
4	12	χ^2	97.34	98.18
5	25	FD	97.22	94.55
6	25	χ^2	100.00	99.15

The results from MLP are very promising. The percentage of correct classifications is high, showing that the MLP was capable of generalizing to novel types of ham and spam e-mail patterns.

In the first experiment, MLP classified incorrectly 112 ham patterns and 73 spam patterns. In the second, it classified incorrectly 53 ham patterns and 29 spam patterns.

In the third experiment, MLP classified incorrectly 73 ham patterns and 33 spam patterns. In the fourth, it classified incorrectly 22 ham patterns and 15 spam patterns.

In the fifth experiment, MLP classified incorrectly 23 ham patterns and 45 spam patterns. In the sixth, it classified correctly all ham patterns, and classified incorrectly 7 spam patterns.

To provide a better understanding of the quality of these results, a comparison with the results reported in a very recent paper is presented below.

Chuan, Xianliang, Mengshu, and Xu [11] carried out experiments with three anti-spam filter models on the SpamAssassin Public Corpus. The first model was a naïve Bayesian classifier (NBC), the second a multilayer perceptron (MLP), and the third a learning vector quantization (LVQ).

Their paper reports that NBC model achieved 86.48% of correct classifications on spam patterns, whilst MLP and LVQ models achieved 91.26% and 93.58%, respectively. These results are poor when compared with those shown in table 2.

7 Conclusion

The paper proposes the use of the multilayer perceptron (MLP) model to the problem of detecting ham and spam e-mail patterns. MLP has been extensively applied to pattern recognition, as well as to several other categories of problems.

The paper proposes an intensive use of data pre-processing and feature selection methods as well. The use of such methods simplifies the task of the MLP in classifying ham and spam e-mails.

MLP is trained and assessed on patterns extracted from the SpamAssassin Public Corpus. The data include both ham and spam e-mail patterns. The

e-mails are pre-processed to make them simpler, more uniform, and to eliminate unnecessary parts. Feature selection methods are employed to extract the most relevant features from the e-mails.

The experiments show that the multilayer perceptron performed very well. The percentage of correct classifications is high, showing that the MLP is capable of generalizing to novel types of ham and spam patterns.

The results achieved may still be improved. Some directions for further work include testing novel neural models as well as testing novel data pre-processing and feature selection methods.

Acknowledgment

This research is supported by CNPq and FAPEMIG, Brazil.

References

1. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press (1995)
2. Haykin, S.: Neural Networks: A Comprehensive Foundation. 2 edn. Prentice-Hall, Inc. (1999)
3. Fawcett, T.: "In vivo" spam filtering: A challenge problem for KDD. ACM SIGKDD Explorations **5** (2003) 140–148
4. Gomes, L.H., Cazita, C., Almeida, J.M., Almeida, V., Meira Junior, W.: Characterizing a spam traffic. In: Proceedings of the Internet Measurement Conference, ACM SIGCOMM (2004)
5. Pfleeger, S.L., Bloom, G.: Canning spam: Proposed solutions to unwanted email. IEEE Security & Privacy **3** (2005) 40–47
6. Cournane, A., Hunt, R.: An analysis of the tools used for the generation and prevention of spam. Computers & Security **23** (2004) 154–166
7. Androutsopoulos, I., Koutsias, J., Chandrinos, K.V., Paliouras, G., Spyropoulos, C.D.: An evaluation of naive Bayesian anti-spam filtering. In: Proceedings of the Workshop on Machine Learning in the New Information Age. (2000) 9–17
8. Özgür, L., Güngör, T., Gürgen, F.: Adaptive anti-spam filtering for agglutinative languages: a special case for Turkish. Pattern Recognition Letters **25** (2004) 1819–1831
9. Zhang, L., Zhu, J., Yao, T.: An evaluation of statistical spam filtering techniques. ACM Transactions on Asian Language Information Processing **3** (2004) 243–269
10. Drucker, H., Wu, D., Vapnik, V.N.: Support vector machines for spam categorization. IEEE Transactions on Neural Networks **10** (1999) 1048–1054
11. Chuan, Z., Xianliang, L., Mengshu, H., Xu, Z.: A LVQ-based neural network anti-spam email approach. ACM SIGOPS Operating Systems Review **39** (2005) 34–39
12. Zorkadis, V., Karras, D.A., Panayotou, M.: Efficient information theoretic strategies for classifier combination, feature extraction and performance evaluation in improving false positives and false negatives for spam e-mail filtering. Neural Networks **18** (2005) 799–807
13. Internet web page: The Apache SpamAssassin Project. The Apache Software Foundation. (2006) *http://spamassassin.apache.org/publiccorpus/*.

14. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proceedings of the International Conference on Machine Learning. (1997)
15. Papoulis, A., Pillai, S.U.: Probability, Random Variables, and Stochastic Processes. 4 edn. McGraw-Hill (2001)
16. Fahlman, S.E.: An empirical study of learning speed in back-propagation networks. Technical Report CMU-CS-88-162, School of Computer Science — Carnegie Mellon University, Pittsburgh, PA (1988)
17. Rumelhart, D.E., Hinton, G.E., McClelland, J.L.: A general framework for parallel distributed processing. In Rumelhart, D.E., McClelland, J.L., the PDP Research Group, eds.: Parallel Distributed Processing. Volume 1. The MIT Press, Cambridge, MA (1986) 45–76

A Neural Model in Intrusion Detection Systems

Otávio A.S. Carpinteiro, Roberto S. Netto, Isaías Lima,
Antonio C. Zambroni de Souza, Edmilson M. Moreira,
and Carlos A.M. Pinheiro

Research Group on Computer Networks and Software Engineering
Federal University of Itajubá
37500–903, Itajubá, MG, Brazil
{otavio, rsnetto, isaias, zambroni, edmarmo, pinheiro}@unifei.edu.br

Abstract. The paper proposes the use of the multilayer perceptron model to the problem of detecting attack patterns in computer networks. The multilayer perceptron is trained and assessed on patterns extracted from the files of the Third International Knowledge Discovery and Data Mining Tools Competition. It is required to classify novel normal patterns and novel categories of attack patterns. The results are presented and evaluated in the paper.

1 Introduction

Neural networks (NNs) have been widely employed in pattern recognition problems [1]. In this class of problems, they present several advantages over other mathematical models. For instance, they make use of parallel processing and present graceful degradation. Moreover, by inductive learning, they establish themselves the function which maps the set of inputs on the set of outputs [2].

The main advantage, however, is the fact that NNs are capable of generalization. They can produce correct outputs even on inputs that were never presented to them during training [2].

The generalization property is particularly interesting in the domain of intrusion detection systems (IDSs) in computer networks [3,4,5]. Novel types of attack emerge frequently across the globe, and spread rapidly on computer systems. IDSs currently in operation are requested to provide defense against these novel types of attack, and commonly fail [6,7,8].

There are some works in the literature suggesting the application of neural networks to IDSs [9,10,11,12]. Their main objective is to show that NNs may provide a better solution to the problem of detecting both known types and novel types of attack patterns.

This paper proposes the use of the multilayer perceptron (MLP) to approach the problem of detecting normal patterns and attack patterns in computer networks. MLP has been extensively applied to pattern recognition, as well as to several other categories of problems [1].

The paper is divided as follows. The second section presents the data representation. MLP is detailed in the third section. The fourth section describes

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 856–862, 2006.

the experiments. The fifth section discusses the results both on normal and on attack pattern recognition. The last section presents the main conclusions of the paper, and indicates some directions for future work.

2 Data Representation

The input data consists of patterns extracted from the files of the Third International Knowledge Discovery and Data Mining Tools Competition (KDD Competition) [13]. The competition was held in conjunction with the Fifth International Conference on Knowledge Discovery and Data Mining.

The input data include both normal and attack patterns. Several input files including such data are set.

Forty-six input neural units are used in the representation, as shown in Table 1. The units represent data fields which characterize computer network traffic. Thus, the first unit, for instance, represents the duration of the network connection, and the sixteenth unit represents the number of failed logins. The KDD Competition site [14] provides an extensive explanation of these data fields.

Table 1. Data fields employed in the representation

Unit	Data Field	Unit	Data Field
1	duration	27	is_guest_login
2–8	service	28	count
9	flag	29	srv_count
10	src_bytes	30	serror_rate
11	dst_bytes	31	srv_serror_rate
12	land	32	rerror_rate
13	wrong_fragment	33	srv_rerror_rate
14	urgent	34	same_srv_rate
15	hot	35	diff_srv_rate
16	num_failed_logins	36	srv_diff_host_rate
17	logged_in	37	dst_host_count
18	num_compromised	38	dst_host_srv_count
19	root_shell	39	dst_host_same_srv_rate
20	su_attempted	40	dst_host_diff_srv_rate
21	num_root	41	dst_host_same_src_port_rate
22	num_file_creations	42	dst_host_srv_diff_host_rate
23	num_shells	43	dst_host_serror_rate
24	num_access_files	44	dst_host_srv_serror_rate
25	num_outbound_cmds	45	dst_host_rerror_rate
26	is_host_login	46	dst_host_srv_rerror_rate

The network traffic data is pre-processed using ordinary normalization. Each unit receives real values with minimum and maximum values in the [0,1] range.

3 Multilayer Perceptron

The multilayer perceptron (MLP) holds forty-six input units. Several architectures including from five up to twenty-five hidden units are tested.

Activation a_i of each hidden unit i is given by the sigmoid function

$$a_i = \frac{1}{1 + e^{-net_i}} \tag{1}$$

net_i is given by

$$net_i = \sum_j w_{ij} a_j + bias_i \tag{2}$$

where w_{ij} is the weight from input unit j to hidden unit i, a_j is the activation of input unit j, and $bias_i$ is a special weight which adjusts values of net_i to make an efficient use of threshold of the sigmoid.

The output layer holds linear units to avoid *flat spots*[1] [15]. Activation a_i of each output unit i is thus given by

$$a_i = net_i = \sum_j w_{ij} a_j + bias_i \tag{3}$$

where w_{ij} is the weight from hidden unit j to output unit i, a_j is the activation of hidden unit j, and $bias_i$ is again a special weight[2].

Weights are updated according to generalized delta rule [16],

$$\Delta w_{ij}(p) = \alpha \delta_i a_j + \beta \Delta w_{ij}(p-1) \tag{4}$$

where $\alpha, \beta \in (0,1)$ are the learning rate and momentum respectively. Subscript p indexes pattern number.

Training takes place on an epoch-by-epoch basis. At the end of each epoch, both learning rate and momentum are modified, and total error is calculated.

Training is performed through cross validation. Therefore, it is halted whenever the total error increases on the testing set.

Learning rate is reduced by 50% when total error increases, and increased by 2% when error decreases. Momentum is disabled until the end of training if total error increases. Total error E is given by

$$E = \sum_p \sum_i \delta_i^2(p) \tag{5}$$

[1] *Flat spots* are points in which the derivative of the sigmoid function approaches zero. The recovery of a non-linear output unit becomes extremely slow when it displays an incorrect output value on a flat spot.

[2] The existence of bias is not necessary, for the output units are linear. However, they were kept.

where subscript p indexes pattern number, and δ_i is the error signal for output unit i.

Error signal δ_i, for an output unit i, is given by

$$\delta_i = t_i - a_i \tag{6}$$

where t_i is the desired activation value and a_i is the activation obtained. For a hidden unit i, δ_i is given by

$$\delta_i = a_i(1 - a_i) \sum_k \delta_k w_{ki} \tag{7}$$

where w_{ki} is the weight from hidden unit i to output unit k.

Two output units were used in all experiments. The model was trained to display activation values (10) in these units when the units in the input layer are representing a *negative pattern*, that means, a normal pattern. It was also trained to display values (01) when the input units are representing a *positive pattern*, i.e., an attack pattern.

The initial weights are given randomly.

4 Experiments

Four experiments are carried out. Each experiment approaches one of the four categories of computer attacks — *user-to-root (u2r)*, *remote-to-local (r2l)*, *probe*, and *denial-of-service (DoS)*. These four categories are under the taxonomy for computer attacks introduced by the intrusion detection evaluations, which were conducted by MIT Lincoln Laboratory [17].

Three sets — training set, testing set, and validation set — are prepared for each experiment. Training and testing sets are employed during training. The validation set was prepared directly by the KDD Competition. It contains totally novel patterns, i.e., normal patterns and novel categories of attack patterns which are neither present in the training set nor in the testing set. The generalization property of the MLP is thus put to the test.

The first experiment aims at detecting *user-to-root* type attacks. In it, the training set contains 37 normal and 37 attack patterns — 17 attacks of *buffer_overflow*, 9 of *loadmodule*, 3 of *perl*, and 8 of *rootkit*. The testing set contains 37 normal and 37 attack patterns — 17 of *buffer_overflow*, 9 of *loadmodule*, 3 of *perl*, and 8 of *rootkit* — as well. The validation set contains 228 normal and 228 attack patterns — 13 attacks of *xterm*, 2 of *sqlattack*, 16 of *ps*, 13 of *rootkit*, 2 of *perl*, 2 of *loadmodule*, 22 of *buffer_overflow*, and 158 of *httptunnel*.

The second experiment aims at detecting *remote-to-local* type attacks. In it, the training set contains 328 normal and 328 attack patterns — 2 attacks of *spy*, 8 of *ftp_write*, 44 of *guess_passwd*, 35 of *imap*, 5 of *phf*, 6 of *multihop*, 144 of *warezclient*, and 84 of *warezmaster*. The testing set contains 78 normal and 78 attack patterns — 2 of *spy*, 4 of *ftp_write*, 4 of *guess_passwd*, 2 of *phf*, 3 of *multihop*, 61 of *warezclient*, and 2 of *warezmaster*. The validation set contains

336 normal and 336 attack patterns — 1 attack of *worm*, 9 of *xlock*, 4 of *xsnoop*, 219 of *snmpgetattack*, 19 of *snmpguess*, 11 of *sendmail*, 1 of *phf*, 3 of *ftp_write*, 59 of *guess_passwd*, and 10 of *named*.

The third experiment aims at detecting *probe* type attacks. In it, the training set contains 688 normal and 688 attack patterns — 137 attacks of *portsweep*, 134 of *ipsweep*, 360 of *satan*, and 57 of *nmap*. The testing set contains 387 normal and 387 attack patterns — 137 of *portsweep*, 158 of *ipsweep*, and 92 of *satan*. The validation set contains 1284 normal and 1284 attack patterns — 51 attacks of *ipsweep*, 529 of *mscan*, 35 of *nmap*, 41 of *portsweep*, 206 of *saint*, and 422 of *satan*.

The fourth experiment aims at detecting *denial-of-service* type attacks. In it, the training set contains 613 normal and 613 attack patterns — 148 attacks of *back*, 56 of *land*, 253 of *neptune*, 86 of *pod*, 34 of *smurf*, and 36 of *teardrop*. The testing set contains 207 normal and 207 attack patterns — 50 of *back*, 19 of *land*, 85 of *neptune*, 29 of *pod*, 12 of *smurf*, and 12 of *teardrop*. The validation set contains 570 normal and 570 attack patterns — 60 attacks of *back*, 9 of *land*, 111 of *neptune*, 23 of *pod*, 55 of *smurf*, 12 of *teardrop*, 2 of *udpstorm*, 151 of *processtable*, 97 of *mailbomb*, and 50 of *apache2*.

5 Results

Table 2 presents the best results achieved by MLP on the validation sets. It shows the percentage of correct classifications both of normal and of attack patterns in the four experiments.

Table 2. Results of the four experiments

Experiment	Correct Classifications (%)	
	Normal Pattern	Attack Pattern
1	100.00	100.00
2	93.15	96.73
3	95.25	100.00
4	95.44	97.02

The results from MLP are very promising. The percentage of correct classifications is high, showing that the MLP was capable of generalizing to novel normal patterns and to novel categories of attack patterns.

In the first experiment, MLP classified correctly all normal and all attack patterns. In the second experiment, MLP classified incorrectly 23 normal patterns and 11 attack patterns — 4 attacks of *xlock*, 1 of *xsnoop*, 1 of *sendmail*, 1 of *phf*, 1 of *ftp_write*, and 3 of *named*. In the third experiment, MLP classified incorrectly 61 normal patterns, and classified correctly all attack patterns. In the fourth experiment, MLP classified incorrectly 26 normal patterns and 17 attack patterns — 1 attack of *pod*, 2 of *teardrop*, 2 of *udpstorm*, and 12 of *processtable*.

6 Conclusion

The paper proposes the use of the multilayer perceptron (MLP) model to the problem of detecting attack patterns in computer networks. MLP has been extensively applied to pattern recognition, as well as to several other categories of problems.

MLP is trained and assessed on patterns extracted from the files of the KDD Competition. The data include both normal and attack patterns. It is pre-processed using ordinary normalization.

The experiments show that the multilayer perceptron performed very well. The percentage of correct classifications is high, showing that the MLP is capable of generalizing to novel normal patterns and to novel categories of attack patterns.

The results achieved may still be improved. Some directions for further work include testing novel neural models as well as testing novel representations for the data fields of network traffic.

Acknowledgment

This research is supported by CNPq and FAPEMIG, Brazil.

References

1. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press (1995)
2. Haykin, S.: Neural Networks: A Comprehensive Foundation. 2 edn. Prentice-Hall, Inc. (1999)
3. Debar, H., Dacier, M., Wespi, A.: Towards a taxonomy of intrusion-detection systems. Computer Networks **31** (1999) 805–822
4. Biermann, E., Cloete, E., Venter, L.M.: A comparison of intrusion detection systems. Computers & Security **20** (2001) 676–683
5. Bai, Y., Kobayashi, H.: Intrusion detection systems: technology and development. In: Proceedings of the 17th International Conference on Advanced Information Networking and Applications, IEEE (2003)
6. Durst, R., Champion, T., Witten, B., Miller, E., Spagnuolo, L.: Testing and evaluating computer intrusion detection systems. Communications of the ACM **42** (1999) 53–61
7. Lippmann, R., Haines, J.W., Fried, D.J., Korba, J., Das, K.: The 1999 DARPA off-line intrusion detection evaluation. Computer Networks **34** (2000) 579–595
8. Champion, T., Denz, M.L.: A benchmark evaluation of network intrusion detection systems. In: Proceedings of the Aerospace Conference, IEEE (2001)
9. Lee, S.C., Heinbuch, D.V.: Training a neural-network based intrusion detector to recognize novel attacks. IEEE Transactions on Systems, Man, and Cybernetics — Part A: Systems and Humans **31** (2001) 294–299
10. Jiang, J., Zhang, C., Kamel, M.: RBF-Based real-time hierarchical intrusion detection systems. In: Proceedings of the International Joint Conference on Neural Networks, IEEE (2003)

11. Joo, D., Hong, T., Han, I.: The neural network models for IDS based on the asymmetric costs of false negative errors and false positive errors. Expert Systems with Applications **25** (2003) 69–75
12. Zhang, C., Jiang, J., Kamel, M.: Intrusion detection using hierarchical neural networks. Pattern Recognition Letters **26** (2005) 779–791
13. Internet web page: KDD Cup 1999 Data. University of California, Irvine. (1999) *http://www.ics.uci.edu/∼kdd/databases/kddcup99/kddcup99.html*.
14. Internet web page: KDD Cup 1999 Data. University of California, Irvine. (1999) *http://www.ics.uci.edu/∼kdd/databases/kddcup99/task.html*.
15. Fahlman, S.E.: An empirical study of learning speed in back-propagation networks. Technical Report CMU-CS-88-162, School of Computer Science — Carnegie Mellon University, Pittsburgh, PA (1988)
16. Rumelhart, D.E., Hinton, G.E., McClelland, J.L.: A general framework for parallel distributed processing. In Rumelhart, D.E., McClelland, J.L., the PDP Research Group, eds.: Parallel Distributed Processing. Volume 1. The MIT Press, Cambridge, MA (1986) 45–76
17. Cabrera, J.B.D., Mehra, R.K.: Control and estimation methods in information assurance — a tutorial on intrusion detection systems. In: Proceedings of the 41st Conference on Decision and Control, IEEE (2002)

Improved Kernel Based Intrusion Detection System*

Byung-Joo Kim[1] and Il Kon Kim[2]

[1] Youngsan University Dept. of Network and Information Engineering
address: 150, Junam-ri, Ungsang-eup, Yangsan-si, Kyoungnam 626-847, Korea
phone: +82-55-380-9447
bjkim@ysu.ac.kr
[2] Kyungpook National University Department of Computer Science, Korea
ikkim@knu.ac.kr

Abstract. Computer security has become a critical issue with the rapid development of business and other transaction systems over the Internet. The application of artificial intelligence, machine learning and data mining techniques to intrusion detection systems has been increasing recently. But most research is focused on improving the classification performance of a classifier. Selecting important features from input data leads to simplification of the problem, and faster and more accurate detection rates. Thus selecting important features is an important issue in intrusion detection. Another issue in intrusion detection is that most of the intrusion detection systems are performed by off-line and it is not a suitable method for a real-time intrusion detection system. In this paper, we develop the real-time intrusion detection system, which combines an on-line feature extraction method with the on-line Least Squares Support Vector Machine classifier. Applying the proposed system to KDD CUP 99 data, experimental results show that it has a remarkable feature feature extraction, classification performance and reducing detection time compared to existing off-line intrusion detection system.

1 Introduction

Intrusion detection aims to detect intrusive activities while they are acting on computer network systems. Most intrusion detection systems(IDSs) are based on hand-crafted signatures that are developed by manual coding of expert knowledge. The major problem with this approach is that these IDSs fail to generalize to detect new attacks or attacks without known signatures. Recently, there has been an increased interest in data mining based approaches to building detection models for IDSs. These models generalize from both known attacks and normal behavior in order to detect unknown attacks. Several effective data mining techniques for detecting intrusions have been developed[1][2][3], many of which perform close to or better than systems engineered by domain experts.

* This study was supported by a grant of the Korea Health 21 R&D Project, Ministry of Health & Welfare, Republic of Korea (A05-0909-A80405-05N1-00000A).

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 863–871, 2006.

However, successful data mining techniques are themselves not enough to create effective IDSs. Despite the promise of better detection performance and generalization ability of data mining based IDSs, there are some difficulties in the implementation of the system. We can group these difficulties into three general categories: accuracy(i.e., detection performance), efficiency, and usability. In this paper, we discuss the accuracy problem in developing a real-time IDS. Another issue with an IDS is that it should operate in real-time. In typical applications of data mining to intrusion detection, detection models are produced off-line because the learning algorithms must process tremendous amounts of archived audit data. An Effective IDS should work in real-time, as intrusions take place, to minimize security compromises. Feature selection therefore is an important issue in intrusion detection.

Principal Component Analysis(PCA)[4] is a powerful technique for extracting features from data sets. For reviews of the existing literature see [5][6][7]. Traditional PCA, however, has several problems. First PCA requires a batch computation step and it causes a serious problem when the data set is large. The second problem is that, in order to update the subspace of eigenvectors with another data, we have to recompute the whole eigenspace. The finial problem is that PCA only defines a linear projection of the data. It has been shown that most of the data in the real world are inherently non-symmetrical and therefore contain higher-order correlation information that could be useful[8]. For such cases, nonlinear transforms are necessary. Recently the kernel trick has been applied to PCA and is based on a formulation of PCA in terms of the dot product matrix instead of the covariance matrix[9]. Kernel PCA(KPCA), however, requires storing and finding the eigenvectors of an $N \times N$ kernel matrix where N is a number of patterns. It is an infeasible method when N is large. This fact has motivated the development of on-line way of KPCA method which does not store the kernel matrix. It is hoped that the distribution of the extracted features in the feature space has a simple distribution so that a classifier can do a proper task. But it is pointed out that features extracted by KPCA are global features for all input data and thus may not be optimal for discriminating one class from others[9]. In order to solve this problem, we developed the two-tier based realtime intrusion detection system. Proposed real time IDS is composed of two parts. The first part is used for on-line feature extraction. The second part is used for classification. Extracted features are used as input for classification. We take on-line Least Squares Support Vector Machines(LS-SVM)[10] as a classifier. This paper is composed of as follows. In Section 2 we will briefly explain the on-line feature extraction method. In Section 3 KPCA is introduced and to make KPCA on-line, empirical kernel map method is is explained. Proposed classifier combining on-line LS-SVM with the proposed feature extraction method is described in Section 4. Experimental results to evaluate the performance of the proposed system is shown in Section 5. Discussion of the proposed IDS and future work are described in Section 6.

2 On-Line Feature Extraction

In this section, we will give a brief introduction to the method of on-line PCA algorithm which overcomes the computational complexity and memory requirement of standard PCA. Before continuing, a note on notation is in order. Vectors are columns, and the size of a vector, or matrix, where it is important, is denoted with subscripts. Particular column vectors within a matrix are denoted with a superscript, while a superscript on a vector denotes a particular observation from a set of observations, so we treat observations as column vectors of a matrix. As an example, A^i_{mn} is the ith column vector in an $m \times n$ matrix. We denote a column extension to a matrix using square brackets. Thus $[A_{mn}b]$ is an $(m \times (n+1))$ matrix, with vector b appended to A_{mn} as a last column.

To explain the on-line PCA, we assume that we have already built a set of eigenvectors $U = [u_j], j = 1, \cdots, k$ after having trained the input data $\mathbf{x}_i, i = 1, \cdots, N$. The corresponding eigenvalues are Λ and $\bar{\mathbf{x}}$ is the mean of input vector. On-line building of eigenspace requires to update these eigenspace to take into account of a new input data. Here we give a brief summarization of the method which is described in [12]. First, we update the mean:

$$\overline{x}' = \frac{1}{N+1}(N\overline{x} + x_{N+1}) \tag{1}$$

We then update the set of eigenvectors to reflect the new input vector and to apply a rotational transformation to U. For doing this, it is necessary to compute the orthogonal residual vector $\hat{h} = (Ua_{N+1} + \overline{x}) - x_{N+1}$ and normalize it to obtain $h_{N+1} = \frac{h_{N+1}}{\|h_{N+1}\|_2}$ for $\| h_{N+1} \|_2 > 0$ and $h_{N+1} = 0$ otherwise. We obtain the new matrix of Eigenvectors U' by appending h_{N+1} to the eigenvectors U and rotating them :

$$U' = [U, h_{N+1}]R \tag{2}$$

where $R \in \mathbf{R}_{(\mathbf{k+1})\times(\mathbf{k+1})}$ is a rotation matrix. R is the solution of the eigenproblem of the following form:

$$DR = R\Lambda' \tag{3}$$

where Λ' is a diagonal matrix of new Eigenvalues. We compose $D \in \mathbf{R}_{(\mathbf{k+1})\times(\mathbf{k+1})}$ as:

$$D = \frac{N}{N+1}\begin{bmatrix} \Lambda & 0 \\ 0^T & 0 \end{bmatrix} + \frac{N}{(N+1)^2}\begin{bmatrix} aa^T & \gamma a \\ \gamma a^T & \gamma^2 \end{bmatrix} \tag{4}$$

where $\gamma = h^T_{N+1}(x_{N+1} - \bar{x})$ and $a = U^T(x_{N+1} - \bar{x})$. Though there are other ways to construct the matrix D[13][14], the only method ,however, described in [12] allows for the updating of the mean.

2.1 Eigenspace Updating Criterion

The on-line PCA represents the input data with principal components $a_{i(N)}$ and it can be approximated as follows:

$$\widehat{x}_{i(N)} = Ua_{i(N)} + \bar{x} \tag{5}$$

To update the principal components $a_{i(N)}$ for a new input x_{N+1} , computing an auxiliary vector η is necessary. η is calculated as follows:

$$\eta = \left[U\widehat{h}_{N+1}\right]^T (\overline{x} - \overline{x}') \tag{6}$$

then the computation of all principal components is

$$a_{i(N+1)} = (R')^T \begin{bmatrix} a_{i(N)} \\ 0 \end{bmatrix} + \eta, \qquad i = 1, \cdots, N+1 \tag{7}$$

The above transformation produces a representation with $k+1$ dimensions. Due to the increase of the dimensionality by one, however, more storage is required to represent the data. If we try to keep a k-dimensional eigenspace, we lose a certain amount of information. It is needed for us to set the criterion on retaining the number of eigenvectors. There is no explicit guideline for retaining a number of eigenvectors. In this paper we set our criterion on adding an Eigenvector as $\lambda_{k+1}^{'} > 0.7\bar{\lambda}$ where $\bar{\lambda}$ is a mean of the λ. Based on this rule, we decide whether adding $u_{k+1}^{'}$ or not.

3 On-Line KPCA

A prerequisite of the on-line eigenspace update method is that it has to be applied on the data set. Furthermore it is restricted to apply the linear data. But in the case of KPCA this data set $\Phi(x^N)$ is high dimensional and can most of the time not even be calculated explicitly. For the case of nonlinear data set, applying feature mapping function method to on-line PCA may be one of the solutions. This is performed by so-called *kernel-trick*, which means an implicit embedding to an infinite dimensional Hilbert space[11](i.e. feature space) F.

$$K(x, y) = \Phi(x) \cdot \Phi(y) \tag{8}$$

Where K is a given kernel function in an input space. When K is semi positive definite, the existence of Φ is proven[11]. Most of the case ,however, the mapping Φ is high-dimensional and cannot be obtained explicitly. The vector in the feature space is not observable and only the inner product between vectors can be observed via a kernel function. However, for a given data set, it is possible to approximate Φ by empirical kernel map proposed by Scholkopf[15] and Tsuda[16] which is defined as $\Psi_N : \mathbf{R}^d \rightarrow \mathbf{R}^N$

$$\begin{aligned} \Psi_N(x) &= [\Phi(x_1) \cdot \Phi(x), \cdots, \Phi(x_N) \cdot \Phi(x)]^T \\ &= [K(x_1, x), \cdots, K(x_N, x)]^T \end{aligned} \tag{9}$$

A performance evaluation of empirical kernel map was shown by Tsuda. He shows that support vector machine with an empirical kernel map is identical with the conventional kernel map[17].

4 Proposed System

In previous Section 3 we proposed an on-line KPCA method for nonlinear feature extraction. It is hoped that the distribution of the mapped data in the feature space has a simple distribution so that a classifier can classify them properly. But it is point out that extracted features by KPCA are global features for all input data and thus may not be optimal for discriminating one class from others. For classification purpose, after global features are extracted using they must be used as input data for classification.

Recently LS-SVM method developed by Suykens is computationally attractive and easier to extend than SVM. But the existed LS-SVM algorithm is trained off-line in batch way. Off-line training algorithm is not fit for the realtime IDS. In this paper we take on-line LS-SVM algorithm because proposed realtime IDS to be more realistic. Proposed real time IDS is composed of two parts. First part is used for on-line feature extraction. To extract on-line nonlinear features, we propose a new feature extraction method which overcomes the problem of memory requirement of KPCA by on-line eigenspace update method incorporating with an adaptation of kernel function. Second part is used for classification. Extracted features are used as input for classification. We take on-line Least Squares Support Vector Machines(LS-SVM)[19] as a classifier.

5 Experiment

To evaluate the classification performance of proposed realtime IDS system, we use KDD CUP 99 data[18]. The following sections present the results of experiments.

5.1 Description of Dataset

The raw training data(kddcup.data.gz) was about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic. This was processed into about five million connection records. Similarly, the two weeks of test data yielded around two million connection records. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type. Each connection record consists of about 100 bytes. Attacks fall into four main categories(DOS, R2L, U2R, Probing).

It is important to note that the test data(corrected.gz) is not from the same probability distribution as the training data, and it includes specific attack types not in the training data. This makes the task more realistic. The datasets contain a total of 24 training attack types, with an additional 14 types in the test data only.

5.2 Experimental Condition

To evaluate the classification performance of proposed system, we randomly split the the training data as 80% and remaining as validation data. To evaluate the classification accuracy of proposed system we compare the proposed system to

SVM. Because standard LS-SVM and SVM are only capable of binary classification, we take multi-class LS-SVM and SVM. A RBF kernel has been taken and optimal hyper-parameter of multi-class SVM and LS-SVM[20] was obtained by 10-fold cross-validation procedure. In [19] it is shown that the use of 10-fold cross-validation for hyper-parameter selection of SVM and LS-SVMs consistently leads to very good results.

In experiment we will evaluate the generalization ability of proposed IDS on test data set since there are 14 additional attack types in the test data which are not included int the training set. To do this, extracted features by on-line KPCA will be used as input for multi-class on-line LS-SVM. Our results are summarized in the following sections.

5.3 Evaluate Feature Extraction and Classification Performance

Table 1 gives the result of extracted features for each class by on-line KPCA method.

Table 2 shows the results of the classification performance by standard SVM using all features. Table 3 shows the results of the classification performance and computing time for training and testing data by proposed system using extracted features. We can see that using important features for classification gives similar accuracies compared to using all features and the training, testing time is proper enough for realtime IDS. Comparing Table 2 with Table 3, we obtain following results. The performance of using the extracted features do not show the significant differences to that of using all features. This means that proposed on-line feature extraction method has good performance in extracting features. Proposed method has another merit in memory requirement. The advantage of proposed feature extraction method is more efficient in terms of memory

Table 1. Extracted features on each class by on-line KPCA

Class	Extracted features
Normal	1,2,3,5,6,7,8,9,10,11,12,14,16,17,18,20,21,23,25,27,29,31,32,34,38,39,41
Probe	3,5,6,24,32,38
DOS	1,3,8,19,23,28,33,35,36,39,41
U2R	5,6,15,18,25,32,33,39
R2L	3,5,6,32,33,34,35

Table 2. Classification Performance by SVM using all features

Class	Accuracy(%)
Normal	98.55
Probe	98.59
DOS	98.10
U2R	98.64
R2L	98.69

requirement than a batch KPCA because proposed feature extraction method do not require the whole N × N kernel matrix where N is the number of the training data. Second one is that proposed on-line feature extraction method has similar performance is comparable in performance to a batch KPCA.

5.4 Suitable for Realtime IDS

Table 3 shows that proposed system operates in a very quick manner whereas traditional batch system requires tremendous computational time when new training data is added. Furthermore classification accuracy of proposed system is similar to using all features. This makes proposed IDS suitable for realtime IDS.

Table 3. Performance of proposed system using extracted features

Class	Accuracy(%)	Training Time(Sec)	Testing Time(Sec)
Normal	98.54	3.12	0.9
Probe	98.64	20.25	1.14
DOS	98.48	10.79	1.10
U2R	98.91	1.2	0.84
R2L	98.74	5.7	0.6

5.5 Comparison with Batch Way LS-SVM

Recently LS-SVM is a powerful methodology for solving problems in nonlinear classification problem. To evaluate the classification accuracy of proposed system it is desirable to compare with batch way LS-SVM.

Table 4. Performance comparison of proposed method and batch way LS-SVM. using all features.

	Normal	Probe	DOS	U2R	R2L
batch LS-SVM	98.76	98.81	98.56	98.92	98.86
proposed system	98.67	98.84	98.48	98.86	98.82

Generally the disadvantage of incremental method is their accuracy compared to batch method even though it has the advantage of memory efficiency and computation time. According to Table 4 we can see that proposed method has better classification performance compared to batch way LS-SVM. By this result we can show that proposed realtime IDS has remarkable classification accuracy though it is worked by incremental way.

6 Conclusion and Remarks

This paper was devoted to the exposition of a new technique on realtime IDSs . Proposed on-line KPCA has following advantages. Firstly, The performance of

using the extracted features do not show the significant differences to that of using all features. This means that proposed on-line feature extraction method has good performance in extracting features. Secondly, proposed method has merit in memory requirement. The advantage of proposed feature extraction method is more efficient in terms of memory requirement than a batch KPCA because proposed feature extraction method do not require the whole N $\times$ N kernel matrix where N is the number of the training data. Thirdly, proposed on-line feature extraction method has similar performance is comparable in performance to a batch KPCA though it works incrementally.

Our ongoing experiment is that applying proposed system to more realistic world data to evaluate the realtime detection performance.

References

1. Eskin, E. :Anomaly detection over noisy data using learned probability distribution. In Proceedings of the Seventeenth International Conference on Machine Learning (2000) 443-482
2. Ghosh, A. and Schwartzbard, A. :A Study in using neural networks for anomaly and misuse detection. In Proceedings of the Eighth USENIX Security Symposium, (1999) 443-482
3. Lee, W. Stolfo, S.J. and Mok, K.:A Data mining in workflow environments. :Experience in intrusion detection. In Proceedings of the 1999 Conference on Knowledge Discovery and Data Mining, (1999)
4. Tipping, M.E. and Bishop, C.M. :Mixtures of probabilistic principal component analysers. Neural Computation 11(2) (1998) 443-482
5. Kramer, M.A.:Nonlinear principal component analysis using autoassociative neural networks. AICHE Journal 37(2) (1991) 233-243
6. Diamantaras, K.I. and Kung, S.Y.:Principal Component Neural Networks: Theory and Applications. New York John Wiley & Sons, Inc (1996)
7. Kim, Byung Joo. Shim, Joo Yong. Hwang, Chang Ha. Kim, Il Kon.: On-line Feature Extraction Based on Emperical Feature Map. Foundations of Intelligent Systems, volume 2871 of Lecture Notes in Artificial Intelligence (2003) 440-444
8. Softky, W.S and Kammen, D.M.: Correlation in high dimensional or asymmetric data set: Hebbian neuronal processing. Neural Networks vol. 4, Nov. (1991) 337-348
9. Gupta, H., Agrawal, A.K., Pruthi, T., Shekhar, C., and Chellappa., R.:An Experimental Evaluation of Linear and Kernel-Based Methods for Face Recognition," accessible at http://citeseer.nj.nec.com.
10. Liu, J. Chen, J.P Jiang, S. and Cheng, J.:Online LS-SVM for function estimation and classification Journal of University of Science and Technology Beijing, vol.10, Num.5, Oct. (2003) 73-77
11. Vapnik, V. N.:Statistical learning theory. John Wiley & Sons, New York (1998)
12. Hall, P. Marshall, D. and Martin, R.: On-line eigenalysis for classification. In British Machine Vision Conference, volume 1, September (1998) 286-295
13. Winkeler, J. Manjunath, B.S. and Chandrasekaran, S.:Subset selection for active object recognition. In CVPR, volume 2, IEEE Computer Society Press, June (1999) 511-516
14. Murakami, H. Kumar.,B.V.K.V.:Efficient calculation of primary images from a set of images. IEEE PAMI, 4(5) (1982) 511-515

15. Scholkopf, B. Smola, A. and Muller, K.R.:Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10(5), (1998) 1299-1319
16. Tsuda, K.:Support vector classifier based on asymmetric kernel function. Proc. ESANN (1999)
17. Mika, S.:Kernel algorithms for nonlinear signal processing in feature spaces. Master's thesis, Technical University of Berlin, November (1998)
18. Accessable at http://kdd.ics.uci.edu/databases/kddcup99
19. Gestel, V. Suykens, T. J.A.K. Lanckriet, G. Lambrechts, De Moor, A. B. and Vandewalle, J.:A Bayesian Framework for Least Squares Support Vector Machine Classifiers. Internal Report 00-65, ESAT-SISTA, K.U. Leuven.
20. Suykens, J.A.K. and Vandewalle, J.:Multiclass Least Squares Support Vector Machines. In Proc. International Joint Conference on Neural Networks (IJCNN'99), Washington DC (1999)

Testing the Fraud Detection Ability of Different User Profiles by Means of FF-NN Classifiers

Constantinos S. Hilas[1] and John N. Sahalos[2]

[1] Dept. of Informatics and Communications, Technological Educational Institute of Serres, Serres GR-621 24, Greece
chilas@teiser.gr
[2] Radiocommunications Laboratory, Aristotle University of Thessaloniki, Thessaloniki GR-541 24, Greece
sahalos@auth.gr

Abstract. Telecommunications fraud has drawn the attention in research due to the huge economic burden on companies and to the interesting aspect of users' behavior characterization. In the present paper, we deal with the issue of user characterization. Several real cases of defrauded user accounts for different user profiles were studied. Each profile's ability to characterize user behavior in order to discriminate normal activity from fraudulent one was tested. Feed-forward neural networks were used as classifiers. It is found that summary characteristics of user's behavior perform better than detailed ones towards this task.

1 Introduction

Telecommunications fraud can be simply described as any activity by which telecommunications service is obtained without intention of paying [1]. Using this definition, fraud can only be detected once it has occurred. So, it is useful to distinguish between fraud prevention and fraud detection [2]. Fraud prevention is all the measures that can be used to stop fraud from occurring in the first place. These, in the case of telecommunication systems, include Subscriber Identity Module (SIM) cards or any other Personal Identification Number (PIN) like the ones used in Private PBXs. No prevention method is perfect and usually it is a compromise between effectiveness and usage convenience. Fraud detection, on the other hand, is the identification of fraud as quickly as possible once it has happened. The problem is that fraud techniques are constantly evolving and whenever a detection method becomes known, fraudsters will adapt their strategies and try others.

Reference [1] provides a classification of telecommunication systems fraud and divides frauds into one of four groups, namely: contractual fraud, hacking fraud, technical fraud and procedural fraud. In [3], twelve distinct fraud types are identified. The authors of the present article have also witnessed fraudulent behavior that is a combination of the above mentioned ones [4].

Telecommunications fraud has drawn the attention of many researchers in the resent years not only due to the huge economic burden on companies' accountings but also due to the interesting aspect of user behavior characterization. Fraud detection

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 872 – 883, 2006.

techniques involve the monitoring of users' behavior in order to identify deviations from some expected or normal norm. Research in telecommunications fraud detection is mainly motivated by fraudulent activities in mobile technologies [1, 3, 5, 6]. The techniques used come from the area of statistical modeling like rule discovery [5, 7, 8, 9], clustering [10], Bayesian rules [6], visualization methods [11], or neural network classification [5, 12, 13]. Combinations of more than one method have also been proposed [14, 15]. In [16] one can find a bibliography on the use of data mining and machine learning methods for automatic fraud detection. Most of the aforementioned approaches use a combination of legitimate user behavior examples and some fraud examples. The aim is to detect any usage changes in the legitimate user's history.

In the present paper we are interested in the evaluation of different user representations and their effect towards the proper discrimination between legitimate and fraudulent activity. The paper proceeds as follows. In the next chapter the data that were used are described along with the different profile representation of the users. In chapter 3 the experimental procedure is presented. The experimental results are given in chapter 4. In the last chapter conclusions are drawn.

2 Profile Building

The main idea behind a user's profile is that his past behavior can be accumulated. So, a profile or a "user dictionary" of what might be the expected values of the user's behavior can be constructed. This profile contains single numerical summaries of some aspect of behavior or some kind of multivariate behavioral pattern. Future behavior of the user can then be compared with his profile. The consistency with normal behavior or the deviation from his profile may imply fraudulent activity. An important issue is that we can never be certain that fraud has been perpetrated. Any analysis should only be treated as a method that provides us with an alert or a "suspicion score". The analysis provides a measure that some observation is anomalous or more likely to be fraudulent than another. Special investigative attention should then be focused on those observations.

Traditionally, in computer security, user profiles are constructed based on any basic usage characteristic such as resources consumed, login location, typing rate and counts of particular commands. In telecommunications, user profiles can be constructed from appropriate usage characteristics. The aim is to distinguish a normal user from a fraudster. The latter is, in most of the cases, a user of the system who knows and mimics normal user behavior. All the data that can be used to monitor the usage of a telecommunications network are contained in the Call Detail Record (CDR) of any Private Branch Exchange (PBX). The CDR contains data such as: the caller ID, the chargeable duration of the call, the called party ID, the date and the time of the call, etc [17]. In mobile telephone systems, such as GSM, the data records that contain details of every mobile phone attempt are the Toll Tickets.

Our experiments are based on real data extracted from a database that holds the CDR for a period of eight years from an organization's PBX. According to the organization's charging policy, only calls to national, international and mobile destinations are charged. Calls to local destinations are not charged so they are not

included in the examples. In order to properly charge users, for the calls they place, a system of Personal Identification Numbers (PIN) is used. Each user owns a unique PIN which "unlocks" the organization's telephone sets in order to place costly outgoing calls. If anyone (e.g., a fraudster) finds a PIN he can use it to place his own calls from any telephone set within the organization.

Several user accounts, which have been defrauded, have been identified. Three of them will be presented in this paper. All three contain both examples of legitimate and fraudulent activity. The specific accounts were chosen as they contain representative types of different fraudster behavior as described below.

The detailed daily accounts were examined by a field expert and each phone call was marked as either normal or defrauded. If during a day no fraudulent activity was present then the whole day was marked as normal. If at least one call from the fraudster was present then the whole day was marked as fraud. Adding to this, each day was also marked, according to the first time that fraudulent activity appeared. Each user's account is split into two sets, one pre- and one post-fraud.

The first example (*User1*) is a case where the fraudster reveals a greedy behavior. After having acquired a user's PIN, he places a large amount of high cost calls to satellite services. The second example (*User2*) is a case where the fraudster did not place considerably costly calls but used the user's PIN during non working hours and days. In the third example (*User3*) the fraudster seems to be more cautious. He did not place costly calls and he mainly uses the account during working hours and days. An interesting observation on the third case is that the stolen PIN was used from different telephone sets and the call destinations are never associated with the legitimate user's telephone set.

For each user, three different profile types are constructed. The first profile (Profile1) is build up from the accumulated weekly behavior of the user. The profile consists of seven fields which are the mean and the standard deviation of the number of calls per week (calls), the mean and the standard deviation of the duration (dur) of calls per week, the maximum number of calls, the maximum duration of one call and the maximum cost of one call (Fig. 1). All maxima are computed within a week's period.

mean(calls)	std(calls)	mean(dur)	std(dur)	max(calls)	max(dur)	max(cost)

Fig. 1. Profile1 of telephone calls

nat_calls_w	nat_dur_w	nat_calls_a	nat_dur_a	nat_calls_n	nat_dur_n
mob_calls_w	mob_dur_w	mob_calls_a	mob_dur_a	mob_calls_n	mob_dur_n
int_calls_w	int_dur_w	int_calls_a	int_dur_a	int_calls_n	int_dur_n

Fig. 2. Profile2 of telephone calls

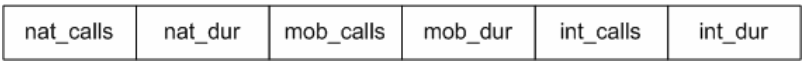


Fig. 3. Profile3 of telephone calls

The second profile (Profile2) is a detailed daily behavior of a user which is constructed by separating the number of calls per day and their corresponding duration per day according to the called destination, i.e., national (nat), international (int), and mobile (mob) calls, and the time of the day, i.e., working hours (w), afternoon hours (a), and night (n) (Fig. 2).

Last, the third profile (Profile3) is an accumulated per day behavior (Fig. 3). It consists of the number of calls and their corresponding duration separated only according to the called destination, that is, national, international and mobile calls.

The last two profiles were also accumulated per week to give Profile2w and Profile3w. So, we have 3 users, 5 profile representations, and 2 different ways to characterize the user accounts as normal of fraudulent. The above give 30 different data sets.

3 Experimental Procedure

All data were standardized so that for each characteristic the average equals zero and the variance equals one. Principal Component Analysis (PCA) was performed in order to transform the input vectors into uncorrelated ones. PCA transforms the k original variables, X_1, X_2… X_k, into p new variables, Z_1, Z_2, …,Z_p, which are linear combinations of the original ones, according to:

$$Z_i = \sum_{j=1}^{p} a_{ij} \cdot X_j \quad . \tag{1}$$

This transformation has the properties:

$Var[Z_1] > Var[Z_2] > \ldots > Var[Z_p]$, which means that Z_1 contains the most information and Z_p the least; and

$Cov[Z_i, Z_j] = 0, \forall i \neq j$,

which means that there is no information overlap between the principal components [18]. In our experiments we were mainly interested in the transformation of the input parameters to uncorrelated ones. So, we kept the 99% of the information in the data sets. However, the 1% loss of information led to a decrease in the data dimensions from 0% to 40%. The highest reduction in the data dimensionality occurred with Profile1 and Profile2. This is easily explained by two observations. According to Profile1 there are weeks where only one telephone call is placed. So, the parameters mean(calls) and max(calls) are equal. The same happens to mean(dur) and max(dur). As for Profile2 there are many zeros present in the data sets due to the absence of certain types of calls. Data sets from more active users suffer less from this dimensionality reduction.

In order to test the ability of each profile to discriminate between legitimate usage and fraud, feed-forward neural networks (FF-NN) were used as classifiers. The feed-forward neural network is defined, [12], by:

$$y = \sum_{j=0}^{M} w_j \cdot g(\sum_{i=0}^{d} w_{ji} \cdot \mathrm{Z}_i) \ . \tag{2}$$

The g is a non-linear function (e.g. $g(x) = \tanh(x)$), w_j are the weights between the output y and the hidden layer, w_{ji} are the weights from the i-th input, Z_i, to the j-th neuron of the hidden layer. Linear outputs were used.

The problem is a supervised learning one with the task to adapt the weights so that the input-output mapping corresponds to the input-output pairs the teacher has provided.

For each of the cases of input data, one feed-forward neural network was build which consisted of one hidden layer and one linear output. The size of a hidden layer is a fundamental question often raised in the application of multilayer FFNN to real world problems. It is usually determined experimentally and by empirical guidelines. For a network of a reasonable size, the size of hidden nodes needs to be only a fraction of the input layer. A simple rule of thumb is:

$$h = (m+n) \setminus 2 + \{...0...1...2...\}, \tag{3}$$

where, the symbol "\" denotes integer division, h the number of neurons in the hidden layer, m the number of neurons in the input layer and n the number of neurons in the output layer. The $+\{...0...1...2...\}$ part of the equation means that if a network fails to converge to a solution with the result of the integer division, it may be that more hidden neurons are needed. So, their number is incremented until convergence [19]. In all the experiments, that are presented here, the above scheme converged for all cases using the initial values of h.

The neural network was trained using resilient backpropagation method [20]. Each input data set was divided into training, validation and test sets. One fourth of the data was used for the validation set, one fourth for the test set and one half for the training set. The validation error was used as an early stopping criterion. In absolute values, the size of the input data sets varied from 400 to 1200 instances. 30% to 50% of the vectors in each data set were fraud cases while the remainder were non-fraud ones.

The evaluation of each classifier's performance was made by means of the corresponding Receiver Operating Characteristic (ROC) curve [21]. A ROC curve is actually a graphical representation of the trade off between the true positive and the false positive rates for every possible cut off point that separates two overlapping distributions. Equivalently, the ROC curve is the representation of the tradeoffs between the sensitivity and the specificity of a classifier.

The cases were compared only in pairs in the following sense. For each profile type and for each user the first case was the one where the user account was split in two parts, before and after the first day that a fraudulent activity appeared. The second case was the one where the same user account was split into normal and fraudulent activity using a detailed day-by-day characterization. The area under the curve was used as the statistic that exhibits the classification performance of each case.

4 Experimental Results

In the following figures (Fig.4 – Fig.8) the ROC curves for *User1* are given. Each one corresponds to the five different user profiling approaches. In each figure two lines are plotted. The solid one (uncharacterized) is the ROC curve that yielded when each case in the user's profile was characterized only relative to its position before or after the first case of fraud. The dashed line (characterized) corresponds to the profiling approach which used a thorough characterization of each case as fraudulent or not. The plots for the other users are similar.

In ROC curve analysis the area under the curve is usually used as a statistic that indicates how well a classifier performs. An area of 1 indicates a perfect classifier while an area of 0.5 indicates performance equal to random selection of cases. The areas under all the ROC curves for each user – profile combination are given in Tables 1 and 2. However, ROC curves should be judged with reservation. An example of misleading conclusion may be exhibited comparing and . According to Table 1, Profile2 seems to give better separation, as the area under the plotted curve (the "characterized" case) is larger that the corresponding area in Fig.4.

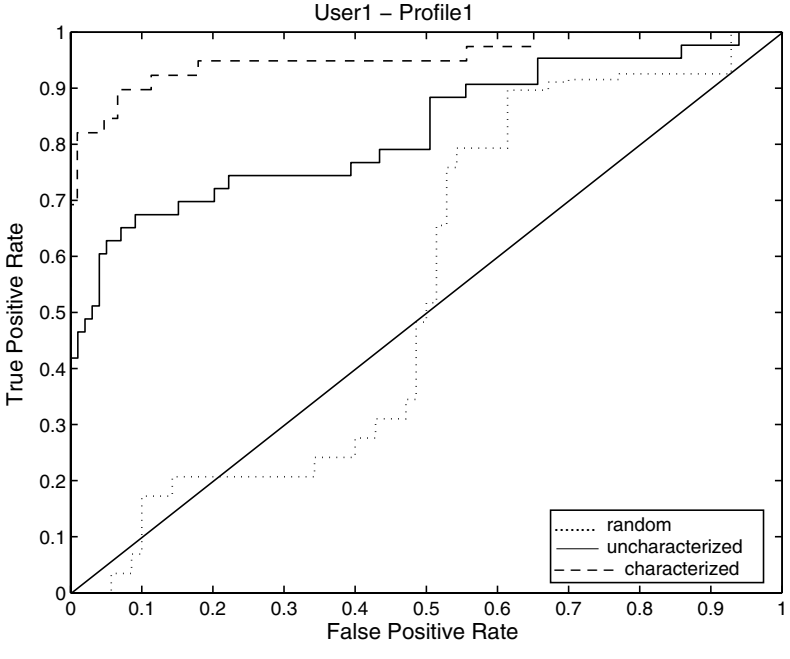


Fig. 4. ROC curves, using Profile1, showing the trade off between true-positive and false-positive rate for *User1*. His behavior is divided as fraudulent or not using the first day that a fraudulent activity appeared (uncharacterized) or using a detailed day-by-day characterization (characterized). The diagonal line is the theoretical ROC curve for random separation of the cases, while the dotted one is an instance of a real random separation.

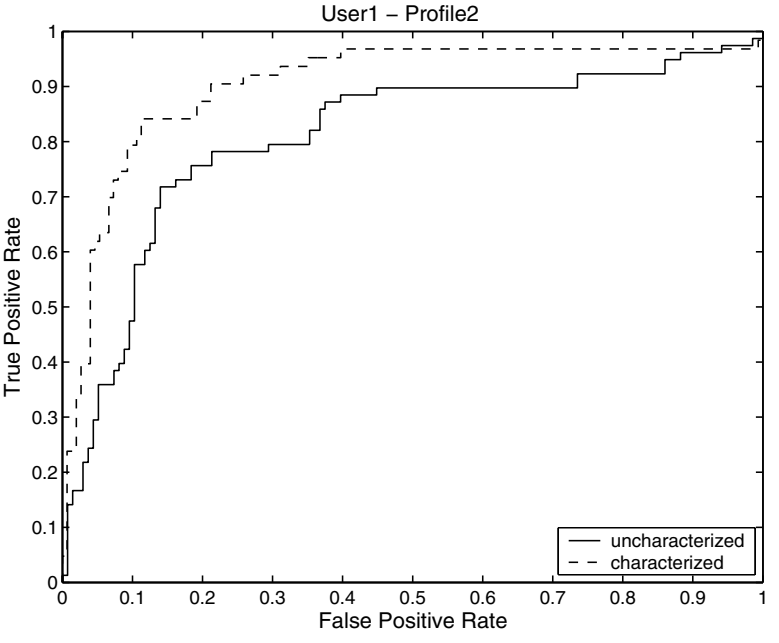


Fig. 5. ROC curves, using Profile2, showing the trade off between true-positive and false-positive rate for *User1* when his behavior is thoroughly characterized as fraudulent or not

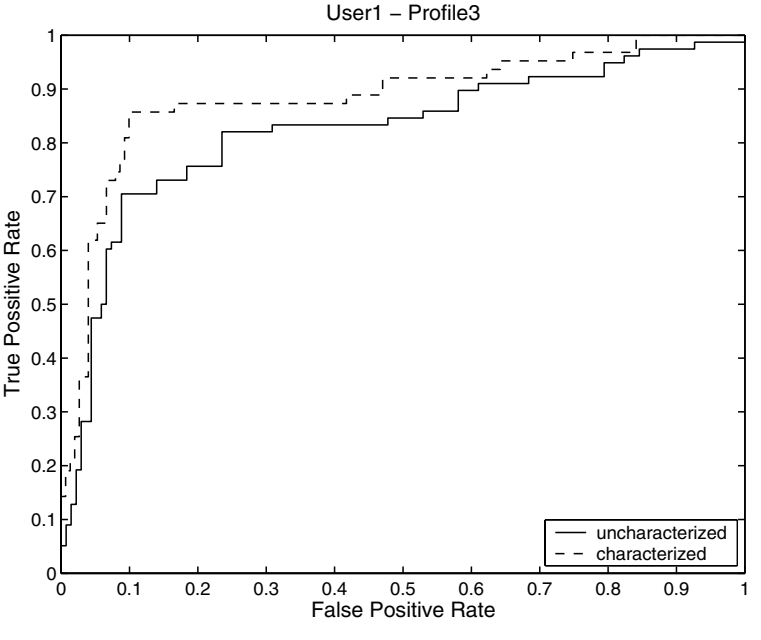


Fig. 6. ROC curves, using Profile3, showing the trade off between true-positive and false-positive rate for *User1* when his behavior is thoroughly characterized as fraudulent or not

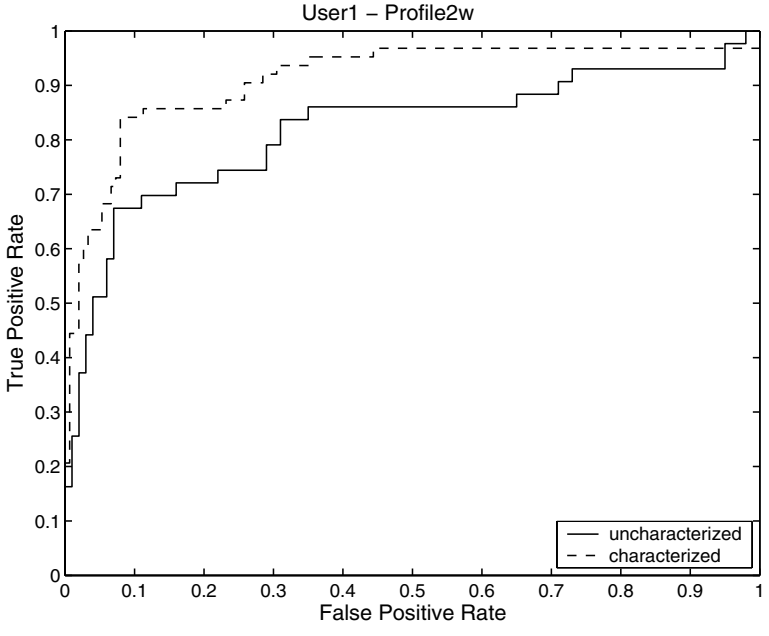


Fig. 7. ROC curves, using Profile2w, showing the trade off between true-positive and false-positive rate for *User1* when his behavior is thoroughly characterized as fraudulent or not

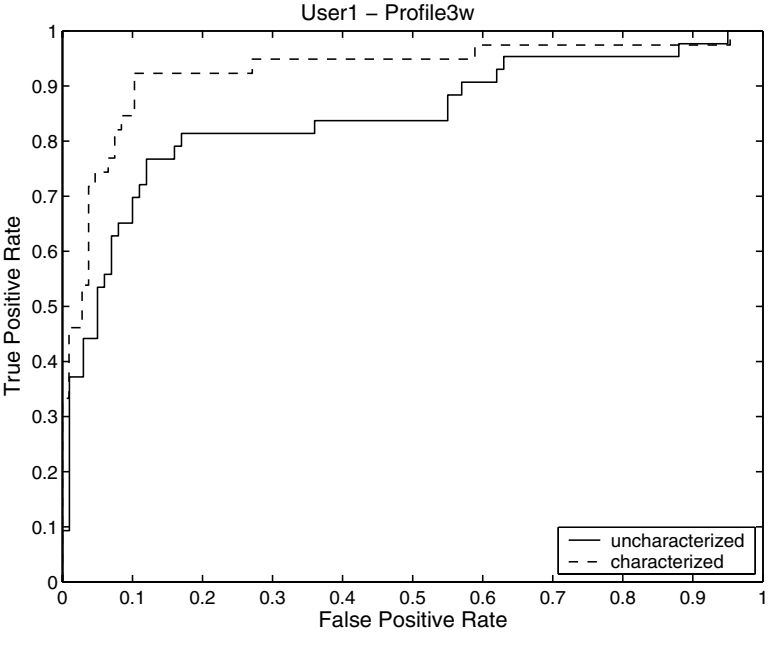


Fig. 8. ROC curves, using Profile3w, showing the trade off between true-positive and false-positive rate for *User1* when his behavior is thoroughly characterized as fraudulent or not

Close examination of the two figures reveals that by using Profile1 one gets more that 70% correct classification without any false alarms and more than 80% of true positives with only 2% of false alarms. For such small percentage of false alarms, Profile2 gives less than 25% of true positive hits.

The aforementioned comment is of great importance when working with large datasets. The telecommunications network from which we drew our examples has more than 6000 users. That means that even a false positive rate of only 1% may give up to 60 false alarms. Close examination of the above may become particularly costly to the organization in terms of lost workforce.

One should also bear in mind that the ROC curves plotted in the figures, belong to families of curves. This is due to the random initialization of the FF-NN classifiers each time they are used. So, each one of the lines depicted here is actually one characteristic instance of the family. Some additional considerations when comparing ROC curves are given in [22].

Profile1 works better than all the others as it exhibits the highest true positive rate with the smallest false positive rate. Among all of the examined user accounts, there were cases where Profile1 gave 90% positive hits without any false positive ones. Our next selection for a fraud detection technique would be Profile2 (or its weekly counterpart) which is actually a detailed profile of the user's actions. In fact, this profile could, also, be used in a rule based approach. Fraudsters tend to be greedy, which for a telecommunications environment means that they tend to talk much or to costly destinations. They are also aware of velocity traps. That is they will try to avoid using a user's account concurrently with the legitimate user. This fact concentrates their activity during the non-working hours. Separating user's activity, both by destination and by time-of-day, acts towards the identification of such actions.

The prevalent observation in all figures is that the "characterized" case gives better separation between the normal and the defrauded part of the user's account. However, a case-by-case characterization of each call is particularly expensive, and this was the main reason why only few examples are examined in the present study. Another observation is that profiles that represent characteristics which are aggregated in time outperform the daily ones. As was expected, individual characterization of each case gives better results.

Table 1. Area under ROC curves for the three basic profiles

	User1			*User2*			*User3*		
	Profile1	Profile2	Profile3	Profile1	Profile2	Profile3	Profile1	Profile2	Profile3
unchar	0.7555	0.8069	0.8238	0.6765	0.7213	0.6490	0.9047	0.7971	0.7853
char	0.9090	0.9134	0.8839	0.8345	0.8811	0.7790	0.9297	0.7789	0.7931

Table 2. Area under ROC curves for the weekly aggregated behavior

	User1		*User2*		*User3*	
	Profile2w	Profile3w	Profile2w	Profile3w	Profile2w	Profile3w
unchar	0.8181	0.8430	0.7241	0.6615	0.8741	0.8536
char	0.9120	0.9267	0.8760	0.8252	0.7298	0.8687

There are cases, like *User3*, where a cautious fraudster can hide his activity beneath the legitimate user's one. Better discrimination of the last case would have been accomplished by means of some other method, like rule discovery.

5 Conclusions and Discussion

In the present paper, tests were performed to evaluate the fraud detection ability of different user profile structures. The profiles are used as a user characterization method in order to discriminate legitimate from fraudulent usage in a telecommunications environment. Feed-forward neural networks were used as classifiers. The input data consisted of real user accounts which have been defrauded.

From the analysis it is concluded that accumulated characteristics of a user yield better discrimination results. Here, only weekly aggregation of the user's behavior was tested against detailed daily profiles. Aggregating user's behavior for larger periods was avoided in order to preserve some level of on-line detection ability.

The user profiling approaches that were presented, here, have the benefit of respecting users' privacy. That is, except from some coarse user behavior characteristics, all private data (e.g. called number, calling location, etc) are hidden from the analyst. Private data would definitely add to the accuracy of fraud detection. In fact, the expert who characterized the data sets, in the first place, used rules based on private data and his domain specific knowledge. However, our aim is to test the ability to detect fraud given the minimum possible information about the user.

Feed-forward neural networks with more hidden layers were also used. However, there was no significant rise in the performance of the classifiers.

Several modeling techniques could have, equally well, been applied, e.g. Classification and Regression Trees (CART), Adaptive Neuro-Fuzzy Systems, Support Vector Machines, etc. Each one would have revealed different aspects of the user characterization problem. Preliminary experiments with cluster analysis showed that the outcome depends on the distance measure used. For example, Euclidean distance produced a distinct cluster of outliers regardless of their class membership, while correlation separated the fraud from the non-fraud cases more clearly. Also, preliminary experimentation with classification trees, like the C4.5 algorithm, showed that this approach provides some clue about the most important feature for case separation. However, the latter yielded higher misclassification rate, than the FFNN, which may be related with the successive nature of node growth. That means that if the first split were suboptimal there is no way of altering its effect.

The aim of the present work is not the comparison of the performance of different modeling techniques but the comparison of different user profile representations. One point for the use of FFNN is their ability to easily find correlations between large numbers of variables. In terms of modeling speed FFNN models work well. One can obtain a comparatively reasonable model more quickly than when one builds a competitive statistical model. FFNN can also cope with non-linear problems and have been used successfully in time-series forecasting [1]. So it is plausible to use them on sequential patterns of a user's behavior, expecting to adapt to changes in it.

The experiments presented here use supervised learning and may have consulting use for the type of user profiles (or user representations) that can better characterize a

user's behavior. The task of detecting fraud in an unsupervised manner is a more difficult one, given the dynamic appearance of new fraud types. The application of an appropriate clustering method, like classic cluster analysis or the more sophisticated self-organizing map (SOM) is considered as the next step to the present work. Moreover, any accurate fraud detection technique is bound to be proprietary to the environment in which it is working.

Acknowledgements

The authors would like to thank the staff of the Telecommunications Center of the Aristotle University for their contribution of data. This work was supported in part by the Research Committee of the Technological Educational Institute of Serres, Greece.

References

1. Gosset, P. and Hyland, M.: Classification, detection and prosecution of fraud in mobile networks. Proceedings of ACTS Mobile Summit, Sorrento Italy (1999)
2. Bolton, R. J. and Hand, D. J.: Statistical fraud detection: a review. Statistical Science, vol. 17, no. 3. (2002) 235–255
3. ACTS ACo95, project ASPeCT: Legal aspects of fraud detection. AC095/KUL/W26/ DS/P/25/2. (1998)
4. Hilas, C. S. and Sahalos, J. N.: User profiling for fraud detection in telecommunications networks. Proc. of the 5th Int. Conf. on Technology and Automation - ICTA '05, Thessaloniki Greece (2005) 382-387
5. Moreau Y., Preneel B., Burge P., Shawe-Taylor J., Stoermann C., Cooke C.: Novel Techniques for Fraud Detection in Mobile Telecommunication Networks. ACTS Mobile Summit. Granada Spain (1997)
6. Buschkes, R., Kesdogan, D., and Reichl, P.: How to increase security in Mobile Networks by Anomaly Detection. Proc. of the 14th Annual Computer Security Applications Conference - ACSAC '98. (1998) 8
7. Rosset, S., Murad, U., Neumann, E., Idan, Y., and Pinkas G.: Discovery of fraud rules for telecommunications – challenges and solutions. Proc. of ACM SIGKDD-99. ACM Press, San Diego, CA, USA (1999) 409-413
8. Adomavicious, G. and Tuzhilin, Al.: User profiling in Personalization Applications through Rule Discovery and Validation. Proc. of ACM SIGKDD-99. ACM Press, San Diego, CA, USA (1999) 377 - 381
9. Fawcett, T. and Provost, F.: Adaptive Fraud Detection. Data Mining and Knowledge Discovery, 1, Kluwer (1997) 291 – 316
10. Oh, S. H. and Lee, W. S.: An anomaly intrusion detection method by clustering normal user behavior. Computers & Security, Vol. 22, No. 7. (2003) 596-612
11. Cox, K. C., Eick, S. G., Wills, G. J. and Brachman R. J.: Visual data mining: Recognizing telephone calling fraud. J. Data Mining and Knowledge Discover, 1 (2). (1997) 225-231
12. Taniguchi, M., Haft, M., Hollmen, J. and Tresp V.: Fraud detection in communication networks using neural and probabilistic methods. Proc. of the 1998 IEEE Int. Conf. in Acoustics, Speech and Signal Processing, Vol. 2. (1998) 1241-1244

13. Manikopoulos, C. and Papavassilliou, S.: Network Intrusion and Fault Detection: A Statistical Anomaly Approach. IEEE Communications Magazine, Vol. 40, 10. (2002) 76 – 82
14. Hollmen, J. and Tresp, V.: Call-based fraud detection in mobile communication networks using a hierarchical regime switching model. Proc. of the 1998 Conf. on Advances in Neural Information Processing Systems II, MIT Press, (1999), 889-895
15. Phua, C., Alahakoon, D. and Lee, V.: Minority Report in Fraud Detection: Classification of Skewed Data. ACM SIGKDD Explorations – Special Issue on Imbalanced Data Sets, vol. 6 (1). (2004) 50 – 58
16. Fawcett, T. and Phua, C.: Fraud Detection Bibliography, Accessed From: http://liinwww.ira.uka.de/bibliography/Ai/fraud.detection.html. (2005)
17. Hinde, S. F.: Call Record Analysis. Making Life Easier - Network Design and Management Tools (Digest No: 1996/217), IEE Colloquium on, (1996) 8/1 – 8/4
18. Manly, B. F. J.: Multivariate Statistical Methods: A Primer. 2nd ed. Chapman & Hall. (1994)
19. Lin, Chin-Teng, and Lee, George C. S.: Neural-Fuzzy Systems: a neuro fuzzy synergism to intelligent systems. Prentice Hall, Upper Saddle River, NJ. (1996)
20. Riedmiller, M., and H. Braun.: A direct adaptive method for faster backpropagation learning: The RPROP algorithm. Proc. of the IEEE Int. Conf. on Neural Networks, Vol.1. (1993) 586-591
21. Downey, T. J., Meyer, D. J., Price R. K., and Spitznagel E. L.: Using the receiver operating characteristic to asses the performance of neural classifiers. Int. Joint Conf. Neural Networks, Vol. 5. (1999) 3642 -3646
22. Provost, F., Fawcett, T. and Kohavi, R.: The case against accuracy estimation for comparing induction algorithms. Proc of the Fifteenth Int. Conf. on Machine Learning. San Francisco, Morgan Kaufmann (1998) 43-48
23. Kajitani, Yoshio, Hipel, Keith W., McLeod Ian A.: Forecasting nonlinear time series with feed-forward neural networks: a case study of Canadian lynx data. Journal of Forecasting, Vol. 24 (2). (2005) 105-117

Predicting User's Movement with a Combination of Self-Organizing Map and Markov Model

Sang-Jun Han and Sung-Bae Cho

Department of Computer Science
Yonsei University
134 Shinchon-dong, Seodaemun-ku, Seoul 120-749, Korea
{sjhan, sbcho}@sclab.yonsei.ac.kr

Abstract. In the development of location-based services, various location-sensing techniques and experimental/commercial services have been used. We propose a novel method of predicting the user's future movements in order to develop advanced location-based services. The user's movement trajectory is modeled using a combination of recurrent self-organizing maps (RSOM) and the Markov model. Future movement is predicted based on past movement trajectories. To verify the proposed method, a GPS dataset was collected on the Yonsei University campus. The results were promising enough to confirm that the application works flexibly even in ambiguous situations.

1 Introduction

Location-based services (LBS) have been a hot topic in the field of wireless and mobile communication devices. One reason for this is because mobile device users want to be able to access information and services specific to their location. As location-sensing and wireless network technologies have developed, various kinds of LBS have emerged. In the field of context-awareness and artificial intelligence, researchers have attempted to develop novel smart location-based applications (see the Related Works section). Prediction of future movement is one key aspect of the next generation of LBS. Current LBS applications attempt to meet the user's present needs. But if the application can also predict where the user will be, it will be able to provide services the user may need in the future, as well as the services they need at present.

In previous research on movement prediction, the method of modeling the transitions between locations was used (Figure 1 (b)). The Markov model was used to represent the transitions between locations and future movement was predicted based on the highest probability transition from the current location [1]. However, this model is inflexible because it takes into account only the current location or place of the user. For example, if the transition from place A to place B has the highest probability and the transition to place C has the second highest probability, the application will always say "you will go to place C" to the user, even if this is untrue. That is to say, the model cannot cope with ambiguous situations.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 884 – 893, 2006.

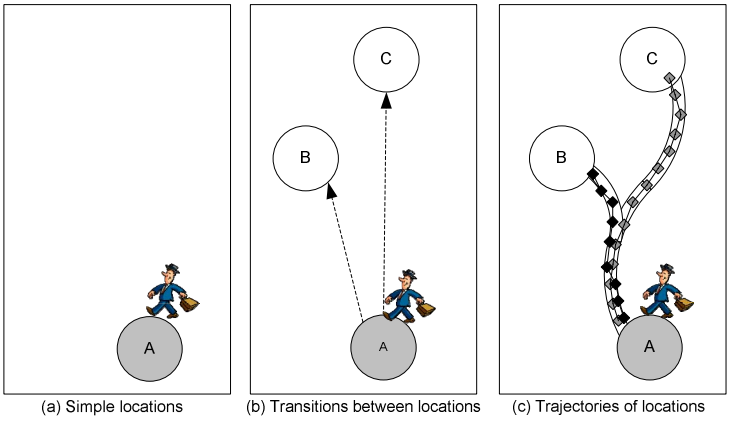


Fig. 1. Comparison of location systems

To achieve more intelligent and flexible predictions, we propose a trajectory-based approach (Figure 1(c)). The main idea is to model the trajectories of locations for movement prediction so that the predictions are based on past trajectories, not the current location. A trajectory-based approach enables the system to distinguish whether the user will head for place B or for place C. It can then react adaptively to the user's destination and future movement can be predicted more flexibly and accurately than when using the transition-based approach. In addition, the system will act differently according to the route taken by the user. For example, different services can be offered to users when they travel along the highway and when they travel along residential roads.

2 Related Works

Many commercial location-based services are already widely used. Wireless service providers offer customer-based plans which assign different rates to calls made from home or from the office [2]. Major credit card companies have created wireless ATM-locator services. AT&T provides `find people nearby' services which allow users to locate friends and family members.

A location-aware event planner designed by Z. Pousman *et al.* integrates a friend finder application which displays locations on a given campus map [3]. The user can organize social events in contextually-enhanced ways. The system also includes privacy management functionality which enables the user to manage visibility to others. Location-based games like `Can You See Me Now' of the University of Nottingham and `Human Pacman' of the National University of Singapore provide novel gaming environments which are enhanced by physical locations [4][5].

Many researchers have attempted to go beyond present-day location systems by extracting high-level information from raw location data. D. Ashbrook *et al.* proposed a method for predicting future movements which used a modification of the k-means

clustering algorithm and the Markov chain model [6]. D.J. Patterson *et al.* proposed a method to be used in the current transportation mode which used a dynamic Bayesian network model [7]. Domain knowledge was incorporated into the Bayesian network model and the parameters of their network were learned using an expectation-maximization (EM) algorithm. In their experimental results, the Bayesian network model outperformed both the decision tree and the Bayesian network model without any domain knowledge. Sto(ry)chastics by F. Sparacino estimated the type of museum visitor for user adaptive storytelling in museums [8]. The visitor's location can be tracked by infrared beacons and a Bayesian network model that estimates the visitor's type as greedy, selective, or busy from the user's location and time spent at each location. Visitors are able to see different explanations about the same exhibits according to their visiting habits.

3 Learning and Predicting Future Movement

Figure 2 shows our movement prediction framework. First, we discover patterns of us er movement by clustering the location dataset (Step 1). This set comprises sequences of GPS records. A sequence represents a movement between places. Then, the model s are built (Step 2). User-preferred services are paired with related movement pattern models. These form user profiles. While the user travels, current movement is compar ed with the movement models (Step 3). Step 3 is repeated whenever the user travels s ome distance. If a movement model is significantly similar to the current movement, t he system will predict that the user will travel along the route of that model (Step 4). User-preferred services related to the selected movement pattern are offered to the use r immediately after the movement prediction.

To develop an easily adaptable system, it is necessary to automatically find what kinds of movements exist in a person's life with minimum pre-knowledge. The self-organized learning approach is suited to this purpose. We employed self-organizing maps to discover significant patterns of user movements from the location dataset.

The benefit of SOM is that it can provide a good approximation of the original input space. A SOM projects the continuous input space to the discrete output space. The output space of a SOM can be viewed as a smaller set of prototypes which store a large set of input vectors. This property helps simplify the problem. The sequences of raw GPS records can be transformed to the sequences of finite units by projecting them onto the SOM output space. We can state the transformed movement data as the state transition sequence of the user's movement. Thus, the user's movement patterns can be modeled more effectively by learning the transition sequences of finite states rather than learning the sequences of the vectors of two floating-point numbers.

A standard SOM is not able to discover significant movement paths because it cannot process temporal sequence data. In order to distinguish different movement patterns, temporal data processing is needed. To cope with this problem, the recurrent SOM (RSOM) is introduced. The RSOM processes the temporal sequence data by maintaining contextual information between the input samples. Even if the GPS data is captured at the same place, the RSOM projects the data into different output units

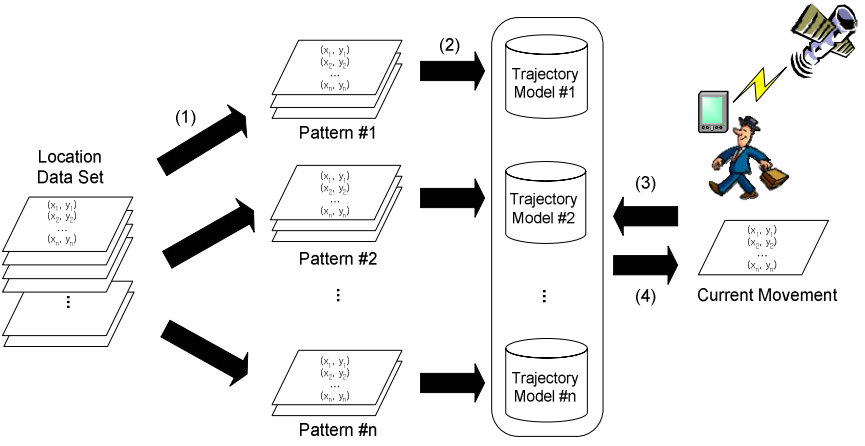


Fig. 2. Movement prediction framework

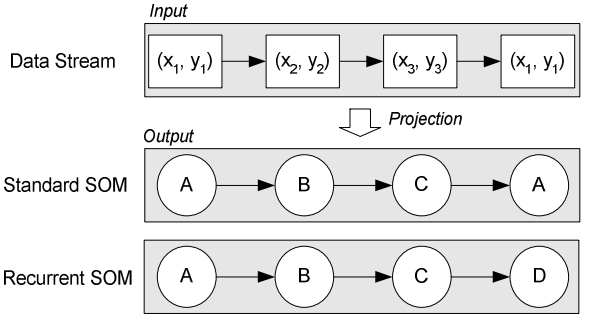


Fig. 3. Differences between the standard SOM and the recurrent SOM

with respect to past movement trajectories. Figure 3 illustrates the difference between the two kinds of SOM. When we project a two-dimensional vector sequence (of which the first and the last vectors are the same) into the two kinds of SOM, they yield different outputs. The standard SOM plots the first and last vectors into an identical output unit without regard to past input. However, in the RSOM, the last vector is mapped into another output unit. Hence, different movement paths and movement trajectories can be distinguished with the last output unit.

3.1 Discovering Patterns of User Movement

The SOM is a representative unsupervised neural network used to solve clustering an d vector quantization problems. The principal goal is to transform an incoming input pattern (of arbitrary dimensions) into a one or two-dimensional discrete map [9]. The output map L consists of a rectangular or hexagonal lattice of $n(i)$ units. The algorith m for training the SOM involves four essential processes: initialization, competition,

cooperation, and adaptation. These are summarized as follows. Here, $b(x)$ is the best matching unit to the input vector, *h* means the neighborhood function and η is the lea rning rate.

1. Initialize the codebook vectors $w_i(0)$.
2. Compute difference and select the best matching unit

$$b(n) = \arg\min \left\| x(n) - w_j(n) \right\| \tag{1}$$

3. Update the codebook vectors

$$w_i(n+1) = w_i(n) + \eta(n) h_{b(n),i}(n)(x(n) - w_i(n)) \tag{2}$$

4. Repeat from 2 to 3 until the stop condition satisfies

Although the RSOM is specialized for temporal sequence processing, it inherits the original properties of the SOM [10]. The differences between the RSOM and the standard SOM are as follows. The RSOM allows the storing of temporal context from consecutive input vectors by putting the leaky integrator into the difference formula of the competition step.

$$y_i(n) = (1-\alpha) y_i(n-1) + \alpha(x(n) - w_i(n)) \tag{3}$$

where α is the leaking coefficient, $y_i(n)$ is the leaked difference vector at step *n* an d $x(n)$ is the input vector at step *n*. The best matching neuron criterion and codebook vector update rules are the same as the standard SOM. The best matching unit (BMU) at time step *n*, $b(n)$ is the unit with the minimum difference.

$$b(n) = \arg\min_i \left\| y_i(n) \right\| \tag{4}$$

The *i*th codebook vector at step *n* , $w_i(n)$ is updated as follows:

$$w_i(n+1) = w_i(n) + \eta(n) h_{b(n),i}(n) y_i(n) \tag{5}$$

The difference vectors are reset to zero after learning each input sequence and the alg orithm is repeated with the next input.

In this problem, the input vector $x(n)$ is a GPS record captured at time *n*, which is a two-dimensional vector composed of the user's specific longitude and latitude. A new GPS record is captured once or twice a second even if the user does not move. The meaningless data in the raw GPS records has to be filtered. Only after the user travels some distance, a new GPS record can be captured and used for training and predicting. Raw GPS data is never 100% accurate. There are many methods to allow for this margin of error. In this research, however, no error correction method was employed and the raw GPS records were directly used to focus on movement prediction.

The procedure of pattern discovery is as follows. First, we train the RSOM with the trajectory dataset obtained from the user. A trajectory dataset

$X_k = \{x(0), x(1), \ldots, x(N)\}$ is a sequence of GPS records captured during a movement *k*. Then, the trajectory dataset is transformed into the sequence of BMU $B_k = \{b(0), b(1), \ldots, b(N)\}$ by projecting it to the trained RSOM to group the similar trajectories. The transformed trajectory dataset is clustered according to the last BMU. A set of the transferred trajectory data that corresponds to the *i*th output unit of the RSOM, $C_i = \{B_1, B_2, \ldots, B_L\}$ represents a discovered pattern of user movement.

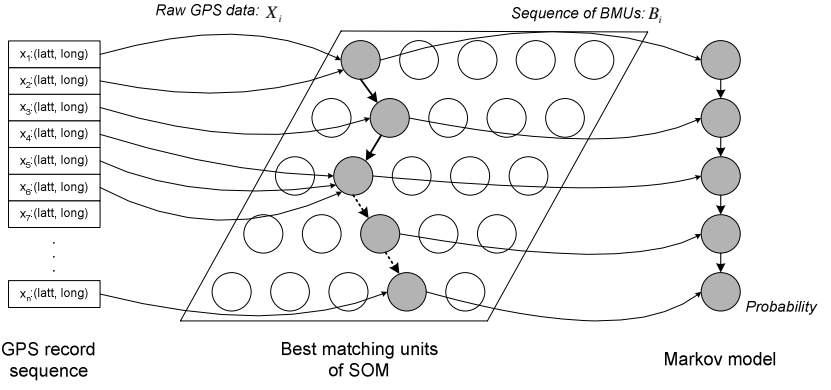


Fig. 4. Combination of RSOM and Markov model

3.2 Building Trajectory Models

A Markov model is a stochastic process based on the Markov assumption, under which the probability of a certain observation only depends on the observation that directly precedes it [11].The trajectory models are built using the first-order Markov models. A Markov model learns the sequences of the best matching units rather than those of the raw GPS data. Changes to the best matching units during the processing sequence can be considered as changes of state because the SOM approximates the input space. A trajectory model M_i is learned with a transformed trajectory dataset C_i. Figure 4 illustrates the combination of the RSOM and Markov model.

3.3 Predicting Future Movements

Figure 5 presents an algorithm which outlines the movement prediction phase. When a new GPS record is captured by the GPS receiver, the traveling distance after the last GPS record is calculated. If this is less than the minimum distance, the record is ignored. If not, it is inputted into the RSOM to get the BMU of the current GPS record. The new sequence of BMU is made by concatenating the newly obtained BMU $b(N)$ and the previously obtained BMUs $b(0), b(1), \ldots, b(N-1)$. Then, the BMU sequence is evaluated with the trajectory models. The state *i* in the Markov model corresponds to the *i*th output unit in the SOM because the sequences of the BMUs are used as

inputs. Hence, the probability that a sequence of the BMUs $B_k = \{b(0), b(1), \ldots, b(N)\}$ will occur from a given trajectory model can be computed using the following equation:

$$P(b(0), b(1), \ldots, b(N) \mid M_i) = q_{b(0)} \prod_{j=2}^{T} P_{b(t-1)b(t)} \tag{6}$$

The higher the probability of the trajectory model, the more likely the user moves similarly to the corresponding trajectory. The simplest way of selecting the most likely movement pattern is applying a threshold to the probability of the local model and selecting the local model whose probability exceeds the predefined threshold. However, this method suffers from a lack of flexibility because the level of probability varies according to the length of the movement. As the user moves, the overall level of probability decreases because the longer the user moves, the more the state transition probability of the Markov model is multiplied. The decision boundary has to vary according to the movement pattern.

```
input: the trained RSOM and the trajectory models
output: future movement pattern
begin
while (end-of-travel is true) do
  record = get-a-new-GPS-record();
  distance = get-traveling-distance(record);
  if (distance is less than minimum-distance)
   continue;
  end if
  bmu = get-the-best-matching-unit (RSOM, record);
  push-back (sequence, bmu);
  for (each trajectory model) do
    model.probability = evaluate-BMU-sequence(sequence, model);
  end for
  for (each trajectory model) do
    model.significance = compute-significance();
  end for
  max-significance = get-maximum-significance()
  if (max-significance exceeds  threshold)
    return model.pattern-number;
  end if
end while
```

Fig. 5. Outline of the movement prediction algorithm

Therefore, the method based on the relative significance of the trajectory model is employed instead of using the probability of the trajectory model directly. We select the outstandingly probable local model. The significance of the trajectory model is computed using the following equation:

$$significance(M_j) = p(B_k \mid M_j) - \sum_{i=1 st. k=I}^{I} \frac{P(B_k \mid M_j)}{I-1} \tag{7}$$

The significance of the trajectory model is defined as the difference between the probability that the current BMU sequence is generated from one model and the mean of the probabilities from the others. If there is the trajectory model with the significance exceeding the predefined threshold, we predict that the user will travel along the corresponding trajectory. In defining the threshold, the trade off between speed and accuracy has to be considered. The lower the threshold, the earlier we can predict the user's movement. If the threshold is set too high, the prediction will be made later with a relatively low risk of false prediction.

Table 1. User's movements in GPS data

Number	Starting Location	Ending Location	Count
1	Main Gate	Engineering Hall I	13
2	Engineering Hall I	College of Liberal Arts II	12
3	College of Liberal Arts II	Auditorium	13
4	Auditorium	College of Social Science	13
5	College of Social Science	Engineering Hall III	13
6	Engineering Hall III	Student Union	12
7	Student Union	Engineering Hall III	13
8	Engineering Hall III	Central Library	13
9	Central Library	College of Liberal Arts I	12
10	College of Liberal Arts I	Main Gate	12

4 Experiments

To test the proposed method, we collected a GPS dataset based on the actual campus life of Yonsei University students. The average student usually moves along 9 buildings for attending a lecture, having lunch, studying and participating in club activities along 10 kinds of paths. Four students walked along these predefined paths, each holding a GPS-enabled handheld computer. Each movement was discriminated by using the loss of the GPS signal and each trajectory was labeled according to its starting location and ending location. 130 trajectory datasets were collected in total, 13 sets for each class. Each trajectory dataset consisted of sequences of two dimensional vectors (longitude and latitude). However, four trajectory datasets were excluded from the experiments due to recording problems in the GPS receivers. Table 1 presents the description of each movement pattern. Due to GPS signal errors, the collected data could differ slightly from the real moving paths. In our experiments, an 8x8 map was used. The initial learning rate was 0.03 and the initial neighborhood radius was 4. The training algorithm was repeated 5000 times.

Prediction performance was evaluated using cross-validation because the size of the dataset was not large. First, the dataset was divided into 13 subsets. 9 subsets contained all kinds of classes and one class was omitted from the 4 subsets. We then chose one subset as the test dataset and the remaining 12 subsets were used for

training. Training and testing were repeated 13 times while changing the test subset. We repeated this cross-validation procedure ten times in order to evaluate the performance accurately. The results of the prediction experiments are given in Table 2 as a confusion matrix.

Table 2. Confusion matrix

		Predicted											
		1	2	3	4	5	6	7	8	9	10	Miss	Accuracy
Actual	1	**41**	0	0	0	0	0	0	0	0	0	89	0.32
	2	10	**110**	0	0	0	0	0	0	0	0	0	0.92
	3	0	0	**116**	0	0	0	0	0	0	0	14	0.89
	4	0	0	0	**130**	0	0	0	0	0	0	0	1.00
	5	0	0	0	0	**129**	0	1	0	0	0	0	0.99
	6	0	0	0	0	0	**88**	0	32	0	0	0	0.73
	7	0	9	0	0	0	0	**121**	0	0	0	0	0.93
	8	0	0	0	0	0	20	0	**110**	0	0	0	0.85
	9	0	0	0	0	4	0	0	0	**116**	0	0	0.97
	10	0	11	0	0	0	0	0	0	0	**99**	0	0.82

The 'Miss' column shows data which was not predicted because the significance did not exceed the threshold until the end of the movement. The prediction accuracy of movement path 1 is the lowest because it shows the most misses. One possible reason for this is because there was not enough time to exceed the threshold because the main gate and engineering hall I are so close to each other. However, besides the misses, no errors in prediction occurred. The lower accuracy of the path 1 reduces the average accuracy (0.84%). However, when the results from path 1 are removed, performance is acceptable. The average accuracy when excluding movement 1 is 0.9. In ambiguous situations (movements 6 and 8), the accuracies are 0.73 and 0.85, respectively. All errors in predicting paths 6 and 8 are due to the confounding of the two paths. The average travel time was 4 minutes 37 seconds and the average time elapsed until prediction was 1 minute 21 seconds. This result indicates that we can predict the user's future movement path before the user arrives at the destination.

5 Conclusion

In this paper, a novel method for learning user's movement patterns and predicting fut ure movements is presented. Our trajectory-based movement prediction method can le ad to more intelligent and proactive location-based services. In the future, we plan to i ncorporate more contexts into the trajectory models. If additional context information such as time of day, transportation mode and current activities are used, needs could b e estimated more accurately. Especially, information about the time of day may be a k ey factor when improving accuracy because many people travel along their regular ro utes during specific time periods.

Acknowledgement

This research was supported by Brain Science and Engineering Research Program sponsored by Korean Ministry of Commerce, Industry and Energy.

References

1. Ashbrook, D. and Starner, T., "Learning Significant Locations and Predicting User Movement with GPS," *Proceedings of IEEE Sixth International Symposium on Wearable Computing,* Seattle, WA. October 2002.
2. Stilp, L., "Carrier and End-User Applications for Wireless Location Systems," *Proceedings of SPIE*, vol. 2602, pp. 119-126, 1996.
3. Pousman, Z., Iachello, G., Fithian, R., Moghazy, J., and Stasko, J., "Design Iterations for a Location-Aware Event Planner," *Personal and Ubiquitous Computing*, vol. 8, no. 2, pp. 117-225, 2004.
4. Benford S., Anastasi R., Flintham M., Drozd A., Crabtree A., Greenhalgh C., Tandavanitj N., Adams, M., Row-Farr J., "Coping with uncertainty in a location-based game," *IEEE Pervasive Computing*, vol. 2, no. 3, pp. 34-41, 2003.
5. Cheok, A.D., Goh, K.H., Liu, W., Farbiz, F., Fong, S.W, Teo, S.L., Li, Y. and Yang, X., "Human Pacman: A Mobile, Wide-area Entertainment System based on Physical, Social, and Ubiquitous Computing," *Personal and Ubiquitous Computing*, vol. 8, no. 2, pp. 71-81, 2004.
6. Ashbrook, D. and Starner, T., "Using GPS to Learn Significant Locations and Predict Movement Across Multiple Users," *Personal and Ubiquitous Computing*, vol. 7, no. 5, pp. 275–286, 2003.
7. Patterson, D., Liao, L., Fox, D., and Kautz, H., "Inferring High-Level Behavior from Low-Level Sensors," *Proceedings of the Fifth International Conference on Ubiquitous Computing*, pp. 73-89, Seattle, WA, October, 2003.
8. Sparacino, F., "Sto(ry)chastics: A Bayesian Network Architecture for User Modeling and Computational Storytelling for Interactive Spaces," *Proceedings of the Fifth International Conference on Ubiquitous Computing*, pp. 54-72, Seattle, WA, October 2003.
9. Haykin, S., *Neural Networks: A Comprehensive Foundation*, Second Ed, Prentice Hall, 1999.
10. Koskela, T., Varsta, M., Heikkonen, J., and Kaski, K., "Temporal Sequence Processing using Recurrent SOM," *Proceedings of Second International Conference on Knowledge-Based Intelligent Engineering Systems*, vol. 1, pp. 290-297, Adelaide, Australia, April 1998.
11. Winston, W.L., *Operations Research: Applications and Algorithms*, Belmont, CA: Duxbury, 1994.

Learning Manifolds in Forensic Data*

Frédéric Ratle[1], Anne-Laure Terrettaz-Zufferey[2], Mikhail Kanevski[1], Pierre Esseiva[2], and Olivier Ribaux[2]

[1] Institut de Géomatique et d'Analyse du Risque, Faculté des Géosciences et de l'Environnement, Université de Lausanne, Amphipôle, CH-1015, Switzerland
frederic.ratle@unil.ch
[2] Institut de Police Scientifique et de Criminologie, Ecole des Sciences Criminelles, Université de Lausanne, Batochime, CH-1015, Switzerland

Abstract. Chemical data related to illicit cocaine seizures is analyzed using linear and nonlinear dimensionality reduction methods. The goal is to find relevant features that could guide the data analysis process in chemical drug profiling, a recent field in the crime mapping community. The data has been collected using gas chromatography analysis. Several methods are tested: PCA, kernel PCA, isomap, spatio-temporal isomap and locally linear embedding. ST-isomap is used to detect a potential time-dependent nonlinear manifold, the data being sequential. Results show that the presence of a simple nonlinear manifold in the data is very likely and that this manifold cannot be detected by a linear PCA. The presence of temporal regularities is also observed with ST-isomap. Kernel PCA and isomap perform better than the other methods, and kernel PCA is more robust than isomap when introducing random perturbations in the dataset.

1 Introduction

Chemical profiling of illicit drugs has become an important field in crime mapping in recent years. While traditional crime mapping research has focused on criminal events, i.e., the analysis of spatial and temporal events with traditional statistical methods, the analysis of the chemical composition of drug samples can reveal important information related to the evolution and the dynamics of illicit drugs market.

As described in [2], many types of substances can be found in a cocaine sample seized from a street dealer. Among those, there are of course the main constituants of the drug itself, but also chemical residues of the fabrication process and cutting agents used to dilute the final product. Each of these can possibly provide information about a certain stage of drug processing, from the growth conditions of the original plant to the street distribution. This study will focus on cocaine main constituants, which are enumerated in section 3.

* This work was supported by the Swiss National Science Foundation (grant no.105211-107862).

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 894–903, 2006.

2 Related Work

A preliminary study was made by the same authors in [1], where heroin data was used. PCA, clustering and classification algorithms (MLP, PNN, RBF networks and k-nearest neighbors) were successfully applied. However, heroin data has less variables (6 main constituants), which makes it more likely to be reduced to few features.

A thorough review of the field of chemical drug profiling can be found in Guéniat and Esseiva [2]. In this book, authors have tested several statistical methods for heroin and cocaine profiling. Among other methods, they have mainly used similarity measures between samples to determine the main data classes. A methodology based on the square cosine function as an intercorrelation measurement is explained in further details in Esseiva et al. [3].

Also, principal component analysis (PCA) and soft independent modelling of class analogies (SIMCA) have been applied for dimensionality reduction and supervised classification. A radial basis function network has been trained on the processed data and showed encouraging results. The classes used for classification were based solely on indices of chemical similarities found between data points. This methodology was further developed by the same authors in [4].

Another type of data was studied by Madden and Ryder [5]: Raman spectroscopy obtained from solid mixtures containing cocaine. The goal was to predict, based on the Raman spectrum, the cocaine concentration in a solid using k-nearest neighbors, neural networks and partial least squares. They have also used a genetic algorithm to perform feature selection. However, their study has been constrained by a very limited number of experimental samples, even though results were good. Also, the experimental method of sample analysis is fundamentally different from the one used in this study (gas chromatography). Similarly, Raman spectroscopy data was studied in [6] using support vector machines with RBF and polynomial kernels, KNN, the C4.5 decision tree and a naive Bayes classifier. The goal of the classification algorithm was to discriminate samples containing acetaminophen (used as a cutting agent) from those that do not. The RBF-kernel SVM outperformed all the other algorithms on a dataset of 217 samples using 22-fold cross-validation.

3 The Data

The data has 13 initial features, i.e., the 13 main chemical components of cocaine, measured by peaks area on the spectrum obtained for each sample:

1. Cocaine
2. Tropacocaine
3. Benzoic acid
4. Norcocaine
5. Ecgonine

6. Ecgonine methyl ester
7. N-formylcocaine
8. Trans-cinnamic acid
9. Anhydroecgonine
10. Anhydroecgonine methyl ester
11. Benzoylecgonine
12. Cis-cinnamoylecgonine methyl ester
13. Trans-cinnamoylecgonine methyl ester.

Time is also implicitely considered in ST-isomap.

Five dimensionality reduction algorithms are used: a standard principal component analysis, kernel PCA [7], locally linear embedding (LLE) [8], isomap [9] and spatio-temporal isomap [10]. The latter has been used in order to detect any relationship in the temporal evolution of the drug's chemical composition, given that the analyses have been sequentially ordered with respect to the date of seizure for that experiment.

Every sample has been normalized by dividing each variable by the total area of the peaks of the chromatogram for that sample, every peak being associated with one chemical substance. This normalization is common practice in the field of chemometrics and aims at accounting for the variation in the purity of samples, i.e., the concentration of pure cocaine in the sample.

9500 samples were considered. It is worth noting that a dataset of this size is rather unusual due to the restricted availability of this type of data.

4 Methodology and Results

Due to the size of the dataset (9500 samples, 13 variables), the methods involving the computation of a Gram matrix or distance matrix were repeated several times with random subsets of the data of 50% of its initial size.

All the experiments were done in Matlab. The kernel PCA implementation was taken from the pattern classification toolbox by Stork and Yom-Tov [11], which implements algorithms described in Duda et al. [12]. LLE, isomap and ST-isomap implementations were provided by the respective authors of the algorithms.

4.1 Principal Component Analysis

Following normalization and centering of the data, a simple PCA was performed. The eigenvalues seem to increase linearly in absolute value, and a subset of at least six variables is necessary in order to explain 80% of the data variability. Fig. 1 shows the residual variance vs the number of components in the subset.

Given that the data can be reduced at most to 6 or 7 components, the results obtained with PCA are not convincing and suggest the use of methods for detecting nonlinear structures, i.e., no simple linear strucure seem to lie in the high-dimensional space. As an indication, the two first principal components are illustrated in Fig. 2.

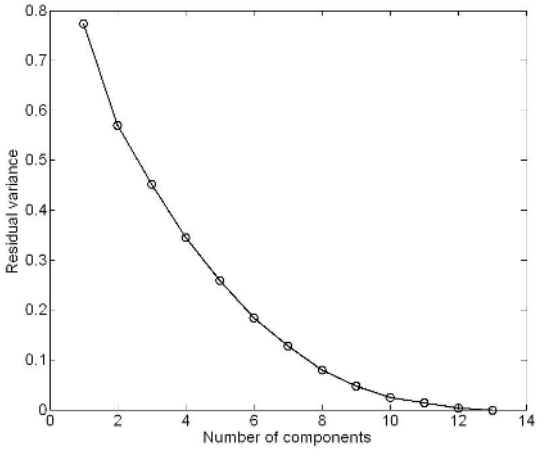


Fig. 1. Residual variance vs number of components

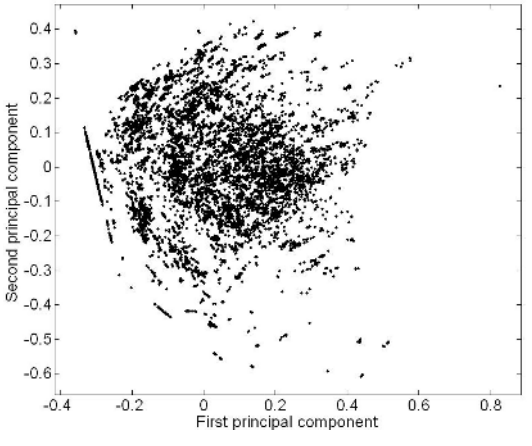


Fig. 2. The two main principal components

4.2 Kernel PCA

Kernel PCA was introduced by Schölkopf et al. [7] and aims at performing a PCA in feature space, where the nonlinear manifold is linear, using the kernel trick. KPCA is thus a simple yet very powerful technique to learn nonlinear structures. Using the Gram matrix K, defined by a positive semidefinite kernel (usually linear, polynomial or Gaussian), rather than the empirical covariance matrix and knowing that, as for PCA, the new variables can be expressed as the product of eigenvectors of the covariance matrix and the data, the nonlinear projection can be expressed as:

$$\left(\mathbf{V}^k \cdot \mathbf{\Phi}\left(\mathbf{x}\right)\right) = \sum_{i=1}^{N} \alpha_i^k K\left(\mathbf{x_i}, \mathbf{x}\right) \tag{1}$$

where N is the number of data points, α_i^k is the eigenvector of the Gram matrix corresponding to the eigenvector $\mathbf{V}^k$ of the covariance matrix in feature space, which does not need to be computed. The radial basis function kernel provided the best results (among linear, polynomial and Gaussian), using a Gaussian width of 0.1. Fig. 3 shows the two-dimensional manifold obtained with KPCA.

Unlike PCA, a coherent structure is recognizable here, and it seems that two nonlinear features reasonably account for the variation in the whole dataset.

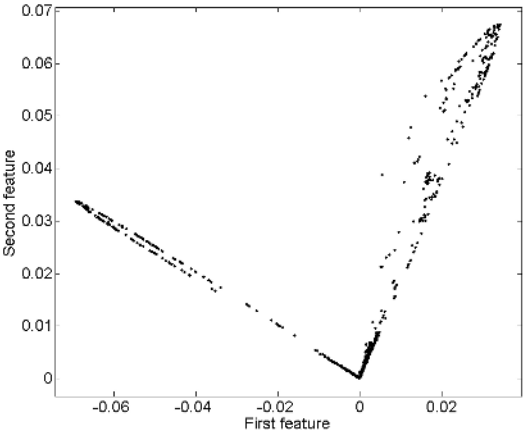


Fig. 3. Two-dimensional embedding with kernel PCA

4.3 Locally Linear Embedding

LLE [8] aims at constructing a low-dimensional manifold by building local linear models in the data. Each point is embedded in the lower-dimensional coordinate system by a linear combination of its neighbors:

$$\hat{X}_i = \sum_{i \in N_k(X_i)} W_i X_i \tag{2}$$

where $N_k\left(X_i\right)$ is the neighborhood of the point X_i, of size k. The quality of the resulting projection is measured by the squared difference between the original point and its projection. The main parameter to tune is the number k of neighbors used for the projection. Values from 3 to 50 have been tested, and the setting $k = 40$ has provided the best resulting manifold, even though this neighborhood value is unusually large. Fig. 4 show the three-dimensional embedding obtained. As for KPCA, a structure can be recognized. However, it is not as distinct, and suggests that LLE cannot easily represent the underlying manifold compared to KPCA.

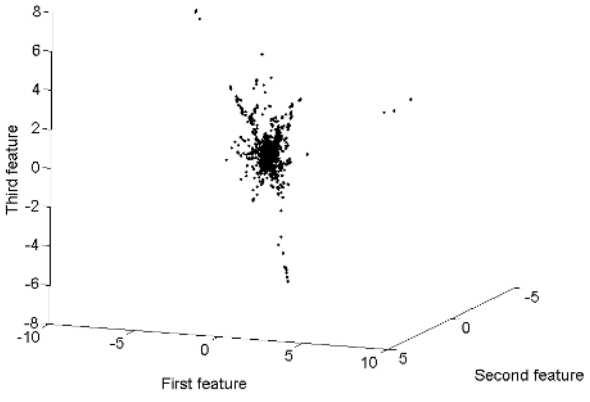


Fig. 4. Three-dimensional embedding with LLE

4.4 Isomap

Isomap [9] is a global method for reduction of dimensionality. It uses the classical linear method of multi-dimensional scaling (MDS) [13], but with *geodesic* distances rather than Euclidean distances. The geodesic distance between two points is the shortest path along the manifold. Indeed, the Euclidean distance does not appropriately estimate the distance between two points lying on a non-linear manifold. However, it is usually locally accurate, i.e., between neighboring points. Isomap can therefore be summarized as:

1. Determination of every point's nearest neighbors (using Euclidean distances);
2. Construction of a graph connecting every point to its nearest neighbours;

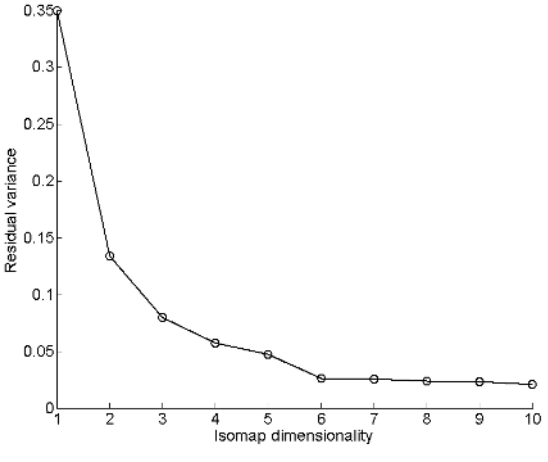


Fig. 5. Residual variance vs Isomap dimensionality

3. Calculation of the shortest path on the graph between every pair of points;
4. Application of multi-dimensional scaling on the resulting distances (geodesic distances).

The application of this algorithm on the chemical variables has also provided good results compared to PCA. As for LLE, the number of neighbors k has been studied and set to 5. As an indication, Fig. 5 shows the residual variance with subsets of 1 to 10 components. As it can be seen, the residual variance with only one component is much lower than for PCA. In Fig. 6, the two-dimensional embedding is illustrated. From this figure, it appears that the underlying structure is better caught than with LLE, which may suggest that isomap is more efficient on this dataset.

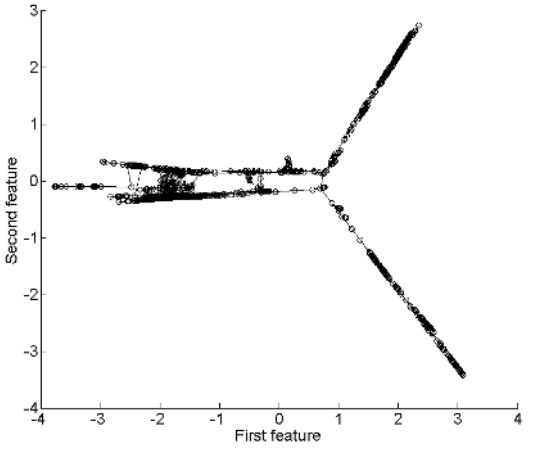


Fig. 6. Two-dimensional embedding with ISOMAP

4.5 Spatio-temporal Isomap

It is well-known in the crime research community that time series analysis often leads to patterns that reflect police activity rather than underlying criminal behavior. This is especially true in drug profiling research, where the police seizures can vary in time independently of criminal activity. On the other hand, for data such as burglaries, time series analysis could prove more efficient, since the vast majority of events are actually reported. Methods assuming *sequential* data rather than time-referenced data are perhaps more promising in the field of drug profiling in order to capture true underlying patterns rather than sampling patterns .

Spatio-temporal isomap [10] is an extension of isomap for the analysis of sequential data and has been presented by Jenkins and Matarić. Here, the data is of course feature-temporal rather than spatio-temporal. The number of neighbors and the obtained embedding are the same as with isomap. However, the feature-temporal distance matrix is shown in Fig. 7. From this figure, it can be seen that

regularities are present in the dataset. Given that the samples cover a period of several years, this data could be used in a predictive point of view and could help understand the organization of distribution networks. This remains the purpose of future study.

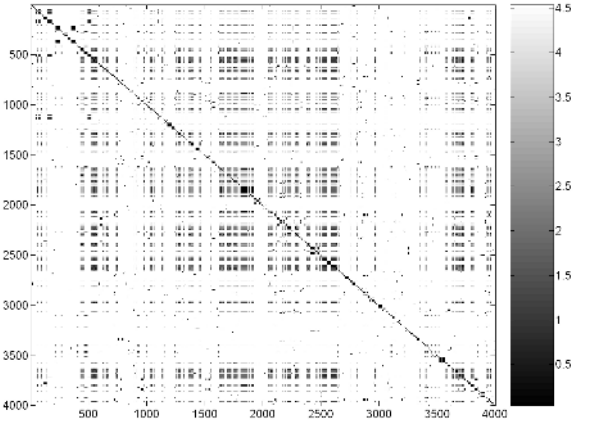


Fig. 7. Feature-temporal distance matrix

4.6 Robustness Assessment

Following these results, the robustness of the two most well-suited methods (KPCA and isomap) was tested using a method similar to that used in [14]. Indeed, few quantitative criteria exist to assess the quality of dimensionality reduction methods, since the reconstruction of patterns in input space is not straightforward and thus limits our ability to measure the accuracy of a given algorithm. The algorithm that has been used follows this outline:

1. Randomly divide the dataset D in three partitions: F, P_1 and P_2.
2. Construct embeddings using $F \cup P_1$ and $F \cup P_2$.
3. Compute the mean squared difference (MSD) between both embeddings obtained for F.
4. Repeat the previous steps for a fixed number of iterations.

The embeddings were constructed 15 times for kernel PCA and isomap, and the results are summarized in Table 1.

Table 1. Normalized mean squared difference for KPCA and isomap

Algorithm	MSD	$std\,(MSD)$
Kernel PCA	0.077	0.001
Isomap	0.174	0.004

It can be observed that here, kernel PCA is considerably more stable than isomap. Isomap, being based on a graph of nearest neighbors, may be more sensitive to random variations in the dataset and could therefore lead to different results with different sets of observations of a given phenomenon.

5 Conclusion

Five methods of dimensionality reduction were applied to the problem of chemical profiling of cocaine. The application of PCA showed that linear methods for feature extraction had serious limits in this field of application. Kernel PCA, isomap, locally linear embedding and ST-isomap have demonstrated the presence of simple nonlinear structures that were not detected by conventional PCA.

Kernel PCA and isomap have given the best results in terms of an interpretable set of features. However, kernel PCA has shown more robust than isomap. Of course, research by experts in drug profiling will yet have to confirm the relevancy of the obtained results and provide a practical interpretation.

Further research will aim at selecting appropriate methods for determination of classes on those low-dimensional structures. This clustering task will enable researchers in the field of crime sciences to determine if distinct production or distribution networks can be put into light by analyzing the data clusters obtained from the chemical composition of the drug seizures. Also, regarding sequential data, other methods could be tested, particularly hidden Markov models.

References

1. F. Ratle, A.L. Terrettaz, M. Kanevski, P. Esseiva, O. Ribaux, Pattern analysis in illicit heroin seizures: a novel application of machine learning algorithms, *Proc. of the 14^{th} European Symposium on Artificial Neural Networks*, d-side publi., 2006.
2. O. Guéniat, P. Esseiva, *Le Profilage de l'Héroïne et de la Cocaïne*, Presses polytechniques et universitaires romandes, Lausanne, 2005.
3. P. Esseiva, L. Dujourdy, F. Anglada, F. Taroni, P. Margot, A methodology for illicit drug intelligence perspective using large databases, *Forensic Science International*, **132**:139-152, 2003.
4. P. Esseiva, F. Anglada, L. Dujourdy, F. Taroni, P. Margot, E. Du Pasquier, M. Dawson, C. Roux, P. Doble, Chemical profiling and classification of illicit heroin by principal component analysis, calculation of inter sample correlation and artificial neural networks, *Talanta*, **67**:360-367, 2005.
5. M.G. Madden, A.G. Ryder, Machine Learning Methods for Quantitative Analysis of Raman Spectroscopy Data, In Proceedings of the *International Society for Optical Engineering* (SPIE 2002), **4876**:1130-1139, 2002.
6. M.L. O'Connell, T. Howley, A.G. Ryder, M.G. Madden, Classification of a target analyte in solid mixtures using principal component analysis, support vector machines, and Raman spectroscopy, In Proceedings of the *International Society for Optical Engineering* (SPIE 2005), **4876**:340-350, 2005.
7. B. Schölkopf, A. Smola, K.R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation* **10**:1299-1319, 1998.

8. S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science* **290**:2323-2326, 2000.
9. J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science* **290**:2319-2323, 2000.
10. O.C. Jenkins, M.J. Matarić, A spatio-temporal extension to isomap nonlinear dimension reduction, *Proc. of the 21st International Conference on Machine Learning*, 2004.
11. D.G. Stork, E. Yom-Tov, *Computer Manual in MATLAB to accompany Pattern Classification*, Wiley, Hoboken (NJ), 2004.
12. R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification, 2nd Edition*, Wiley, New York, 2001.
13. J.B. Kruskal and M. Wish, *Multidimensional Scaling*, SAGE Publications, 1978.
14. Y. Bengio, J.F. Paiement, P. Vincent, O. Delalleau, N. Le Roux and M. Ouimet. Out-of-samples extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. *Advances in Neural Information Processing Systems 16*, 2004.

A Comparison of Target Customers in Asian Online Game Markets: Marketing Applications of a Two-Level SOM

Sang-Chul Lee[1], Jae-Young Moon[2], and Yung-Ho Suh[2]

[1] Department of Management Information Systems, Korea Christian University, San 204, Hwagok 6-Dong, Kangseo-Ku, Seoul 157- 722, South Korea
leecho@kcu.ac.kr
[2] School of Business Administration, Kyung Hee University, 1 Hoegi-Dong, Dongdaemoon-Gu, Seoul 130-701, South Korea
{moonlight, suhy}@khu.ac.kr

Abstract. The purpose of our research is to identify the critical variables, to implement a new methodology for Asian online game market segmentation, and to compare target customers in Asian online game markets; Korea, Japan and China. Conclusively, the critical segmentation variables and the characteristics of target customers were different among countries. Therefore, online game companies should develop diverse marketing strategies based on characteristics of their target customers.

1 Introduction

Recently, Asian online game industry has been grown rapidly. 2005 e-Business White Paper of Korea Institute for Electronic Commerce (KIEC) [17] indicated that the online game market was increased 155.36% from $ 5.6 billion in 2002 to $ 14.3 billion in 2007, compared with 61.2% for global game market. Additionally, Asian online game market held 22.68% ($ 2.2 billion) of market ratio in 2005, compared with 14.3 % ($ 0.8 billion) in 2002. Especially, Korea, Japan and China held over 90% in Asian online game market [17].

With the rapid growth and the higher competitive power, the importance of this industry has been realized not only in the world cultural business but as a profitable business model. Therefore, many online game companies hoped that the first mover would be successful and recklessly entered into online game markets without understanding the core needs of those audiences. However, the lack of consideration has forced many online game companies to fail to survive in game market [16]. To survive in today's competitive markets, online game companies need to understand their loyal customers and concentrate their limited resources into them [21].

However, previous research had problems of international application, methodologies and variables. Firstly, the problem of international application was how well the results performed within a country apply to the other nations. The results

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 904 – 912, 2006.

of previous research were difficult to be generalized into other countries in that the market situation and customers' characteristics in each country are different respectively [9]. Therefore, a comparison analysis of target customer was important to understand their domestic and foreign customers.

Secondly, the traditional methodology for market segmentation was based mainly on statistical clustering techniques; hierarchical and partitive approaches. However, hierarchical method can not provide a unique clustering because a partitioning to cut the dendrogram at certain level is not precise. This method ignores the fact that the within-cluster distance may be different for different clusters [6], [23]. Partitive method predefines the number of clusters, before performing it. It can be part of the error function and can not identity the precise number of clusters [7], [19], [23]. Additionally, these algorithms are known to be sensitive to noise and outliers [4], [5], [23].

To settle these problems, we segment Korean online game market using a two-level Self-Organizing Map (SOM): SOM training and clustering [23]. Instead of clustering the data directly, a large set of prototypes is formed using the SOM. The prototypes can be interpreted as proto-cluster, which are combined in the next phase from the actual clusters. The benefit of using this method is to effectively reduce the complexity of the reconstruction task and to reduce the noise. Our research implements this method into marketing research field.

The purpose of our research is to identify the critical variables, to implement a new methodology for Asian online game market segmentation, and to compare target customers in Asian online game markets. To implement our methodology, Korean, Japanese and Chinese online game data were analyzed because they were located in the center of those trends. Therefore, our research will be helpful for other countries to understand the change of Asian and global game markets.

2 Theoretical Background

2.1 Determinant Variables for Market Segmentation

The convenience of the operator was defined as the manipulatability of operators to play games [22]. Operator is an important determinant of influencing interaction between users and games [2], [12], [24]. Feedback is the reaction from online games [3], [10]. For example, when players kill a monster within NCsoft's Lineage, they receive feedback upgrading their level. The reality of design is defined as the design of interface making gamers feel online games as part of the real world [1], [20], [25]. Information is the contents from online game to achieve the stated goals. Gamers who received more precise information about how to play the games tended to achieve online game goals and experience flow easier [10], [18]. Virtual community is defined as computer-mediated spaces with potential for integration of member-generated content and communication [14]. Online game users should solve problems together interacting with other users in virtual communities [10].

2.2 A Two-Level SOM

Vesanto and Alhoniemi [23] proposed a two-level SOM: SOM training and clustering. A two-level SOM was combined SOM, K-means and DB Index. In the first level (SOM training), the data were clustered directly in original SOM to form a large set of prototypes. In the second level (SOM clustering), the prototypes of SOM are clustered using k-means and the validity of clusters is evaluated using DB index.

To select the best one among different partitioning, a two-level SOM used DB index. Generally, there are several validity indices of clustering methodology; DB (Davies-Bouldin) index [11], Dunn's index [13], CH (Calinski-Harabasz) index [8], index I [19] and so forth. DB index was suitable for evaluation of k-means partitioning because it gives low values, indicating good clustering results for k-means cluster [21].

DB index is a function of the ratio of the sum of within-cluster scatter to between-cluster separation. The DB index is defined as equation (1).

$$DB(U) = \frac{1}{c}\sum_{i=1}^{c} \max_{i \neq j} \left\{ \frac{\Delta(X_i) + \Delta(X_j)}{\delta(X_i, X_j)} \right\} \tag{1}$$

The scatter within the ith cluster, ΔX_i, is computed as equation (2), the distance between cluster X_i,and X_j, is denoted as equation (3). Here, Z_i represents the ith cluster centre.

$$\Delta X_i = \frac{1}{|C_i|}\sum_{x \in C_i} \{\| x - z_i \|\} \tag{2}$$

$$\delta(X_i, X_j) = \| z_i - z_j \| \tag{3}$$

Conclusively, the proper clustering is achieved by minimizing the DB index.

$$DB \downarrow = Validity \uparrow = \frac{Intracluster \downarrow}{Intercluster \uparrow} \tag{4}$$

3 Research Methods

3.1 Research Framework

To segment the online game market and develop marketing strategies, our research approach is categorized into two phases. Firstly, we perform the confirmatory factor analysis (CFA) and structural equation model (SEM) for Korean, Japanese, and Chinese samples to identify the critical segmentation variables for clustering. Secondly, a two-level SOM is used to segment online game market. After segmentation of the markets, we use ANOVA and cross tabulation analysis to recognize the characteristics of sub-divided clusters. Finally, we target a segment market with the highest customer loyalty and compare the target customers of Korea, Japan, and China.

3.2 Data and Measurement

To test the model, a convenience sample of 1704 (KR), 602 (JP), and 592 (CN) online game users were available for analysis, after elimination of missing data. Our research developed multi-item measures for each construct. Twenty-one items for five determinants are selected. We asked respondents to indicate on a five (KR/JP) and seven (CH) point Likert scale to what extent the determinants influence on flow in online game. We used CFA to evaluate convergent validity for five constructs. The results indicated that 15 items for five determinants remained within Korean and Japanese model. However, 11 items for four determinants remained for Chinese model, eliminating the suitability of feedback because it was not significant. All the fit statistics of the measurement model were acceptable.

4 Results

4.1 Identification of Critical Factors

To find the critical factors for segmentation, we used AMOS 4.0 in structural equation modeling (SEM). The structural model of Korea, Japan and China was well converged. The results indicated that the chi-square of the model was 201.01(KR)/ 215.55(JP)/ 146.11(CN) with d.f. of 104(KR)/ 104(JP)/ 55(CN), the ratio of chi-square to d.f. was 2.702(KR)/ 2.073(JP)/ 2.656(CN), GFI was 0.981(KR)/ 0.959(JP)/ 0.965 (CN), AGFI was 0.972(KR)/ 0.940(JP)/ 0.942(CN) and RMSR was 0.019(KR)/ 0.040(JP)/ 0.074(CN); all the fit statistics were acceptable.

Table 1. The results of Stuctural Equeation Model

	Path		Korea	Japan	China
O	-->		0.030	0.273**	0.072*
FB	-->		0.116**	0.058	-
IF	-->	F	0.079*	0.136*	-0.312**
D	-->		0.283**	0.160**	0.236**
C	-->		0.417**	0.385**	0.001

* p<0.05, ** p<0.01

O: The convenience of operator, FB: The suitability of feedback
IF: The precision of information, D: Reality of Design
C: The involvement of virtual community, F: Flow,

The results of SEM indicated that the significant variables for market segmentation were different among each nation. For Korean market, four of the five paths were statistically significant and the path from the convenience of operator to flow was insignificant, as shown in Table 1. For Japanese market, four of the five paths were statistically significant and the path from the suitability of feedback to flow was insignificant. For Chinese market, three of the four paths were statistically significant and the path from the involvement of virtual community to flow was insignificant. Especially, the precision of information influenced negatively.

4.2 Market Segmentation

To segment the Korean online game market, our research was conducted using a two-level SOM. In the experiments, the first level was SOM training. 1704 (KR)/ 602(JP)/ 592(CN) data samples and 4(KR)/ 4(JP)/ 3(CN) significant variables were used. A SOM was trained using the sequential training algorithm for data samples. A neighborhood width decreased linearly 5 to 1 using the Gaussian function. A map was used by 19*11 (KR)/ 15*9 (JP)/ 13*9 (CN)/ matrix and 209(KR)/ 120(JP)/ 117(CN) prototypes were developed.

The second level was SOM clustering. The partitive clustering of 209(KR)/ 120(JP)/ 117(CN) SOM's prototypes was carried out using batch K-means algorithm. The K-means ran multiple times for each k. The DB index was used to select the best clustering. The analysis of the DB index resulted in the development of ten (KR)/ nine (JP)/ six (CN) market segments in Fig. 1. The results were visualized in Fig. 2.

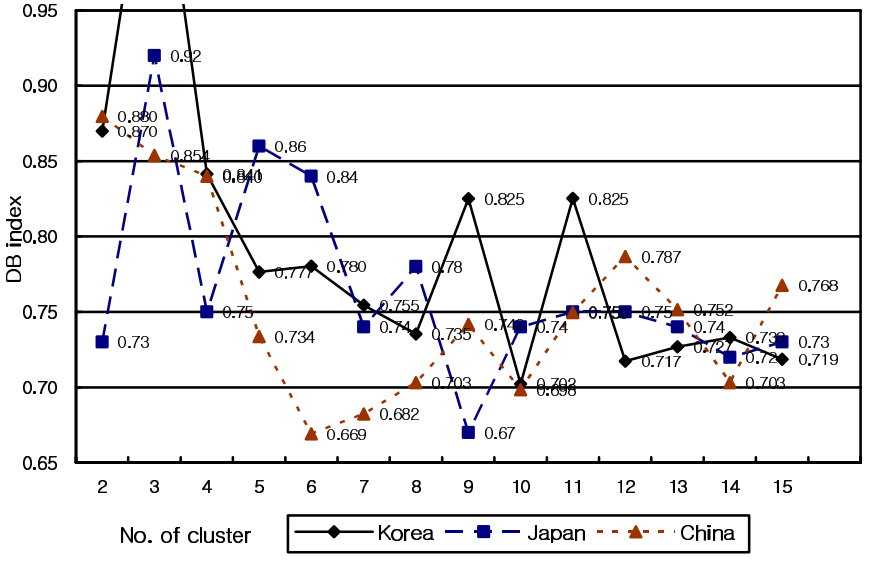


Fig. 1. DB Index

4.3 Determination of Target Market

After segmenting the markets, we used ANOVA to recognize the variable characteristics of each cluster. According to results of ANOVA, all variables (components) were significant; F=91.259(KR)/ 9.115(JP)/ 55.163(CN) to 461.598(KR)/ 55.574(JP)/ 553.220(CN) and p=0.00. To precisely recognize the variable characteristics of clusters, we categorized the effectiveness of the variables into 3 levels; high, middle and low. The middle level ranged between 3±2.5 because our research measurement was used on a five point Likert scale. The high score suggested that the cluster was influenced by the variables positively, the middle score was normal, the low score was negative.

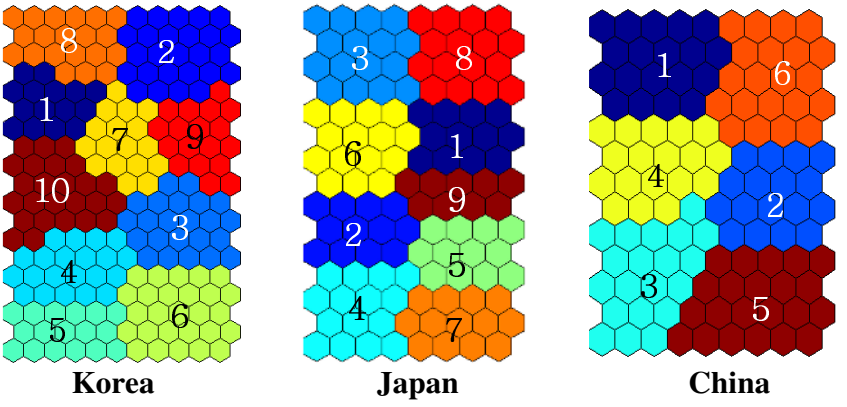


Fig. 2. Visualization of clusters

Table 2. Profiles of clusters

	Target Market		
	Korea	Japan	China
Cluster	C6 (n=224)	C7 (n=103)	C6 (n=168)
O	-	3.76 (High)	3.85 (High)
FB	2.72 (Low)	-	-
IF	3.63 (High)	3.19 (Middle)	2.12 (Low)
D	3.80 (High	3.75 (High)	3.69 (High)
C	4.00 (High)	3.57 (High)	-
Gender	female	male	male
Age	26-30	36-	-18
Job	Student	Employee	Student
School	University Graduate	High School Graduate	High School
Income ($)	501-1,000	-1,000	- 50
i_year	2-4	1-3	2-3
i_day	5, 10	2	0-1
G_day	2	4-	0-1
Revisit	4.02	4.15	3.85
WOM	4.02	3.65	3.77
Loyalty**	4.02	3.90	3.82

* L=Low, M=Middle, H=High
** Loyalty is estimated by average of revisit and WOM
O: The convenience of operator, FB: The suitability of feedback
IF: The precision of information, D: Reality of Design
C: The involvement of virtual community,

Additionally, to identify the structure of the clusters, we conducted on the analysis of the demographic and behavioral variables using cross tabulation analysis: gender, age, job, school, income level, i_year (how long did gamers use the Internet), i_day

(how many hours did gamer use the Internet per day), and g_day (how many hours did gamer play online games per day). The characteristics and structure of clusters are summarized in Table 2.

The analysis of customer loyalty indicated that the target market was cluster 6 (KR)/ cluster 7 (JP)/ cluster 6(CN) among their clusters in Table 2. The other analysis of the intention of revisit and WOM (Word of Mouth) indicated the same results.

5 Conclusion and Limitation

Our research was performed to identify the critical variables, to segment Asian online game market using a two-level SOM, and to compare their target customers. The results indicated that the critical segmentation variables and the characteristics of loyal customers were different among each nation.

Firstly, the results of SEM indicated that the convenience of operator was not significant in Korean model, the suitability of feedback was not significant in Japanese model and the precision of information had negative influence in Chinese model. In comparison with Korea and Japan, the convenience of operator was not significant but the suitability of feedback was significant in Korean model, on the contrary to Japanese model. It was interpreted that Korean online gamers preferred to be reacted appropriately and faster when they completed their missions, because they wanted to achieve a high status in virtual community. Conversely, the Japanese gamers preferred to grow their characters at their convenience. This hypothesis was proven in the case of "Vandai's damakuchi", which is the game growing animal characters for a long time [16].

Additionally, more than 70% of Chinese online games were foreign. However, most information provided by these online games was mistranslated and incorrect so that gamer had to obtain precise information from other channels, such as online game magazine and community sites [15]. Therefore, Chinese gamers did not like incorrect information provided when they played online game and showed negative attitude to the provision of information.

Secondly, the results of characteristics indicated that companies should develop strategies depending on the effectiveness of the variables and the demographic and behavioral characteristics of target market. The main variable was the involvement of virtual community (KR)/ the convenience of operator (JP/CN). Therefore, online game companies should develop strategies depending on the effectiveness of the variables within each cluster. The strategies for virtual community proposed that companies need to provide the different villages and guilds which were harmonized with customer needs. For example, 'Lineage' provided 15 villages to satisfy the different gamers' needs. The strategies for the convenience of operator proposed that companies should provide the diverse characters (Avatar) and items, which are harmonized with customers needs and were manipulated conveniently. For example, 'Lineage' provided knight, wizard, elf, dark elf for male and female and prince/princess, total 10 Avatars and 1,150 items to play games.

As to the demographic information, gender of the Korean primary target market was female while that of the Japanese and Chinese markets was male. An age of the Korean markets was 26-30 years while the Japanese market was over 36 years and

Chinese market was under 18. A job of the Korean and Chinese markets were student while the Japanese market was employee. A school career of the Korean markets was university graduate while the Japanese market was high school graduate and Chinese market was high school student. An income level of the Korean markets was $ 501-1000 while the Japanese market was under $ 1,000 and Chinese market was under $ 50. An income level of the Korean markets was $ 501-1000 while the Japanese market was under $ 1,000 and Chinese market was under $ 50. The target customers of the Korean and Chinese markets used the Internet more than Japanese customers. Time of Internet usage was 5 or 10 hours per day in Korea, compared with 2 hours in Japan and 0-1 hour in China. Time of playing game was 2 hours per day in Korea, compared with over 4 hours in Japan and 0-1 hour in China.

However, results of our research might not be generalized and directly applicable to other countries because our research was conducted only on Korean, Japanese and Chinese online game market. Countries with different cultural and industrial background might have to be very careful about developing their own marketing strategies using our methods due to the difference in gaming population and perception of people toward games.

References

1. Ackley, J.: Roundtable Reports: Better Sound Design. Gamasutra. (1998). http://www. Gamasutra.com /features/gdc_reports/cgdc_98/ ackley.htm
2. Agarwal, R., Karahanna, E.: Time Files When You're Having Fun: Cognitive Absorption and Beliefs About Information Technology Usage. MIS Quarterly, Vol.24, No.4. (2000) 665-694
3. Baron, J.: Glory and Shame: Powerful Psychology in Multiplayer Online Games. Gamasutra. (1999). http: //www.gamasutra.com/features/19991110/Baron_01.htm
4. Bezdek, J. C.: Some New Indexes of Cluster Validity. IEEE Transactions on Systems, Man, and Cybernetics, Part B, Vol.28, No.3. (1998) 301-315
5. Blatt, M., Wiseman, S., Domany, E.: Super-paramagnetic Clustering of Data. Physical Review Letters, Vol.76, No.8. (1996) 3251-3254
6. Boudaillier, E., Hebrail, G.: Interactive Interpretation of Hierarchical Clustering. Intelligent Data Analysis, Vol.2, No.3. (1998) 41-
7. Buhmann, J., KÜhnel, H.: Complexity Optimized Data Clustering by Competitive Neural Networks. Neural Computation, Vol.5, No.3. (1993) 75-88
8. Calinski, R. B., Harabasz, J.: A Dendrite Method for Cluster Analysis. Communication in Statistics, Vol.3. (1974) 1-27
9. Calantone, R. J., & Zhao, Y. S. (2000). Joint ventures in china: a comparative study of Japanese, Korean, and U.S. partners. *Journal* of International Marketing, Vol. 9, No. 1, 1-23
10. Choi, D. S., Park, S. J., Kim, J. W.: A Structured Analysis Model of Customer Loyalty in Online Games. Journal of MIS Research, Vol.11, No.3. (2001) 1-20
11. Davies, D. L., Bouldin, D. W.: A Cluster Separation Measure. IEEE Transactions Pattern Analysis and Machine Intelligence, Vol.1. (1979) 224-227
12. Davies, F. D., Bagozzi, R. P., Warshaw, P. R.: Extrinsic and Intrinsic Motivation to Use Computers in the Workplace. Journal of Application Society Psychology, Vol. 22, No. 14. (1992) 1111-1132

13. Dunn, J. C.: A Fuzz Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. Journal of Cybernetics, Vol. 3. (1973).32-57
14. Hagel, J., Armstrong, A.: Net Gain: Expanding Markets through Virtual Communities. Harvard Business School Press, Boston. (1997)
15. iResearch China Online Game Users Survey Report 2003, Shanghai, 2003.
16. KGDI: 2003 Korean White Game Paper. KGDI, Seoul. (2003)
17. KIEC: 2005 e-Business White Paper. Korea Institute for Electronic Commerce (KIEC), Seoul. (2005)
18. Lewinski, J. S.: Developer's Guide to Computer Game Design. Wordware Publishing Inc, Texas. (2000)
19. Maulik, U., Bandyopadhyay, S.: Performance Evaluation of Some Clustering Algorithms and Validity Indices. IEEE Transactions of Pattern Analysis and Machine Intelligence, Vol. 24, No.12. (2002) 1650-1654
20. Sanchez-Crespo, D.: 99 from a Game Development Perspective. Gamasutra. (1999). http://gamasutra.com/ features/19990802/siggraph_01.html
21. Shim, Y. S., Chung, J. W., Choi, I. C.: A Performance Comparison of Cluster Validity Indices based on K-means Algorithm. Journal of MIS Research, Vol.16, No.1. (2006) 127-144
22. Spector, W.: Remodeling RPGs for the New Millennium. Gamasutra. (1999). http://www.gamasutra.com/ features/game_desing/19990115/remodeling_01.htm
23. Vesanto, J., Alhoniemi, E.: Clustering of the Self-organizing Map. IEEE Transactions on Neural Networks, Vol.11, No.3. (2000) 586-600
24. Webster, J., Martocchio, J. J.: Micro Computer Playfulness: Development of a Measure with Workplace Implications. MIS Quarterly, Vol.16, No.2. (1992) 201- 226
25. Woodcock, W.: Game AI: the State of the Industry. Gamasutra. (1999). http://www.gamasutra.com /features/19990820/game_ai_01.html

A Neural Network Approach to Study O_3 and PM_{10} Concentration in Environmental Pollution

Giuseppe Acciani, Ernesto Chiarantoni, and Girolamo Fornarelli

Politecnico di Bari, Dipartimento di Elettronica ed Elettrotecnica,
Via E.Orabona 4, 70125 Bari, Italy
{acciani, chiarantoni}@poliba.it,
fornarelli@deemail.poliba.it

Abstract. In this paper two artificial neural networks are trained to determine Ozone and PM10 concentrations trying to model the environmental system. Then a method to partition the connection weights is used to calculate a relative importance index which returns the relative contribution of each chemical and meteorological input to the concentrations of Ozone and PM10. Moreover, an investigation of the variances of the input in the observation time contribute to understand which input mainly influence the output. Therefore a neural network trained only by the variables with higher values of relative importance index and low variability is used to improve the accuracy of the proposed model. The experimental results show that this approach could help to understand the environmental system.

1 Introduction

The analysis of environmental data has become an emergent question with the increase in human activity. One of the most important task is to investigate the conditions which cause concentrations of dangerous pollutants for environment and human health.

The photochemical smog is a mixture of pollutants that chemically reacts triggered by the solar radiation. Its generation is quite complex because a lot of variables are involved as concentration of precursors and meteorological conditions (wind, rain, pressure etc.). The high grade of complexity of the involved phenomena allows only a classification of pollutants in primary and secondary pollutants. The firsts are directly introduced in the air, the others come from a chemical reaction in the atmosphere. On the other hand, many important pollutants are the secondary pollutants contained in the photochemical smog. As a consequence it is hard to define the levels of danger for each pollutant, because they also depend on the meteorological conditions, their permanence in the atmosphere and the chain of involved reactions. For this reason, the monitoring activity has to be coupled with a forecasting activity. There are many studies which deal with environmental data and forecasting of air pollutants [1-6], even if there are not so many works about the possible relations between these pollutants and the meteorological factors or other chemical pollutants [7-9]. Instead this could be a very useful support to human activity of analysis devoted to determine the critical values of pollutants that produce dangerous conditions for environment and human health.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 913 – 922, 2006.

It has been reported that Ozone is the most important index substance of photochemical smog and it has been recognized as one of the key pollutants degrading the air quality. It is a highly reactive chemical, capable of attacking surfaces and materials. Moreover, it is also toxic to certain crops and can cause health hazards. Therefore, it is very important to determine the concentration of Ozone in the lower atmosphere [10] and in the last ten years many studies attempted a forecasting of Ozone concentration in metropolitan areas. On the other hand, atmospheric dusts with a diameter lower than 10μm (PM_{10}) are suspected to be the main cause of some health diseases. Therefore, the relevance of monitoring and predicting the concentrations of atmospheric dusts is becoming a basic task. For these reasons there is a considerable interest in determining the most influent causes on the concentrations of Ozone and PM_{10} in the lower atmosphere.

Mathematical models are often exploited to search for relations among different variables; but the creation of photochemical smog is intrinsically non-linear process and it could be very hard to arrange a mathematical model and to express the relations between the environmental conditions and a specific pollutant. On the contrary, neural networks appear to be an useful approach to deal with non linear systems like environmental pollution. In this paper a neural network approach to study Ozone and PM_{10} concentration in air pollution data is presented.

The present work starts from the method and results described by Garson [11] and Elkamel [7] and tries to improve the accuracy of the results. The aim is to understand the arising connections among single pollutants and both other chemical atmospheric components and meteorological conditions. The proposed approach is based on a partitioning method of connection weights of a trained Multi-Layer Perceptron neural network (MLP). The partitioning method allows to achieve a Relative Importance index (RI) which determines how much an element of input vector affects the output. In this way it seems possible to evaluate how a chemical atmospheric component and/or a meteorological event contribute to generate and keep a pollutant in lower atmosphere. The numerical results confirm the technique adopted by Elkamel et al. . Moreover our approach is validated by studying the variances of variables in the same data set. This analysis returns the variability of element values of vectors in the data set under test and shows that the vector elements with higher variability have less influence on the output, according with the results of RI index. Therefore a new MLP neural network is trained only with the less floating variables and the higher value of RI, obtaining a better accuracy. In other words, the inputs with large values of RI influence the output mostly while the ones with lower value could be considered as noise. Therefore they could be unnecessary to determine the output value.

This paper is organized as follows. In Section 2 the trained artificial neural networks to predict O_3 and PM_{10} concentrations are introduced and the adopted index to evaluate the relative contribution of each input variable is defined. Section 3 describes the data collection and the experimental results. Finally the conclusions are reported in Section 4.

2 Neural Network Model and Relative Importance Index

One of the advantages of neural network modeling is its ability to estimate the importance of each of the input variables using the network weights. For this reason, the connection weights were used to determine the RI index of chemical atmospheric components and meteorological events (neural network inputs).

MLPs are artificial neural networks consisting of a set of sensory units that constitute the input layer, one or more hidden layers and an output layer of computation nodes and the input signal propagates through the network in a forward direction. Neurons, arranged in these parallel layers, form weighted connections with the following layer. It is just this structure that can give advantages if a problem is modeled by an MLP. In fact, the connection weights can be used to interpret the influence of the input variables and understand the role played by each neuron in the hidden layer.

To assess the relative importance of the different input variables quantitatively, the connection weights of the trained neural network are used according to the procedure developed in [7]. The procedure essentially involves partitioning of the connection weights of each hidden neuron into the components associated with each input neuron. The equation proposed by Garson to evaluate the relative importance of the *i-th* component of the input vector for multilayer feed forward networks with *n* input neurons, one hidden layer with *h* neurons and *k* output neurons is as follows:

$$RI_{ik} = \frac{\sum_{j=1}^{h} \left| \frac{w_{ji} w_{kj}}{\sum_{i=1}^{n} |w_{ji}|} \right|}{\sum_{i=1}^{n} \sum_{j=1}^{h} \frac{|w_{kj}||w_{ji}|}{\sum_{i=1}^{n} |w_{ji}|}} \tag{1}$$

where w_{ji} is the connection weight between the *i-th* input neuron and the *j-th* hidden neuron and w_{kj} is the weight between the *j-th* hidden neuron and the *k-th* output neuron.

The relative importance RI_{ik} assumes higher values in correspondence with high values of the weights which link the *i-th* input neuron with the *k-th* output neuron, through the *j-th* neuron of the hidden layer. The denominator in the equation (1) is a factor of normalization.

Nevertheless the values of RI obtained by means of the above procedure lack of accuracy when the input data are unsufficient to train adequately the neural network. This method is so completed with an estimation of the input data variances: it has been experimentally found that the variances provide a significant aid to choose, among the inputs with high and near RI values, the ones more influencing the output. Therefore, the proposed approach consists of four steps: the training of an MLP neural network to evaluate the concentration of the pollutant under test; the identification of the inputs with the higher RI; the evaluation of the variances of the inputs to the neural network; the training of a second neural network by the pollutant with high RI and low variance.

3 Experimental Results

As it has been mentioned before, the first target of this work is to develop an MLP neural network for predicting Ozone and PM_{10} levels in photochemical smog. Therefore, the first step is to assemble data that can be used for training the network. In this case the focus of the data collection step is to generate data for the pollutants under investigation as a function of primary pollutants and meteorological conditions.

To perform this task the data of a monitoring station situated in a urban critical area of a city of Southern Italy have been considered [12]. The location of monitoring stations were selected to coincide with nodal confluence of principal road. A collection of measures of 8 chemical factors and 7 meteorological factors was taken into account, they are shown in table 1. A period of nine months from 5th of May 2000 to 2nd of February 2001 has been considered as study period. The measurements of chemical concentrations and meteorological conditions were recorded and collected on various days of different meteorological conditions and each hour for 24 hours a day. Each station is provided of a self validation system which is able to automatically validate each measured data. If some problems occur, for example about the sensors, the acquired datum is invalidated automatically and the datum is not stored. An off line human validation is then performed to avoid erroneous invalidations. During the data collection, and due to equipment maintenance or to the automatic invalidation of data, one or more of the variables may not have been measured or validated at a given time. In such cases all measures at that time instant were not considered in the data analysis. The number of complete data points with values for all 15 variables recorded was therefore reduced to 910 values for PM_{10} and to 2926 values for Ozone; each measure was collected at each time instant and for each monitoring station, in a vector $x(t) \subset \Re^{15}$.

Table 1. Chemical and meteorological input variables

Chemical variables	Unit	Meteorological variables	Unit
SO_2	[µg/m^3]	Wind Direction	[Sector]
NO_X	[ppb]	Wind Velocity	[m/s]
NO	[µg/m^3]	σ_p	[°]
NO_2	[µg/m^3]	DVG	[Sector]
CO	[mg/m^3]	Radiations	[W/m2]
Benzene	[µg/m^3]	Pressure	[mbar]
Toluene	[µg/m^3]	Rain	[mm]
O-xilene	[µg/m^3]		

Finally, a normalization of input and output data is required so that they are in the same range of the used transfer function. Data were normalized using:

$$x = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{2}$$

where x is the normalized value and, x_{min} and x_{max} are the minimum and maximum values of each vector of data respectively.

3.1 Neural Network Predictions

In the case of the Ozone prediction the 90% of the valid collected data were used as training set and the remaining 10% was used as testing set. After the data normalization, a network of 15 neurons as input layer, 5 neurons as hidden layer, and one neuron as output (concentration).

To verify a correct training of the neural network a linear regression coefficient between measured data and output of network was evaluated obtaining the value R=0,935. Obviously, it represents how near to measured data the predicted data are. In fig. 1 the dotted line represents the best fit between measured and simulated data, the solid line is the regression line. It is possible to see that the two lines are very close. As in the case of Ozone in PM_{10} prediction the 90% of the data were used to train the network. The remaining patterns were used as testing set. A normalization was then performed and a network with 15 neurons as input layer, 15 neurons as hidden layer, and one neuron as output layer was found to give the best result. Fig. 2 shows the same quantities of fig. 1 related with PM_{10}. It is worth noting that the regression coefficient (R=0,796) is lower than in the previous case. It is possible to argue that this value is due to the lack of data which does not allow a network training as efficient as in the first case.

3.2 Data Variances and RI index

The values of relative importance indexes evaluated on the neural network used in the prediction the Ozone concentration are given in fig. 3a. It can be seen that the main contribution to the output comes from CO, Toluene, Wind Velocity and Rain for the Ozone. Even if CO is not enough reactive and participates very little in the chemistry of Ozone formation, its importance stems from the fact that CO gives an indirect quantification of wind drift. It is also important to note that rain and wind velocity have an high value of relative importance. It is an expected result because the Ozone is a secondary pollutant and its concentration is known to depend on both the meteorological factors and the concentration of other chemicals (complex chemical interaction of oxides of nitrogen (NO_x)).

Fig. 3b displays relative importance about the prediction of the PM_{10} concentration. The most influent chemical variables in keeping PM_{10} concentration are CO and Benzene. It can be explained because the suspended dusts often contain benzene and carbon. In fact, this is one of the reason of its dangerousness. Moreover, the wind velocity exhibits a certain importance as it could be expected; as matter of fact, dust concentration can be influenced by this meteorological factor.

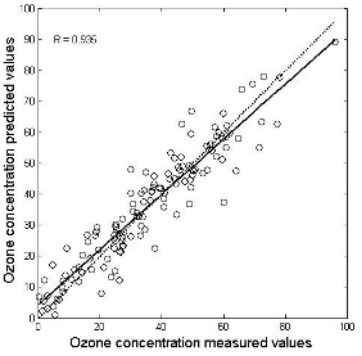


Fig. 1. Regression line and best fit line for Ozone concentration

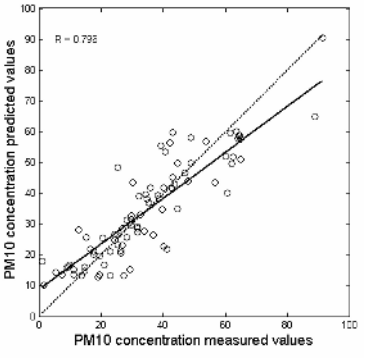


Fig. 2. Regression line and best fit line for PM_{10} concentration

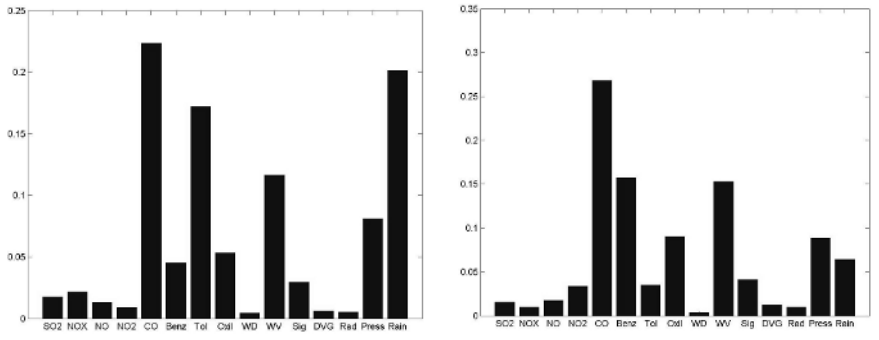
To confirm and, if it is possible, to improve the results obtained the proposed method is completed with an estimation of the input data variances.

The charts for the variances of the input variables have been generated and they have been called Variability Chart (VC). They display the variances of the values of vector elements as bars drawn in descending order, while the line shows the cumulative percentage. Fig. 4a shows the variability of chemical variables for Ozone while fig. 4b refers to the meteorological ones. In the same way fig. 4c and 4d are the charts for PM_{10}. Each bar is labeled with the name of the corresponding element of the input vector.

It is worth noting that the Variability Charts give a complementary information to that coming from RI indexes.

For sake of simplicity, it is possible to look at fig. 4c and 4d. The chemical pollutants and the meteorological variables, which have high variability, present a very low relative importance index (fig. 3).

Indeed four chemical and three meteorological variables (NO_x, NO, NO_2, SO_2, DVG, RADS, WD) have more than 90% of the whole variability of measured data. This analysis confirms that the variables with more influence in generating and keeping of PM_{10} are those with low variations with respect to their mean values.



(a) (b)

Fig. 3. Coefficient of relative importance of input variables on Ozone and PM_{10} concentration

The phenomenon could be explained with the fact that an environmental condition at a given moment is determined by pollutants and meteorological factors evaluated in that time. On the contrary, the keeping in atmosphere of a specific pollutant is due to the persistence of chemical substances and meteorological conditions. The VCs on the input data sets of Ozone give rise to analogous valuations.

The above considerations suggest a methodology to improve the accuracy of the neural approach to understand better the influence of the pollutant when the number of input data is not very high. Therefore in this paper the focus is on the PM_{10} concentration.

To upgrade the prediction of PM_{10} concentration a new MLP has been trained, using as elements of input vectors the variables with high value of RI and low variances. As matter of fact, the value of the variances has been used as a second criterion to split the variables with very close RI. For example Toulene and NO_2 can be considered, they have a RI value very close (fig. 3), but NO_2 has a greater value of variability. For this reason it is chosen as input Toulene but not NO_2. According this criterion 4 chemical and 5 meteorological variables are selected; they are reported in table 2.

Now the choice of the RI value that can be considered "high" becomes a problem of optimal threshold. In the tests of the present work the value of RI such that the input vector contains at least the 60% of all the initial variables has been found to give the best results.

By using the new training set a MLP network with 8 neurons as input layer, two hidden layers with 15 and 10 neurons respectively, and one neuron as output layer has been used to predict the PM_{10} concentration.

The regression coefficient obtained in this case was R = 0.91 (see fig. 5); it is worth noting that the last results are better than those obtained by means of the network trained by all measured variables. Therefore the RI index is very important to understand the influence of a specific factor, but the variability chart can be a useful support to underline the difference between two or more pollutants with similar RI values. Moreover the inputs with high variability do not allow a good training of neural network and could be considered as noise.

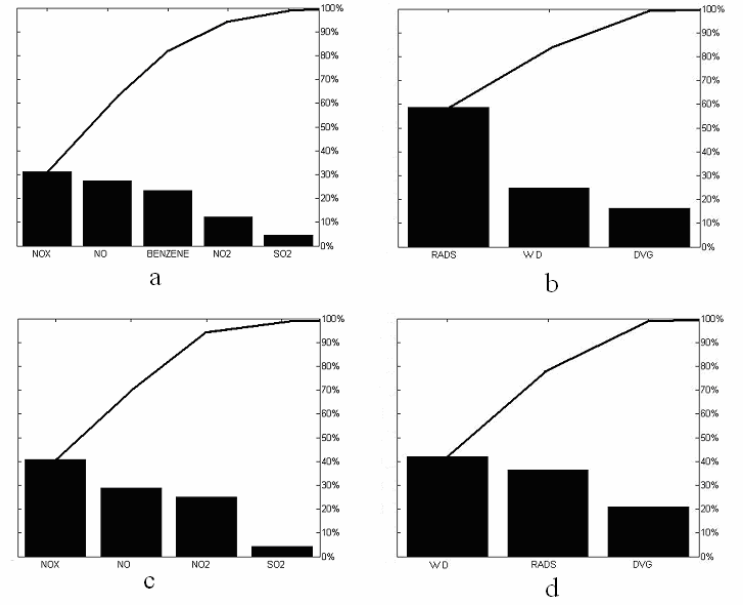


Fig. 4. Variability of chemical variables for Ozone (a); meteorological variables for Ozone (b); chemical variables for PM_{10} (c); meteorological variables for PM_{10} (d)

Table 2. Chemical and meteorological input variables with high RI for PM_{10}

Chemical variables	Unit	Meteorological variables	Unit
CO	[mg/m^3]	Wind Velocity	[m/s]
Benzene	[µg/m^3]	σ_p	[°]
Toluene	[µg/m^3]	Rain	[mm]
O-xilene	[µg/m^3]	Pressure	[mbar]

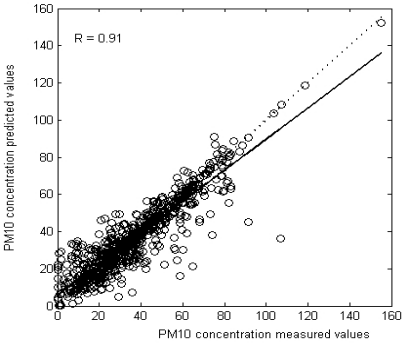


Fig. 5. Regression line and best fit line for PM_{10} concentration using only chemical and meteorological input variables with high RI and low variance

4 Conclusions

In this paper a neural network approach to study air pollution data is presented. The proposed approach is very simple and allows a more detailed analysis of the primary pollutants and meteorological events on the pollutants under study. It is based on a partitioning method of neural weights of a trained MLP. The partitioning method determines a Relative Importance index which provides the chemical components and the meteorological events with great contribution to generation and keeping in atmosphere of specific pollutant. The method is completed with an analysis of data by means of variability charts. It returns the variability of pollutants and meteorological factors measured, and shows that the factors with high variability have a little influence on a specific pollutant. For this reason, it is a useful support to highlight the difference between two or more pollutants with close values of RI. Therefore the proposed method could allow to understand the relation between a single pollutants and other chemical components in atmosphere and/or meteorological conditions.

The analysis has been focused on the concentrations of two pollutants, Ozone and PM_{10}. The results show how meteorological conditions and chemical pollutants are related to them. Rain, wind velocity, Carbon monoxide and Toluene were found to have a major effect on the Ozone concentration, whereas the Carbon monoxide, Benzene and wind velocity have an high influence in keeping PM_{10} concentration.

References

1. Corani G.: Air quality prediction in Milan: feed-forward neural networks, pruned neural networks and lazy learning. Ecological Modelling, Vol. 185, 2005, pp. 513–529.
2. Acciani G., Chiarantoni E., Fornarelli G., Vergura S.: A Feature Extraction Unsupervised Neural Network for Environmental Data Set. Neural Networks, Vol. 16, 2003, pp.427-436.
3. Mintz R., Young B. R., Svrcek W. Y.: Fuzzy logic modeling of surface ozone concentrations. Computers and Chemical Engineering, Vol. 29, 2005, pp. 2049–2059.
4. Hooyberghsa J., Mensinka C., Dumontb G., Fierensb F., Brasseurc O.: A neural network forecast for dailyaverage PM10 concentrations in Belgium. Atmospheric Environment, Vol. 39, 2005, pp. 3279–3289.
5. Abdollahian M., Foroughi R.: Optimal statistical model for forecasting ozone. International Conference on Information Technology: Coding and Computing, Vol. 1, 4-6 April 2005, pp. 169-173. on
6. Jeong-Sook H., Dong-Sool K.: A new method of ozone forecasting using fuzzy expert and neural network systems. Science of the Total Environment, Vol. 325, 2004, pp. 221–237.
7. Elkamel A., Abdul-Wahab S., Bouhamra W., Alper E.: Measurement and prediction of ozone levels around a heavily industralized area: a neural network approach. Advances in environmental research, Vol 5, 2001, pp. 47-59.
8. Muir D.: PM10 Particulates in Relation to Other Atmospheric Pollutants. Kluwer Academic Publishers, Vol. 52, 1998, pp. 29-42.
9. Barrero M. A., Grimalt J.O., Cantòn L.: Prediction of daily ozone concentration maxima in the urban atmosphere. Chemometrics and Intelligent Laboratory Systems, Vol. 80, 2006, pp. 67-76.

10. Brunekreef B., Holgate S. T.: Air pollution and Health. The Lancet, Vol. 360, 2002, pp. 1233-1242.
11. Garson G. D.: Interpreting Neural Network Connection Weights, AI Expert, 1991, pp. 47-51.
12. Borri D., Concilio G., Conte E.: A KBDSS for traffic air pollution control in urban environment. CUPUM Computers in Urban Planning and Urban Management Conference Hawaii, 2001, pp.10-14.

ROC Analysis as a Useful Tool for Performance Evaluation of Artificial Neural Networks

Fikret Tokan, Nurhan Türker, and Tülay Yıldırım

[1] Yıldız Technical University, Electrical and Electronics Faculty, Department of Electronics and Communication Engineering, Yıldız, Istanbul, 34349, Turkey
{ftokan, nturker, tulay}@yildiz.edu.tr

Abstract. In many applications of neural networks, the performance of the network is given by the classification accuracy. While obtaining the classification accuracies, the total true classification is computed, but the number of classification rates of the classes and fault classification rates are not given. This would not be enough for a problem having fatal importance. As an implementation example, a dataset having fatal importance is classified by MLP, RBF, GRNN, PNN and LVQ networks and the real performances of these networks are found by applying ROC analysis.

1 Introduction

Although neural networks are subject to criticism due to their "black-box" structures, the fact that neural networks can efficiently be trained for totally different applications has resulted as their use in diverse fields such as pattern recognition, speech processing, control, medical applications, and so forth. In these application examples, the performance of the networks is mostly considered as the number of true result/number of total data rate of the test set. This performance evaluation method would not be sufficient in conditions where a wrong decision may result in danger for human such as diagnosis of a cancer patient as healthy although he is ill or determination of an ally as an enemy in military applications. In these situations, the accuracy results of the network structures must be given with the rates of ill to ill and healty to ill diagnosis or enemy to enemy and enemy to ally decision rates. The sensitivity (SE); the proportion of patients with disease whose tests are positive, and specificity (SP); the proportion of patients without disease whose tests are negative and Receiver Operating Characteristics (ROC) curves, a trade off between specificity and sensitivity, should be given in order to predict the real performances of the networks.

In this work, an implementation of the ROC Analysis is realized to find the real performances of the networks used to classify the echocardiogram dataset, which is available in the machine learning database repository [1]. The neural networks investigated for this purpose are Multi Layer Perceptrons (MLP), Radial Basis Function Networks (RBF), Probabilistic Neural Networks (PNN), Generalized Regression Neural Networks (GRNN) and Learning Vector Quantization Networks (LVQ). To estimate the accuracy of the neural network models, cross-validation, which "provides a nearly unbiased estimate" of the accuracy, is used. In the following section, ROC analysis is defined in the form it is used in the analysis of the echocardiogram dataset.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 923 – 931, 2006.

The implementation example is given in the third section with a brief description of echocardiogram dataset, cross-validation method, applied neural network structures with their accuracies and ROC analysis of applied networks. The discussion on the ROC curves and sensitivity-specificity tables of the networks is given in the conclusion part.

2 Roc Analysis

Receiver Operating Characteristics (ROC) analysis is originated from signal detection theory, as a model of how well a receiver is able to detect a signal in the presence of noise. ROC analysis has also widely been used in medical data analysis to study the effect of varying threshold on the numerical outcome of a diagnostic test. Recently, it has been introduced to machine learning relatively in response to classification tasks with varying class distributions or misclassification costs [2]. In the following, ROC analysis definitions applied to medical diagnosis problems are given due to the selected dataset.

Commonly used diagnostic variables for the performance of a test are the sensitivity (SE) and specificity (SP). ROC of a classifier shows its performance as a trade off between specificity and sensitivity. Sensitivity is the proportion of patients with disease whose tests are positive and specificity is the proportion of patients without disease whose tests are negative. The equations of these measures can be given by (1) and (2) [3]:

$$sensitivity = \frac{number\,of\,true\,positives}{number\,of\,true\,positives + number\,of\,\,false\,negatives} \,. \tag{1}$$

$$specificit\,y = \frac{number\,\,of\,\,true\,negatives}{number\,\,of\,\,true\,negatives\,\, + number\,\,of\,\,\,false\,\,positives} \,. \tag{2}$$

where true positive (TP), true negative (TN), false positive (FP) and false negative (FN) means to diagnose ill as ill, healthy as healthy, ill as healthy and healthy as ill in medical diagnosis, respectively [4]. The ideal condition of SE and SP is to be one. Increasing either SE or SP will usually result in a decrease in the other measure.

Typically a curve of false positive rate versus true positive rate is plotted while a sensitivity or threshold parameter is varied. ROC curves are widely used in the medical literature to assess the performance of a diagnostic test. ROC curves contain a wealth of information for understanding and improving performance of classifiers but require visual inspection. When the curves are mixed it is hard to recognize the best classifier. The area under the ROC curve (AURC) helps to decide the appropriate one for the problem. AURC is a summary statistic of diagnostic performance and it assesses the ranking in terms of separation of the classes. The ROC curves are most helpful when comparing two or more risk stratification systems [5-7]. In the implementation part of this work, the real performances of MLP, RBF, PNN, GRNN and LVQ neural networks for the classification of echocardiogram dataset are compared by performing ROC analysis. For this purpose, ROC curves and SE-SP values are used.

3 An Implementation Example

The coronary attack is one of the most important and common reason for death all over the world. Since most of the deaths are from coronary heart disease, it is important to diagnose heart disease from simple clinical tests or determine whether a patient has risk factor after the coronary attack. This emphasizes the importance of an alternative method which may be helpful for early and accurate decision giving. Thus, so far many works have been realized on the prognosis of heart diseases using test results of the patients by researchers [8-11].

3.1 Echocardiogram Dataset

Echocardiogram dataset which takes place in the UCI repository of machine learning databases consists of some information about 132 patients who have suffered heart attacks at some point in the past. Some of the patients are alive after one year and some are not. The survival and still-alive variables, when taken together, indicate whether a patient survived for at least one year following the heart attack. The most difficult part of this problem is correctly predicting that the patient will not survive. This problem can be reduced by adding new samples to the dataset. The dataset has 13 raw attributes, however only 9 of them are used. All attributes are numeric-valued. The definitions of the 13 attributes are as follows:

1. Survival: the number of months patient survived (has survived, if patient is still alive);

It is possible that some patients have survived less than one year but they are still alive because all the patients had their heart attacks at different times. Thus, the second variable should be investigated to confirm this.

2. Still-alive: a binary variable (0: dead at end of survival period, 1: still alive);
3. Age at heart-attack: age when heart attack occurred;
4. Pericardial-effusion: Pericardial effusion is the fluid around the heart (0:no fluid, 1: fluid);
5. Fractional-shortening: a measure of contractility around the heart;
6. epss: E-point septal separation, another measure of contractility;
7. lvdd: left ventricular end-diastolic dimension. This is a measure of the size of the heart at end-diastole;
8. Wall-motion-score: It is a measure of how the segments of the left ventricle are moving;
9. Wall-motion-index: It is equal to the wall-motion-score divided by number of segments seen. Usually 12-13 segments are seen in an echocardiogram. This variable can be used instead of the wall-motion-score;
10. Mult: a derivate variable which can be ignored;
11. Name: the name of the patient;
12. Group: meaningless;
13. Alive after 1 year: Derived from the first two attributes (0: patient was either dead after 1 year or had been followed for less than 1 year, 1: patient was alive at 1 year).

Real-world data commonly contains instances with missing attribute values. The completion of the missing attribute values is one of the problems that most learning

models have to handle. The echocardiogram dataset used in this work has also missing attribute values in both input and output attributes. In this work, the missing input attribute values are completed by a pre-processing method [12-13] which is done by replacing the missing attribute values with the average value of the attribute and the instances with missing output attribute values are discarded. Since these methods transform the echocardiogram data before it is given to the neural network model, the pre-processing method used are applied both in training and testing. After the pre-processing, 117 instances; 24 instances from the class who were alive at one year and 93 instances from the class who were either dead after one year or had been followed for less than one year have remained in echocardiogram dataset.

3.2 Cross-Validation Method

In this work, different neural networks were used to decide whether a patient will live one year after a heart attack using echocardiogram dataset. To estimate the accuracy of the neural network models included in this work, cross-validation, which "provides a nearly unbiased estimate" of the accuracy, is used. Cross-validation in its simplest form is the division of a dataset into two subsets and training the network with one of the subsets while testing it with the other subset [14]. As noted in [14], cross-validation estimates of accuracy can have a high variability especially with small sample sizes, such as in echocardiogram dataset. Thus, in this work, two-fold cross validation method is used in order to remove the potential imbalance in the class distributions. The echocardiogram data set (117 instances) is divided into two subsets which are denoted as A and B. Classification accuracies of 1st case, which means the subset A is used for training and subset B is used for testing and of 2nd case, which means the subset B is used for training and subset A is used for testing, are obtained. Also the averages of these two cases are found.

3.3 Applied Neural Network Structures

MLP network, which has configuration of 8 input neurons, 5 neurons in hidden layer, and 1 output neuron with learning rate, 0.1, was trained for 400 epochs. Tangent sigmoid and logarithmic sigmoid transfer functions were used in MLP training. The input values have been normalized between 0 and 1. MLP network models were trained with almost all network learning algorithms. Among all these algorithms, the one giving the best results for MLP network, which is BFGS quasi-Newton (trainbfg) learning algorithm, takes place in Table 1. In RBF network, spread value is chosen as 0.1 which gives the best accuracy. The spread values are chosen as 0.1 for GRNN networks, 1.9 for PNN networks and 0.1 for LVQ networks. MATLAB 7.0 Neural Network Toolbox is used in the simulation of the networks.

3.4 Accuracy Results of the Networks

The classification accuracies obtained in two cases and the average of these cases for MLP, RBF, PNN, GRNN and LVQ networks are given in Table 1.

Table 1. Classification accuracies of MLP, RBF, GRNN, PNN and LVQ networks

	% Accuracy	MLP	RBF	GRNN	PNN	LVQ
1st case	TRAINING	100	100	100	94.92	100
	TEST	98.28	93.1	96.55	58.62	96.55
2nd case	TRAINING	100	98.28	100	87.93	98.28
	TEST	91.53	93.22	98.31	93.22	100
Average of two cases	TRAINING	100	99.14	100	91.42	99.14
	TEST	94.905	93.16	97.43	75.92	98.27

It can be seen from Table 1 that MLP network has highest classification accuracy in first case, whereas GRNN and LVQ have relatively high classification accuracies in second case. In the average, GRNN and LVQ networks again have high classification accuracies as in second case.

Prognostic performance is mostly defined by the accuracy of test which is the percentage of the prognostic decisions that turned out to be correct. By looking at the accuracy results in Table1, one would say that GRNN or LVQ is the best classifier. However, it would be wrong to say this before looking at the ROC analysis results. In the following section, the real performances of the networks are found by performing ROC analysis and which of these networks has the best performance for echocardiogram dataset is decided.

3.5 ROC Analysis of Applied Neural Networks

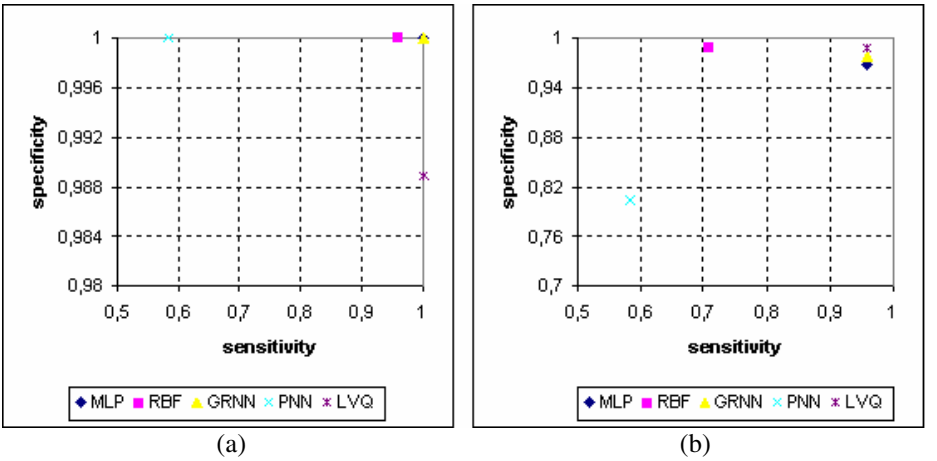
As the first step of ROC analysis, the sensitivity and specificity values of MLP, RBF, GRNN, PNN and LVQ networks are found as given in Table 2.

As can be seen from Table 2, the specificity values for both cases are all equal to nearly one in training. However, PNN has relatively low specificity value in testing. Also, sensitivity of PNN networks has the lowest value both in training and testing.

In Table 1, test accuracy of RBF network in the average case is %93.16. With this result, one may say that RBF networks are appropriate for the classification of echocardiogram data. However, in Table 2, the SP and SE of RBF in the average case are 0.989 and 0.708, respectively. This situation means that RBF networks are able to diagnose healty as healty with a good accuracy, but the percentage of diognosing ill to ill is only %70.8 which means %29.2 of the ill patients are classified as healty. Thus, RBF networks are not useful in the classification of the echocardiogram data. The SE-SP values of the five networks used are given in Fig. 1 for training and test sets of the average case.

Table 2. SE – SP values of MLP, RBF, GRNN, PNN and LVQ networks

			MLP	RBF	GRNN	PNN	LVQ
1st case	specificity	TRAIN	1	1	1	1	1
		TEST	1	1	0.978	0.608	0.978
	sensitivity	TRAIN	1	1	1	0.75	1
		TEST	0.916	0.666	0.916	0.50	0.916
2nd case	specificity	TRAIN	1	1	1	1	0.978
		TEST	0.936	0.978	0.978	1	1
	sensitivity	TRAIN	1	0.916	1	0.416	1
		TEST	1	0.75	1	0.666	1
Average of the 2 cases	specificity	TRAIN	1	1	1	1	0.989
		TEST	0.968	0.989	0.978	0.804	0.989
	sensitivity	TRAIN	1	0.958	1	0.583	1
		TEST	0.958	0.708	0.958	0.583	0.958



(a) (b)

Fig. 1. SE-SP graphics of the five networks for the average case in (a) training set, (b) test set

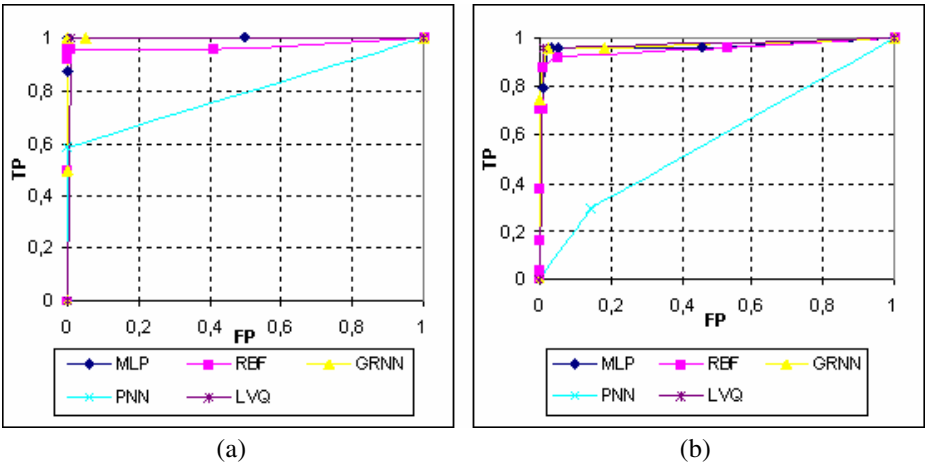
The distances between SE-SP values and (1,1) point would be helpful to put the networks used in order according to their performances when the points are near to each other. The ideal condition of SE and SP is to be one. In Table 3, the distances SE-SP values to (1,1) point for MLP, RBF, PNN, GRNN and LVQ networks are

given. LVQ network can be chosen as the best network with distance of 0.043 in testing for the average case. GRNN and MLP networks come after LVQ networks with distances of 0.047 and 0.052, respectively.

Table 3. The distances SE-SP values to (1,1) point for MLP, RBF, PNN, GRNN and LVQ networks

SE-SP distances		1st case	2nd case	Average
MLP	train	0	0	0
	test	0.084	0.064	0.052
RBF	train	0	0.084	0.042
	test	0.334	0.251	0.292
GRNN	train	0	0	0
	test	0.086	0.022	0.047
PNN	train	0.250	0.583	0.414
	test	0.635	0.333	0.460
LVQ	train	0	0.022	0.011
	test	0.086	0	0.043

As the second step of the ROC analysis, ROC curves of the networks which contain information for understanding the performances of the networks are obtained. In Fig.2, ROC curves of the MLP, RBF, GRNN, PNN and LVQ networks for training and test sets are given for the average cases.



(a) (b)

Fig. 2. ROC curves of the MLP, RBF, GRNN, PNN and LVQ networks in (a) training set; (b) test set for the average of two cases

These figures require visual inspection. When the curves of the networks are hard to investigate by visual inspection, the area under the ROC curve (AURC) would be helpful to compare the performances of the networks. In Table 4, the AURC values for MLP, RBF, GRNN, PNN and LVQ networks are given. As it was chosen from

SE-SP distance values in Table 2, again LVQ network with area of 0.9737 in test set for the average can be chosen as the best classifier for echocardiogram dataset. GRNN with area of 0.9731 and MLP network with area of 0.9628 come after LVQ. Although RBF network with area of 0.9530 seems to have good performance for this problem, it is not appropriate due to its comparatively low SE-SP values.

Table 4. The area under the ROC curves of MLP, RBF, PNN, GRNN and LVQ networks

AURC		1st case	2nd case	Average
MLP	train	1	1	1
	test	0.9565	0.9823	0.9628
RBF	train	1	0.9420	0.9707
	test	0.9203	0.9947	0.9530
GRNN	train	1	0.9420	1
	test	0.9529	0.9973	0.9731
PNN	train	0.8750	0.7083	0.7917
	test	0.5543	0.8333	0.5729
LVQ	train	1	0.9891	0.9946
	test	0.9475	1	0.9737

4 Conclusion

In this work, an implementation of the ROC Analysis is realized to find the real performances of the networks which give reasonably good solutions to prognosis problems are used to classify the echocardiogram dataset available in the machine learning database repository. The neural networks investigated for this purpose are Multi Layer Perceptrons (MLP), Radial Basis Function Networks (RBF), Probabilistic Neural Networks (PNN), Generalized Regression Neural Networks (GRNN) and Learning Vector Quantization Networks (LVQ). To estimate the accuracy of the neural network models two-fold cross-validation method is used.

In many applications of neural networks, the performance of the network is given by the classification accuracy. While obtaining the classification accuracies, the total true diagnosis (healthy to healthy and ill to ill) is computed, but the number of classification rates of the classes and fault classification (healthy to ill and ill to healthy) rates are not given. This would not be enough for a problem having fatal importance as in our implementation. Thus, ROC analysis of the networks is performed in order to decide which of these networks give the best performance. From the sensitivity-specificity values and ROC curves, LVQ network is determined as the network having the best performance for evaluating the echocardiogram dataset. GRNN and MLP networks come after LVQ network. Although RBF network seems to have good performance for this problem from ROC curves, it is not appropriate due to its low SE-SP values. Thus, when performing ROC analysis, not only one of the ROC curve or SE-SP value, both of them should be taken into account.

References

1. Murphy, P. M., Aha, D. W.: UCI Repository of Machine Learning Databases. University of California, Department of Information and Computer Science (ics.uci.edu). Irvine, CA
2. Flach, P. A.: Tutorial on "The Many Faces of ROC Analysis in Machine Learning". The Twenty-First International Conference on Machine Learning, Banff Canada, (2004)
3. Sboner, A., Eccher, C., Blanzieri, E., Bauer, P., Cristofolini, M., Zumiani, G., and S. Forti, S.: A Multiple Classifier System for Early Melanoma Diagnosis. AI in Medicine, Vol. 27. (2003) 29-44
4. Metz, C. E.: Basic Principles of ROC Analysis. Sem Nuc Med, (1978) 283-298
5. Pinna-Pintor, P., Bobbio, M., Colangelo, S., Veglia, F., Giammaria, M., Cuni, D., Maisano F., and Alfieri, O.: Inaccuracy of Four Coronary Surgery Risk-Adjusted Models to Predict Mortality in Individual Patients. Eur J Cardiothorac Surg, (2002) 199-204
6. Karabulut, H., Toraman, F., Alhan, C., Camur, G., Evrenkaya, S., Dagdelen, S., and Tarcan, S.: EuroSCORE Overestimates the Cardiac Operative Risk. Cardiovasc Surg, (2003) 295–8
7. Gefeller, O., Brenner, H.: How to Correct for Chance Agreement in the Estimation of Sensitivity and Specificity of Diagnostic Tests. Methods Inf Med, (1994) 180-6
8. Gennari, J. H., Langley P., and Fisher, D.: Models of Incremental Concept Formation. Artificial Intelligence. (1989) 11-61
9. Li, T., Zhu, S., and Ogihara, M.: A New Distributed Data Mining Model Based on Similarity. Proceedings of the 2003 ACM symposium on Applied computing, Melbourne Florida, (2003) 432 – 436
10. Wei, L., Altman, R. B.: An Automated System for Generating Comparative Disease Profiles and Making Diagnoses. IEEE Transactions on Neural Networks, Vol. 15. (2004) 597
11. Poirazi, P., Neocleous, C., Pattichis, C. S., and Schizas, C. N.: Classification Capacity of a Modular Neural Network Implementing Neurally Inspired Architecture and Training Rules. IEEE Transactıons On Neural Networks, Vol. 5. (2004) 597-612
12. Prechelt, L.: Summary of Usenet Discussion on Unknown Input Values in Neural Networks. (1994)
13. Vamplew, P., Adams, A.: Missing Values in Backpropagation Neural Net. Technical Report Department of Computer Science, University of Tasmania (1993)
14. Efron, B.: Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. Journal of the American Statistical Association, (1983) 316-331

NewPR-Combining TFIDF with Pagerank

Hao-ming Wang[1,2], Martin Rajman[2], Ye Guo[3], and Bo-qin Feng[1]

[1] School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, P.R. China
{wanghm, bqfeng}@mail.xjtu.edu.cn
[2] School of I & C, Swiss Federal Institute of Technology(EPFL), 1015 Lausanne, Switzerland
Martin.Rajman@epfl.ch
[3] School of Information, Xi'an University of Finance & Economics, Xi'an, Shaanxi 710061, P.R. China
guoyexinxi@126.com

Abstract. TFIDF was widely used in IR system based on the vector space model (VSM). Pagerank was used in systems based on hyper-link structure such as Google. It was necessary to develop a technique combining the advantages of two systems. In this paper, we drew up a framework by using the content of web pages and the out-link information synchronously. We set up a matrix M, which composed of out-link information and the relevant value of web pages with the given query. The relevant value was denoted by TFIDF. We got the NewPR (New Pagerank) by solving the equation with the coefficient M. Experimental results showed that more pages, which were more important both in content and hyper-link sides, were selected.

1 Introduction

With information proliferate on the web as well as popularity of Internet, how to locate related information as well as providing accordingly information interpretation has created big challenges for research in the fields of data engineering, IR as well as data mining due to features of Web (huge volume, heterogeneous, dynamic and semi-structured etc.). [1, 2]

As a user, in order to find, collect and maintenance the information, which maybe useful for the specific aims, s/he has to pay more time, money and attention on the retrieval course.

While web search engine can retrieve information on the Web for a specific topic, users have to step a long ordered list in order to locate the valuable information, which is often tedious and less efficient due to various reasons like huge volume of information. For most of the users, they may not express their needs clearly with a few keywords. Users may be just interested in "most qualified" information or one peculiar part of returned information.

The search engines are based on one of the two methods, the content of the pages and the link structure.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 932–942, 2006.

The first kind of search engineers works well for traditional documents, but the performance drops significant when applied to the web pages. The main reason is that there are too much irrelevant information contained in a web page.

The second one takes the hyperlink structures of web pages into account in order to improve the performance. The examples are `Pagerank` and `HITS`. They are applied to `Google` and the `CLEVER` project respectively.

However, these algorithms have shortcomings in that (1) the weight for a web page is merely defined; and (2) the relativity of contents among hyper linking web pages is not considered. [2]

In this paper, we combine the relevance and the Pagerank of the web page in order to refine the retrieval results. We compute the `TFIDF` value firstly. And then, we compute the new Pagerank by the `TFIDF` and the out-link information of every page. The new Pagerank is called `NewPR`.

This paper is organized as follows: Section 2 introduces the concept of Pagerank and `TFIDF`. Section 3 describes the algorithm of `NewPR`. Section 4 presents the experimental results for evaluating our proposed methods. Finally, we conclude the paper with a summary and directions for future work in Section 5.

2 Basic Concept

2.1 Pagerank

The Google search engine is based on the popular Pagerank algorithm first introduced by Brin and Page in Ref. [3].

Considering the pages and the links as a graph $G = P(Page, Link)$, we can describe the graph by using the adjacency matrix. The entries of the matrix, for example p_{ij}, can be defined as:

$$p_{ij} = \begin{cases} 1 & \exists\, Link(i \to j) \\ 0 & Otherwise. \end{cases}$$

Here $i, j \in (1, n)$ and n is the number of web pages. Because the total probability from one page to others can be considered 1, the rows, which correspond to pages with a non-zero number of out-links $deg(i) > 0$, can be made row-stochastic (row entries non-negative and sum to 1) by setting $p_{ij} = p_{ij}/deg(i)$. That means if the page u has m out-links, the probability of following each of out-links is $1/m$. We assume all the m out-links from page u have the similar probability.

For a real adjacency matrix P, in fact, there are many special pages without any out-link, which are called *dangling page*. Any other pages can reach the dangling page in $n(n \geq 1)$ steps, but it is impossible to get out. In the adjacency matrix, the row, corresponding to the dangling page is all zeros. Thus, the matrix P is not a row-stochastic. It should be deal with in order to meet the requirement of the row-stochastic.

One of the ways to overcome this difficulty is to change the transition matrix P slightly. We can replace the rows, all of the zeros, with $v = (1/n)e^T$, where

e^T is the row vector of all 1s and n is the number of pages of P contains. The P will be changed to $P' = P + d \cdot v^T$. Where

$$d = \begin{cases} 1 & if \quad deg(i) = 0 \\ 0 & Otherwise. \end{cases}$$

is the dangling page indictor [4]. If there were a page without any out-link from it, we could assume it can link to every other pages in P with the same probability. After that there is not row with all 0s in matrix P'.

P' is row-stochastic and it corresponds to the stochastic transition matrix over the graph G. Pagerank can be viewed as the stationary probability distribution over pages induced by a random walk on the web. It can be defined as a limiting solution of the iterative process.

Because of the existing of zero entries in the matrix P', it cannot guarantee the existence of the stationary vector. The problem comes from that the P' may be reducible. In order to solve the problem, P' can be modified by adding the connection between every pair of pages [4].

$$Q = P'' = cP' + (1-c)ev^T, \quad e = (1, 1, \cdots, 1)^T.$$

Where c is called dangling factor, and $c \in (0, 1)$. In most of the references, the c is set [0.85,1). [3]

After that, the Q is irreducible because all of the pages are connected (strong connection). For $Q_{ii}^{(k)} > 0, (i, k \in (1, n))$, the Q is aperiodic too. The Perron-Frobenius theorem guarantees the equation $x^{(k+1)} = Q^T x^{(k)}$ (for the eigensystem $Q^T x = x$) converges to the principal eigenvector with eigenvalue 1, and there is a real, positive, and the biggest eigenvector. [5,6]

2.2 TFIDF

`TFIDF` is the most common weighting method used to describe documents in the Vector Space Model (VSM), particularly in IR problems. Regarding text categorization, this weighting function has been particularly related to two important machine learning methods: kNN (k-nearest neighbor) and SVM(Support Vector Machine). The `TFIDF` function weights each vector component (each of them relating to a word of the vocabulary) of each document on the following basis. [7]

Assuming vector $\tilde{d} = (d^{(1)}, d^{(2)}, ..., d^{|F|})$ represents the document d in a vector space. Each dimension of the vector space represents a word selected by the feature selection. The value of the vector element $d^{(i)} (i \in [1, |F|])$ is calculated as a combination of the statistics $TF(w, d)$ and $DF(w)$.

$TF(w, d)$ is the number of the word w occurred in document d. $DF(w)$ is the number of documents in which the word w occurred at least once time. The $IDF(w)$ can be calculated as

$$IDF(w) = log \frac{N_{all}}{DF(w)}.$$

Where N_{all} is the total number of documents. The value $d^{(i)}$ of feature w_i for the document d is then calculated as $d^{(i)} = TF(w_i, d) \times IDF(w_i)$. Where $d^{(i)}$ is called the weight of word w_i in document d. [7]

The TFIDF algorithm learns a class model by combining document vectors into a prototype vector $\tilde{C}$ for every class $C \in$ Ç. Prototype vectors are generated by adding the document vectors of all documents in the class.

$$\tilde{C} = \sum_{d \in C} \tilde{d}.$$

This model can be used to classify a new document d'. Assuming vector $\tilde{d'}$ represents d', the cosine distance between $\tilde{d'}$ and $\tilde{C}$ is calculated. The d' is belonged to the class with which the cosine distance has the highest value.

3 Algorithm of the NewPR

3.1 Precision and Recall

For a retrieval system, there are 2 sides should be considered, the *precision* and the *recall*. Just as the illustrator in Fig.1, we can get,

$$Precision = \frac{B}{Ret}; \quad Recall = \frac{B}{Ref}; \quad \gamma = \frac{Ref}{A+B+C+D} = \frac{Ref}{N}.$$

For a given retrieval system, the average value of *precision* and γ can be estimated. As the N is very large, γ is expected to be very small.

3.2 Page Link

We donate the query from the user with Q, all of the pages selected by retrieval system relevant to Q with $Y = \{y_i, i \in (1, n)\}$. The probability from Q to Y is $P = \{p_i, i \in (1, n)\}$, and from y_i returns to Q is $1 - \pi$. In our experiment, P is the TFIDF values of Q to Y.

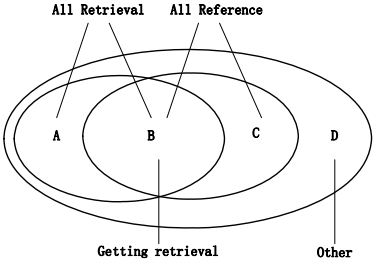


Fig.1 Concept of Information Retrieval

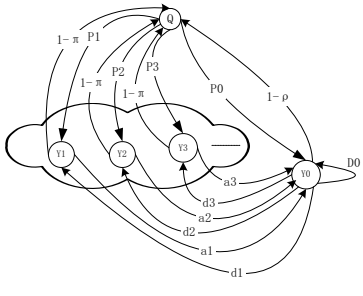
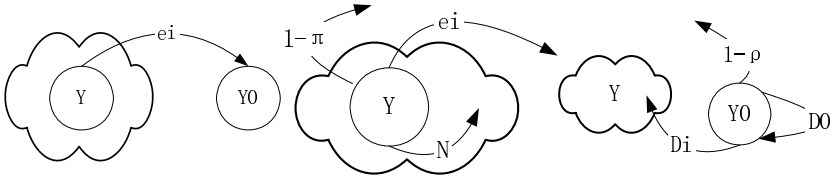


Fig.2 Information of Links



(A) From Y to Y_0 (B) From Y to others (C) From Y_0 to others

Fig. 3. Link Information of each page

We assume all the pages, which are not included in set Y, are included in a set Y_0. The probability of Y_0 transfers to itself is D_0, to Y is $D = \{d_i, i \in (1,n)\}$ and to Q is $1-\rho$. p_0 is probability from Q to Y_0. It is the sum of `TFIDF` values of Q to Y_0. The link information is showed in Fig.2.

Because the return link from Y to Q means the page belonged to part A in Fig.1, the probability $1-\pi = \dfrac{A}{Ret} = \dfrac{A+B-B}{Ret} = \dfrac{Ret-B}{Ret} = 1-Precision. \Rightarrow \pi = Precision.$

For the Q, assuming s_i is the `TFIDF` value, we get,

$$p_0 + \sum_{i=1}^{n} p_i = 1, \quad p_0 = \beta \sum_{i \notin (1,n)} s_i, \quad p_i = \beta s_i \Rightarrow \beta \sum_{i \notin (1,n)} s_i + \beta \sum_{i \in (1,n)} s_i = 1$$

$$\Rightarrow \quad \beta = \frac{1}{\sum\limits_{i \in ALL} s_i}, \quad p_0 = 1 - \frac{\sum\limits_{i \in (1,n)} s_i}{\sum\limits_{i \in ALL} s_i}, \quad p_i = \frac{s_i}{\sum\limits_{i \in ALL} s_i}. \quad (1)$$

In Fig.3(A), we assume the probability of page $y_i \in Y$ points to Y_0 is $e_i = \sum\limits_{j \notin Ret} n_{ij}$, where n_{ij} is the initial probability that page i points to page j.

In Fig.3(B), the page $y_i \in Y$ has three kinds of links: links to Q, links to Y, and links to Y_0. Thus, we get $(1-\pi) + e_i + \sum\limits_{j \in Ret} n_{ij} = 1$.

We define the link matrix $U = \{u_{ij} | i, j \in (1,n)\}$ as,

$$u_{ij} = \begin{cases} u_{ii} = 1 & \sum\limits_j u_{ij} = 0 \quad Dangling \quad page \\ 1 & \sum\limits_j u_{ij} > 0 \quad and \exists link\,(i \to j) \\ 0 & \sum\limits_j u_{ij} > 0 \quad and \nexists link\,(i \to j). \end{cases}$$

For the Y, we get

$$(1-\pi) + \sum_{j \in Ret} \beta u_{ij} + \sum_{j \notin Ret} \beta u_{ij} = (1-\pi) + \beta \sum_{j \in ALL} u_{ij} = 1$$

$$\Rightarrow \begin{cases} \beta & = \dfrac{\pi}{\sum\limits_{j\in ALL} u_{ij}} = \dfrac{\pi}{OutlinkNum(i)}, \quad n_{ij} = \pi \dfrac{u_{ij}}{OutlinkNum(i)} \\ e_i & = \pi(1 - \dfrac{\sum\limits_{j\in Ret} u_{ij}}{OutlinkNum(i)}) = \pi(\dfrac{\sum\limits_{j\notin Ret} u_{ij}}{OutlinkNum(i)}). \end{cases} \tag{2}$$

For the Y_0, which is showed in Fig.3(C), we get

$$\rho = \frac{C}{C+D} = \frac{Ref - B}{N - Ret} = \frac{\gamma N - \pi Ret}{N - Ret} \approx \frac{\gamma N - \pi Ret}{N} \approx \gamma.$$

$$(1-\rho) + D_0 + \sum_{i\in Ret} d_i = (1-\rho) + \beta \sum_{j\notin Ret} \sum_{i\notin Ret} u_{ji} + \beta \sum_{j\notin Ret} \sum_{i\in Ret} u_{ji} = 1$$

$$\Rightarrow \quad D_0 = \rho \frac{\sum\limits_{j\notin Ret} \sum\limits_{i\notin Ret} u_{ji}}{\sum\limits_{j\notin Ret} OutlinkNum(j)} \quad , \quad d_i = \rho \frac{\sum\limits_{j\notin Ret, i\in Ret} u_{ji}}{\sum\limits_{j\notin Ret} OutlinkNum(j)}. \tag{3}$$

3.3 The Link Matrix

We assume the links among the pages in set Y composed the link matrix U.

$$\mathbf{U} = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n1} & u_{n2} & \dots & u_{nn} \end{pmatrix} \quad \widetilde{\mathbf{U}} = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} & AA_1 \\ u_{21} & u_{22} & \dots & u_{2n} & AA_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ u_{n1} & u_{n2} & \dots & u_{nn} & AA_n \\ BB_1 & BB_2 & \dots & BB_n & BB_0 \end{pmatrix}$$

Adding the set Y_0 , U changes to $\widetilde{U}$. Where

$$AA_i = \sum_{j\notin Ret} u_{ij}, \quad BB_i = \sum_{j\notin Ret, i\in Ret} u_{ji}, \quad BB_0 = \sum_{i,j\notin Ret} u_{ij}.$$

We normalize the $\widetilde{U}$ by

$$\widetilde{m_{ij}} = \frac{\widetilde{u_{ij}}}{\sum\limits_j \widetilde{u_{ij}}} \quad i \in (1,n]\,; \quad \widetilde{a_i} = \frac{\widetilde{u_{ij}}}{\sum\limits_j \widetilde{u_{ij}}} \quad i \in (n, ALL)\,;$$

$$b_i = \frac{\sum\limits_{j\notin Ret, i\in Ret} u_{ji}}{\sum\limits_{j\notin Ret} OutlinkNum(j)}\,; \quad B_0 = \frac{\sum\limits_{j\notin Ret} \sum\limits_{i\notin Ret} u_{ji}}{\sum\limits_{j\notin Ret} OutlinkNum(j)}\,.$$

Adding the query , we get the transfer matrix T,

$$\mathbf{T} = \begin{pmatrix} 0 & p_1 & p_2 & \dots & p_n & p_0 \\ 1-\pi & n_{11} & n_{12} & \dots & n_{1n} & e_1 \\ 1-\pi & n_{21} & n_{22} & \dots & n_{2n} & e_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1-\pi & n_{n1} & n_{n2} & \dots & n_{nn} & e_n \\ 1-\rho & d_1 & d_2 & \dots & d_n & D_0 \end{pmatrix} = \begin{pmatrix} 0 & P & p_0 \\ 1-\pi & \pi M & \pi A \\ 1-\rho & \rho B & \rho B_0 \end{pmatrix}.$$

Where $A = (a_1, a_2, \cdots, a_n)'$,$B = (b_1, b_2, \cdots, b_n)$. P is the normalized value of TFIDF of Q to page $y_i (y_i \in Y)$. p_0 is the sum of normalized value of TFIDF of Q to pages in Y_0.

For a giving retrieval system, we could compute the B_0 and $B_i (i \in (1, n))$. We can get

$$\mathbf{T}' = \begin{pmatrix} 0 & 1-\pi & 1-\rho \\ P' & \pi M' & \rho B \\ p_0 & \pi A' & \rho B_0 \end{pmatrix}.$$

3.4 Computing Equation

From the equation $T'X = X$, we can get,

$$\begin{pmatrix} 0 & 1-\pi & 1-\rho \\ P' & \pi M' & \rho B \\ p_0 & \pi A' & \rho B_0 \end{pmatrix} \begin{pmatrix} x_0 \\ Y \\ y_0 \end{pmatrix} = \begin{pmatrix} x_0 \\ Y \\ y_0 \end{pmatrix}$$

$$x_0 = (1-\pi)\|Y\|_1 + (1-\rho)y_0 \tag{4}$$
$$Y = x_0 P' + \pi M' Y + \rho y_0 B' \tag{5}$$
$$y_0 = x_0 p_0 + \pi A' Y + \rho B_0 y_0 \tag{6}$$
$$x_0 + \|Y\|_1 + y_0 = 1 \tag{7}$$

As the T is stochastic matrix, we get (7).

Changing (6), we get,

$$y_0 = \frac{x_0}{1-\rho B_0} p_0 + \frac{\pi}{1-\rho B_0} A' Y \,. \tag{8}$$

Changing (5), we get,

$$(I - \pi M' - \frac{\rho\pi}{1-\rho B_0} B' A') Y = x_0 (P' + \frac{\rho p_0}{1-\rho B_0} B') \,. \tag{9}$$

Assuming $C = \frac{\rho}{1-\rho B_0} B'$, we get,

$$y = x_0 [I - \pi(M' + CA')]^{-1} (P' + p_0 C) \,. \tag{10}$$

Assuming $V = [I - \pi(M' + CA')]^{-1}(P' + p_0 C)$, we get,

$$Y = x_0 V \Rightarrow \|Y\|_1 = x_0 \|V\|_1 \,. \tag{11}$$

Changing (4), we get,

$$[1 - (1-\pi)\|V\|_1] x_0 = (1-\rho) y_0$$
$$y_0 = \frac{1 - (1-\pi)\|V\|_1}{1-\rho} x_0 \,. \tag{12}$$

Combining the formula (7)(11)(12), we get

$$x_0 = \frac{1}{1 + \dfrac{1 + (\pi - \rho)\|V\|_1}{1-\rho}} \quad ; \quad y_0 = \frac{1 - (1-\pi)\|V\|_1}{1-\rho} x_0 \quad ; \quad Y = x_0 X \,. \tag{13}$$

4 Experimental

4.1 Experimental Setup

We construct experiment in order to verify the retrieval methods of our approach described in Section 3.

The experiment is constructed by using the `TREC WT10g` test collection, which contains about 1.69 million Web pages. Stop words have been eliminated from all Web pages in the collection based on the stop-word list and stemming has been performed using Porter Stemmer. [7]

(1) Selecting test pages. We construct the set R with all pages which are relevant to the query $q_i, i \in (1, 100)$. The data-set D can be set up just as

$$d_i = \begin{cases} d_i, & (d_i \in R); \\ d_j, & \exists\,(link(i \to j) \wedge link(j \to k)),\ (j \notin R; \quad i,k \in R) \\ d_j, d_l & \exists\,(link(i \to j) \wedge link(j \to l) \wedge link(l \to k)),\ (j,l \notin R; \quad i,k \in R)\,. \end{cases}$$

We name all pages in D from 1 to 12486 and pick up all out-links from those pages.

(2) Computing the old Pagerank. In order to compare the result of new method with the traditional one, we compute the pagerank of the every page in traditional way firstly. In this method, we ignore the last column of link matrix P , and it guarantee the link matrix is square one.

It must be noticed that the pagerank value of pages in our experiment are not very precise. The reason is that we consider the link information of pages belonged to the data set D only. There may be many important links out of the D have not be considered. Table. 1 shows the top 10 results of pagerank according to the traditional method.

(3) Computing the NewPR. We compute the NewPR by using *Matlab* with the parameter of link matrix P. The formula (1)(2)(3)(13)have been mentioned above. In the program, we assume the two parameters $\pi = 0.6$ and $\rho = 0.1$. Table. 3 shows the NewPR of the query 511. The detail of this query can be checked in `WT10g`. Due to the capability of the computer, we compute the first 5000 pages.

4.2 Experiment Results

In order to compare the two methods, the OldPR and the NewPR, we need to consider two questions, (1) Are the NewPR and the OldPR similar? (2) Is the NewPR better than OldPR?

We can compute the Spearman Rank Correlation Coefficient in order to determine the difference between the OldPR and the NewPR. The Spearman Rank Correlation Coefficient is defined by

$$r' = 1 - 6 \sum \frac{d^2}{N(N^2 - 1)}.$$

Table 1. OldPagerank

Rank	Page_No	Old_Value
1	3971	10.00000
2	3973	10.00000
3	3976	10.00000
4	3682	5.91328
5	3897	5.47881
6	3898	5.47881
7	3901	5.47881
8	1104	5.28466
9	1664	4.89549
10	1396	4.81016
...	...	...

Table 2. TFIDF

Rank	Page_No	TFIDF-Q511
1	1798	16.52122
2	1805	16.40058
3	7280	16.28200
4	1609	16.20084
5	1787	16.09591
6	11459	15.90077
7	1780	15.85325
8	1851	15.83046
9	1745	15.80058
10	11443	15.76907
...	...	...

Table 3. NewPR

Rank	Page_No	NewPR
1	3560	0.64832
2	2597	0.47899
3	3553	0.46598
4	3558	0.41606
5	3559	0.40520
6	1673	0.39539
7	2857	0.35789
8	4848	0.31863
9	1776	0.30918
10	1790	0.29264
...	...	...

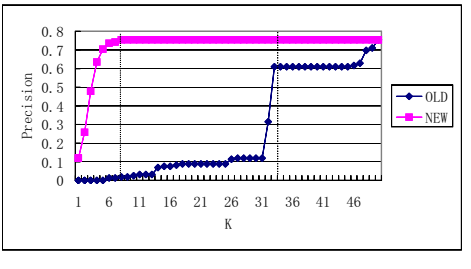
Table 4. Rank

Page_No	Old_Rank	New_Rank	d^2
1	1972	1319	426409
2	949	2283	1779556
3	1973	2304	109561
4	1974	2150	30976
5	4625	2203	5866084
6	1975	2216	58081
7	1976	2243	71289
8	4798	2255	6466849
...	...	...	...

Table 5. Precision

K	Old_Num of Rele	New_Num of Rele	Old_Prec	New_Prec
100	0	19	0.00000	0.11950
200	0	41	0.00000	0.25786
300	0	76	0.00000	0.47799
400	0	101	0.00000	0.63522
500	0	112	0.00000	0.70440
600	2	117	0.01258	0.73585
700	2	118	0.01258	0.74214
800	3	120	0.01887	0.75472
...	...	...	...	...

Table 6. Feedback Rele/AllRele



Where N is the number of total pages, and d is the difference in statistical rank of corresponding variables, and $r' \in [-1, +1]$. $r' = 0$ means that there is no correlation between the two quantities. They are completely independent of one another. Table.4 shows the Old_Rank, New_Rank and the d^2. We can compute $r' = 0.0046$ of all 5000 pages. That means the two algorithms, OldPR and NewPR are almost independent. This result answers the first question.

For the second question, we check the first top 100, 200, $\cdots$, 5000 pages of two methods, calculate the number of pages related to the query 511. In order to

compare the speeds of two methods' of reaching the maximal number of relevance pages, we compute the *precision*, ratio of relevance pages in the feedback pages list over all relevance pages. The result is showed in Table.5. From Table.6, we can find that the speed of new method is faster than that of old one. In the new method, it reach the top value in about 800 pages, meanwhile it needs almost all 5000 pages in the old method.

5 Conclusion

This paper introduces the methods of information retrieval on the web, and the concept of `TFIDF` and `Pagerank`. Due to the different methods of these two kinds of technologies use, the `TFIDF` cannot reflect the link information among pages. Meanwhile the `Pagerank` does not consider the content of pages.

We draw up a new framework by combining the `TFIDF` and `Pagerank` in order to support the precise results to users. We test the framework by using `TREC WT10g` test collection. The experimental result shows that the new method gives a better effect. But we find that the effect is not so distinct, we want to consider the in-link of every page in the future. In other side, we should change the value of α, which affects the final result of page order.

However, in order to satisfy the users' actual information need, it is more important to find relevant Web page from the enormous web space. Therefore, we plan to address the technique to provide users with personalized information.

Acknowledgements

This work was supported by project *2004F06* and *2005F08*, Research of Nature Science of *Shaanxi Province, P.R.China*.

References

1. Raghavan, S., Garcia-Molina, H.: Complex queries over web repositories. In: Proceedings of 29th International Conference on Very Large Data Bases (VLDB 2003), September 9-12, Berlin, Germany, Morgan Kaufmann (2003) 33–44
2. Delort, J.Y., Bouchon-Meunier, B., Rifqi, M.: Enhanced web document summarization using hyperlinks. In: Proceedings of the 14th ACM conference on Hypertext and hypermedia(HYPERTEXT 2003), ACM Press (2003) 208–215
3. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project (1998)
4. Bianchini, M., Gori, M., Scarselli, F.: Inside pagerank. ACM Transactions on Internet Technology **5**(1) (2005) 92–128
5. Eiron, N., McCurley, K.S., Tomlin, J.A.: Ranking the web frontier. In: Proceedings of the 13th international conference on WWW2004, ACM Press (2004) 309–318

6. Boldi, P., Santini, M., Vigna, S.: Pagerank as a function of the damping factor. In: Proceedings of the 14th international conference on World Wide Web(WWW 2005), ACM Press (2005) 557–566
7. Sugiyama, K., Hatano, K., Yoshikawa, M., Uemura, S.: Improvement in tf-idf scheme for web peges based on the contents of their hyperlinked neighboring pages. Syst. Comput. Japan **36**(14) (2005) 56–68

A Fast Algorithm for Words Reordering Based on Language Model

Theologos Athanaselis, Stelios Bakamidis, and Ioannis Dologlou

Institute for Language and Speech Processing
Artemidos 6 and Epidavrou, GR-15125,
Maroussi, Greece
Tel.: +302106875300, Fax: +302106854270
{tathana, bakam, ydol}@ilsp.gr
http://www.ilsp.gr

Abstract. What appears to be given in all languages is that words can not be randomly ordered in sentences, but that they must be arranged in certain ways, both globally and locally. The "scrambled" words into a sentence cause a meaningless sentence. Although the use of manually collected grammatical rules can boost the performance of grammar checker in word order diagnosis, the repairing task is still very difficult. This work proposes a method for repairing word order errors in English sentences by reordering words in a sentence and choosing the version that maximizes the number of trigram hits according to a language model. The novelty of this method concerns the use of a permutations' filtering approach in order to reduce the search space among the possible sentences with reordered words. The filtering method is based on bigrams' probabilities. In this work the search space is further reduced using a threshold over bigrams' probabilities. The experimental results show that more than 95% of the test sentences can be repaired using this technique. The comparative advantage of this method is that it is not restricted into a specific set of words, and avoids the laborious and costly process of collecting word order errors for creating error patterns. Unlike most of the approaches, the proposed method is applicable to any language (language models can be simply computed in any language) and does not work only with a specific set of words. The use of parser and/or tagger is not necessary.

1 Introduction

Automatic grammar checking is traditionally done by manually written rules, constructed by computer linguists. Methods for detecting grammatical errors without manually constructed rules have been presented before. Atwell (1987) uses the probabilities in a statistical part-of the speech tagger, detecting errors as low probability part of speech sequences. Golding (1995) showed how methods used for decision lists and Bayesian classifiers could be adapted to detect errors resulting from common spelling confusions among sets such as "there", "their" and "they're". He extracted contexts from correct usage of each confusable word in a training corpus and then identified a new occurrence as an error when it matched the wrong context.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 943 – 951, 2006.

Chodorow and Leacock (2000) suggested an unsupervised method for detecting grammatical errors by inferring negative evidence from edited textual corpora. Heift (1998, 2001) released the German Tutor, an intelligent language tutoring system where word order errors are diagnosed by string comparison of base lexical forms. Bigert and Knutsson (2002) presented how a new text is compared to known correct text and deviations from the norm are flagged as suspected errors. Sjobergh (2005) introduced a method of grammar errors recognition by adding errors to a lot of (mostly error free) unannotated text and by using a machine learning algorithm.

Unlike most of the approaches, the proposed method is applicable to any language (language models can be computed in any language) and does not work only with a specific set of words. The use of parser and/or tagger is not necessary. Also, it does not need a manual collection of written rules since they are outlined by the statistical language model.

The paper is organized as follows: the architecture of the entire system and a description of each component follow in section 2. The language model is described in section 3. The 4th section shows how permutations are filtered by the proposed method. The 5th section specifies the method that is used for searching valid trigrams in a sentence. The results of using WSJ experimental scheme are discussed in section 6. Finally, the concluding remarks are made in section 7.

2 System's Architecture

This work presents a new method for detecting and repairing sentences with word order errors that is based on the statistical language model (N-grams). It is straight forward that the best way for reconstructing a sentence with word order errors is to reorder the words. However, the question is how it can be achieved without knowing the attribute of each word. Many techniques have been developed in the past to cope with this problem using a grammar parser and rules. However, the success rates reported in the literature are in fact low. A way for reordering the words is to use all the possible permutations. The crucial drawback of this approach is that given a sentence with length N words the number of all permutations is N!. This number is very large and seems to be restrictive for further processing. The novelty of the proposed method concerns the use of a technique for filtering the initial number of permutations. The process of repairing sentences with word–order errors incorporates the followings tools:

- a simple, and efficient confusion matrix technique
- and language model's trigrams and bigrams.

Consequently, the correctness of each sentence depends on the number of valid trigrams. Therefore, this method evaluates the correctness of each sentence after filtering, and provides as a result, a sentence with the same words but in correct order (Figure 1).

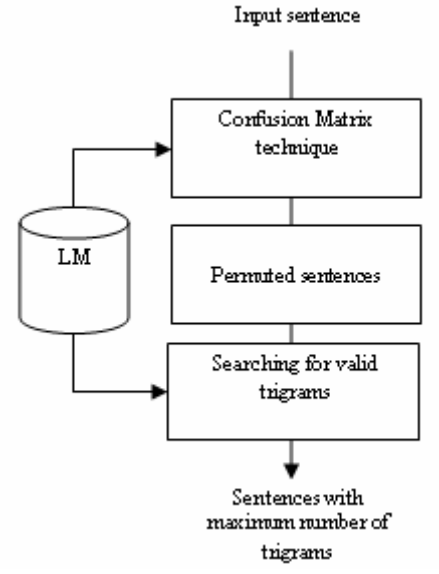


Fig. 1. System's architecture

3 Language Model

The language model (LM) that is used subsequently is the standard statistical N-grams (Young, 1996). The N-grams provide an estimate of $P(W)$, the probability of observed word sequence W. Assuming that the probability of a given word in an utterance depends on the finite number of preceding words, the probability of N-word string can be written as:

$$P(W) = \prod_{i=1}^{N} P(w_i \mid w_{i-1}, w_{i-2}, ..., w_{i-(N-1)}) \quad (1)$$

One major problem with standard N-gram models is that they must be trained from some corpus, and because any particular training corpus is finite, some perfectly acceptable N-grams are bound to be missing from it. That is, the N-gram matrix for any given training corpus is sparse; it is bound to have a very large number of cases of putative "zero probability N-grams" that should have some non zero probability. Some part of this problem is endemic to N-grams; since they can not use long distance context, they always tend to underestimate the probability of strings that happen no tot have occurred nearby in their training corpus. There are some techniques that can be used in order to assign a non zero probability to these zero probability N-grams. In this work, the language model has been trained using BNC and consists of trigrams with Good-Turing discounting (Good, 1953) and Katz back off (Katz, 1987) for smoothing. BNC contains about 6.25M sentences and 100 million words. The figure below depicts the number of bigrams of the LM (Language Model) with respect to their logarithmic probabilities. The 80% of the LM's bigrams are between -5,2 and -1,6.

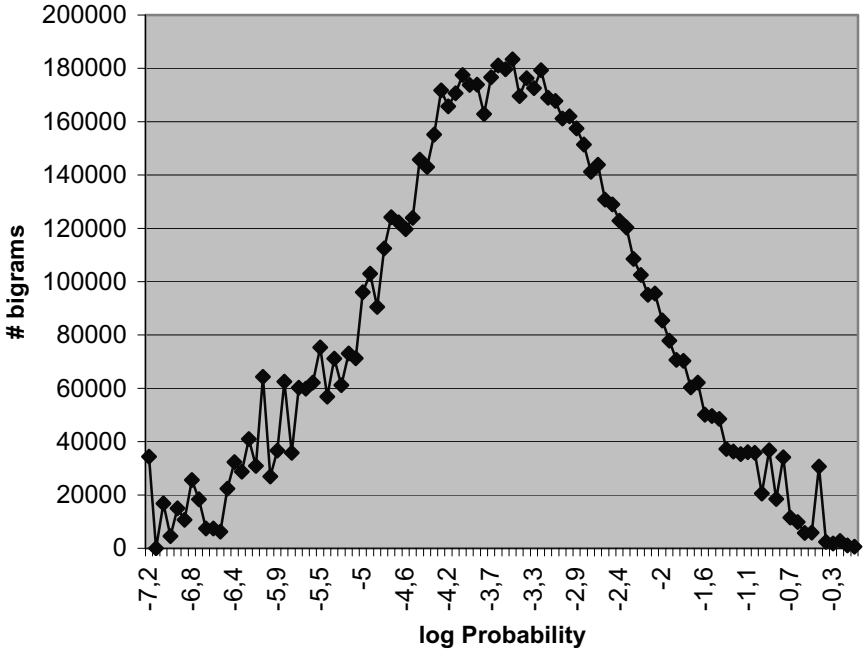


Fig. 2. The bigrams' distribution with regard to their log probabilities

4 Filtering Permutations

Considering that an ungrammatical sentence includes the correct words but in wrong order, it is plausible that generating all the permuted sentences (words reordering) one of them will be the correct sentence (words in correct order). The question here is how feasible is to deal with all the permutations for sentences with large number of words. Therefore, a filtering process of all possible permutations is necessary. The filtering involves the construction of a confusion matrix NxN in order to extract possible permuted sentences.

Given a sentence $a = [w[0], w[1],...w[n-1], w[n]]$ with N words, a confusion matrix $A \in R^{NXN}$ can be constructed.

The size of the matrix depends on the length of the sentence. The objective of this confusion matrix is to extract the valid bigrams according to the language model. The element $P[i, j]$ indicates the validness of each pair of words $(w[i]w[j])$ according to the list of language model's bigrams. If a pair of two words $(w[i]w[j])$ cannot be found in the list of language model bigrams then the corresponding $P[i, j]$ is taken equal to 0 otherwise it is equal to one. Hereafter, the pair of words with $P[i, j]$ equals to 1 is called as valid bigram. Note that, the number of valid bigrams is M lower

Table 1. The construction of a NxN confusion matrix, for the sentence $a = [w[0], w[1], ...w[n-1], w[n]]$

WORD	**w[0]**	**w[1]**	**…….**	**w[n]**
w[0]	P[0,0]	P[1,0]	…….	P[n,0]
w[1]	P[0,1]	P[1,1]	…….	P[n,1]
.	.	.		.
.	.	.		.
.	.	.		.
w[n]	P[0,n]	P[1,n]	…….	P[n,n]

than the size of the confusion matrix which is N^2, since all possible pairs of words are not valid according to the language model. In order to generate permuted sentences using the valid bigrams all the possible words' sequence must be found. This is the search problem and its solution is the domain of this filtering process.

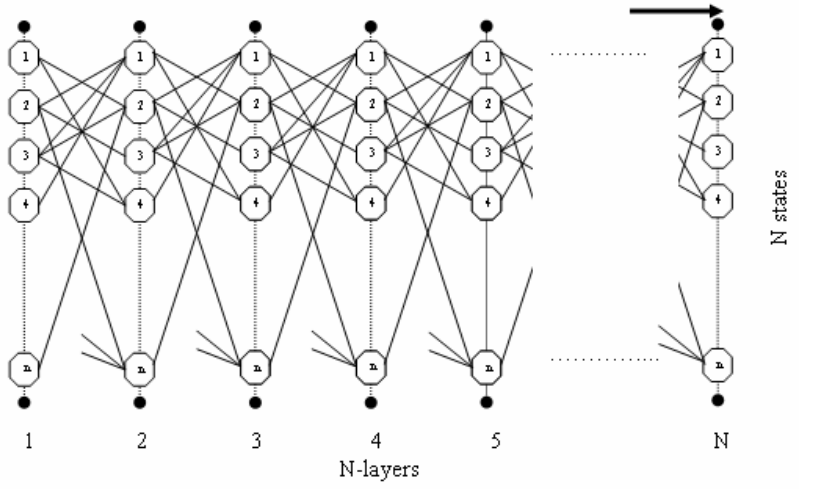


Fig. 3. Illustration of the lattice with N-layers and N states

As with all the search problems there are many approaches. In this paper a left to right approach is used. To understand how it works the permutation filtering process, imagine a network of N layers with N states. The factor N concerns the number of sentence's words. Each layer corresponds to a position in the sentence. Each state is a possible word. All the states on layer 1 are then connected to all possible states on the second layer and so on according to the language model. The connection between two states (i, j) of neighboring layers $(N-1, N)$ exists when the bigram $(w[i]w[j])$

is valid. This network effectively visualizes the algorithm to obtain the permutations. Starting from any state in layer 1 and moving forward through all the available connections to the *N*-th layer of the network, all the possible permutations can be obtained. No state should be "visited" twice in this movement.

5 Searching Valid Trigrams

The prime function of this approach is to decompose any input sentence into a set of trigrams. To do so, a block of words is selected. In order to extract the trigrams of the input sentence, the size of each block is typically set to 3 words, and blocks are normally overlapped by two words. Therefore, an input sentence of length N, includes N-2 trigrams.

The second step of this method involves the search for valid trigrams for each sentence. A probability is assigned to a valid trigram, which is derived by the frequency of its occurrences in the corpus.

In the third step of this method the number of valid trigrams per each permuted sentence is calculated. Considering that the sentence with no word-order errors has the maximum number of valid trigrams, it is expected that any other permuted sentence will have less valid trigrams. Although some of the sentence's trigrams may be typically correct, it is possible not to be included into the list of LM's trigrams. The plethora of LM's trigrams relies on the quality of corpus. The lack of these valid trigrams does not affect the performance of the method since the corresponding trigrams of the permuted sentence will not be included into LM as well. The criterion for ranking all the permuted sentences is the number of valid trigrams. The system provides as an output, a sentence with the maximum number of valid trigrams. In case where two or more sentences have the same number of valid trigrams a new distance metric should be defined. This distance metric is based on the total log probability of the sentence's trigrams. The total log probability is computed by adding the log probability of each valid trigram, whereas the probability of non valid trigrams is assigned to -100000. Therefore the sentence with the maximum total log probability is the system's response.

6 Experimentation

6.1 Experimental Scheme

The experimentation involves a test set of 500 sentences, with 4518 words. Test sentences have been selected randomly from WSJ (Wall Street Journal) corpus. They have variable length with minimum 7 words and maximum 12 words. The 90% of the test words belong to the BNC vocabulary (training data). For experimental purposes our test set consists of sentences with no word order errors and the system's response incorporates 10-best sentences. The goal of this experimentation is to show that the input sentence is included into the 10-best sentences. Note that the test sentences are not included into the training set of the statistical language model that is used as tool for the proposed method.

6.2 Experimental Results

6.2.1 WSJ Test Cases

Figure 4 shows the repairing results using the test sentences. This figure depicts the capability of the system to give as output the correct sentences in the 10-best list. The x-axis corresponds to the place of the correct sentence into this list. The last position (11) indicates that the correct sentence is out of this list.

The findings from the experimentation show that 455 sentences (91% in total) have been repaired using the proposed method (True Corrections). On the other hand, the result for 45 sentences (9% in total) was false (False Corrections). In case of "False Corrections" the system's response does not include the correct sentence into the N-best. The incorrect output of the system can be explained considering that some words are not included into the BNC vocabulary, hence some of the sentences' trigrams are considered as invalid.

It is obvious that the system's performance for detecting and repairing method of ill-formed sentences with word order errors depends mainly on the quality of the corpus. The high success rate of the system is achieved using the grammatically and syntactically correct sentences of BNC.

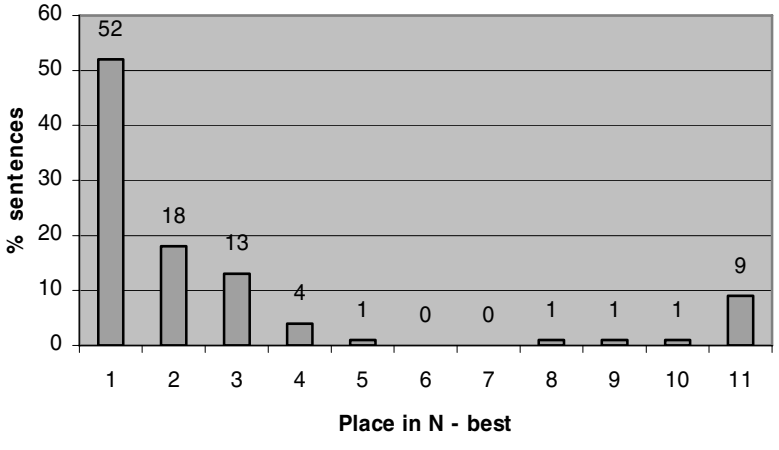


Fig. 4. The percentage of test sentences in different places into the N-best list (N=10)

6.2.2 Reducing Search Space

The figure below depicts the differences in the number of permutations for sentences with length from 7 to 12 words. The point is that the number of permutations that are extracted with the filtering process is significantly lower than the corresponding value without filtering. For sentences with length up to 6 words, the number of permutations is slightly lower when the filtering process is used, while for sentences with length greater than 7 words the filtering process provides a drastical reduction of permutations. It is obvious that the performance of filtering process depends mainly on the

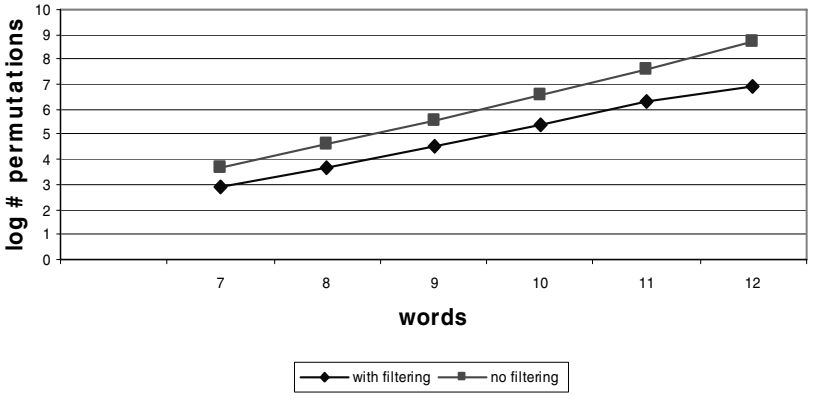


Fig. 5. The logarithmic number of permutations with and without filtering for TOEFL's sentences with 7 up to 12 words

number of valid bigrams. This implies that the language model's reliability affects the outcome of the system and especially of the filtering process.

7 Conclusions

The findings show that most of the sentences can be repaired by this method independently from the sentence's length and the type of word order errors. The major advantage of this technique concerns the application of novel fast algorithm in reducing permutations. The results show that the gain factor for permutations in case of sentences with 12 words is 35dB. With no filtering the number of permutations is 479001600 while with the confusion matrix this quantity decreases drastically to 8790541. The proposed method is effective in repairing erroneous sentences. Therefore the method can be adopted by a grammar checker as a word order repairing tool. The necessity of the grammar checkers in educational purposes and e-learning is more than evident.

By the permutation's filtering process, the system takes advantage of better performance, rapid response and smaller computational space. A comparative advantage of this method is that avoids the laborious and costly process of collecting word order errors for creating error patterns. One of the key questions for further research is whether the use of language model can correct other grammatical errors such as subject- verb disagreement, and if it is possible a further reduction in permutations using probabilities thresholds.

Acknowledgments

The authors would like to thank Mr Kostantinos Mamouras for his programming skills and the insightful comments.

References

1. Atwell, E.S., How to detect grammatical errors in a text without parsing it. In Proceedings of the 3rd EACL, (1987) 38–45
2. Bigert, J., Knutsson. O., Robust error detection: A hybrid approach combining unsupervised error detection and linguistic knowledge. In Proceedings of Robust Methods in Analysis of Natural language Data, (ROMAND 2002), (2002) 10–19
3. Chodorow M., Leacock C. An unsupervised method for detecting grammatical errors. In Proceedings of NAACL'00, (2000) 140–147
4. Feyton, C. M., Teaching ESL/EFL with the internet. Merill Prentice- Hall, (2002)
5. Folse, K.S., Intermediate TOEFL Test Practices (rev. ed.). Ann Arbor, MI: The University of Michigan Press., (1997)
6. Good, I.J., The population frequencies of species and the estimation of population parameters. Biometrika, 40(3 and 4): (1953) 237-264,
7. Golding, A. A., Bayesian hybrid for context-sensitive spelling correction. Proceedings of the 3rd Workshop on Very Large Corpora, (1995) 39-53
8. Hawkins, J. A., A Performance Theory of Order and Constituency. Cambridge, Cambridge University Press, (1994)
9. Heift, T., Intelligent Language Tutoring Systems for Grammar Practice. Zeitschrift für Interkulturellen Fremdsprachenunterricht (Online), 6 (2), (2001) 15 pp
10. Katz S.M., Estimation of probabilities from sparse data for the language model component of a speech recogniser. IEEE Transactions on Acoustics, Speech and Signal Processing, 35(3): (1987) 400-401,
11. Sjöbergh, J., Chunking: an unsupervised method to find errors in text, Proceedings of the 15th Nordic Conference of Computational Linguistics, NODALIDA (2005)
12. Young, S.J., Large Vocabulary Continuous Speech Recognition, IEEE Signal Processing Magazine 13, (5), (1996) 45-57,

Phonetic Feature Discovery in Speech Using *Snap-Drift* Learning

Sin Wee Lee and Dominic Palmer-Brown

Innovative Informatics Research Group
University of East London, Essex, Rm8 2AS, UK
{SinWee, D.Palmer-Brown}@uel.ac.uk
http://www.uel.ac.uk/scot/ii/index.htm

Abstract. This paper presents a new application of the *snap-drift* algorithm [1]: feature discovery and clustering of speech waveforms from non-stammering and stammering speakers. The learning algorithm is an unsupervised version of snap-drift which employs the complementary concepts of fast, minimalist learning (*snap*) & slow *drift* (towards the input pattern) learning. The *Snap-Drift* Neural Network (SDNN) is toggled between snap and drift modes on successive epochs. The speech waveforms are drawn from a phonetically annotated corpus, which facilitates phonetic interpretation of the classes of patterns discovered by the SDNN.

1 Introduction

Stuttering (stammering) is a highly variable condition which occurs across ages and cultures. There is a lack of consensus in establishing the criteria for a definition. Finding a way of identifying exactly what phonetic characteristics are associated with stammering, as opposed to non-stammering speech, has proved elusive. Perceptual analysis is known to be compromised by its subjectivity [2], [3]. In contrast, a correlative data analysis to characterise the acoustic properties of stammering is realisable. There are four classes of sound pressure wave that form the acoustic structure of utterances [4]: Periodic 'voice': regular repeating fluctuations produced by vocal fold vibration; Aperiodic 'noise': ongoing irregular fluctuations in voiceless fricatives; Transient 'burst': brief irregular fluctuations as in voiceless plosives; or Silent: no acoustic energy is emitted. The speech sounds used in human languages are made up of combinations of the four categories.

The *snap-drift* learning algorithm first emerged as an attempt to overcome the limitations of ART learning in non-stationary environments where self-organisation needs to take account of periodic or occasional performance feedback. Since then, the *snap-drift* algorithm has proved invaluable for continuous learning in several applications.

The *reinforcement* versions [5], [6] of *snap-drift* are used in the classification of user requests in an active computer network simulation environment whereby the system is able to discover alternative solutions in response to varying performance requirements. Furthermore, the *unsupervised snap-drift* algorithm, without any form of reinforcement, has been used in the analysis and interpretation of data representing

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 952 – 962, 2006.

interactions between trainee network managers and a simulated network management system [7]. New patterns of the user behaviour were discovered.

The further exploration of *snap-drift*, in the form of a classifier [8] has been used in attempting to discover and recognize phrases extracted from Lancaster Parsed Corpus (LPC) [9]. Comparisons carried out between *snap-drift* and MLP with back-propagation, show that the former is faster and just as effective.

This paper describes the further exploration of snap-drift, in unsupervised form, in attempting to discover the defining and unique millisecond features in the speech patterns, which will be used to help understand the language learning of non-stammering and stammering speakers.

2 The Snap-Drift Neural Network (SDNN) Architecture

The modular neural network modified from the Performance-guided Adaptive Resonance Theory (P-ART) network, first introduced by Lee & Palmer-Brown [1] is shown in Fig. 1.

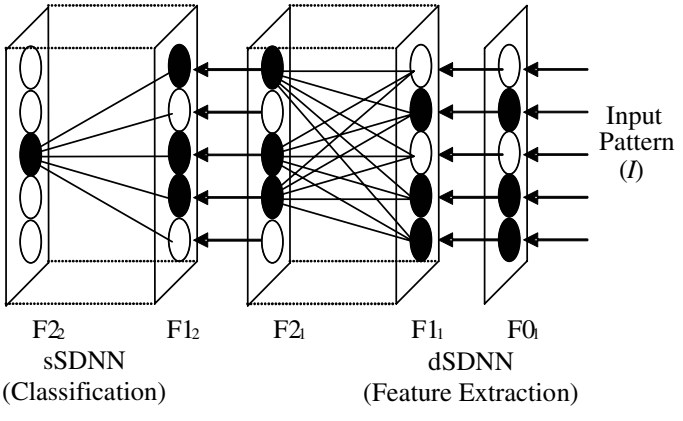


Fig. 1. SDNN Architecture

On presentation of an input pattern at the input layer $F0_1$, dSDNN will learn to group the input patterns according to their general features. In this case, 10 $F2_1$ nodes, whose weight prototypes best match the current input pattern, are used as the input data to the sSDNN module for feature classification. In both of the modules, the standard matching and reset mechanism of ART [5], [6] is discarded. Instead, in the dSDNN module, the output nodes with the highest net input are *always* accepted as winners. In the sSDNN module, a quality assurance threshold is introduced. If the net input of a sSDNN node is above the threshold, the output node is accepted as the winner, otherwise a new uncommitted output node will be selected as the new winner and initialised with the current input pattern.

In this version of SDNN we introduce *weight re-initialisation*. The main purpose of weight re-initialisation is to enable unused output nodes to be reinstated into the

competition for winning nodes. Weight re-initialization is invoked after many epochs since the SDNN must first allow input patterns to settle into their categories. After a duration defined by a certain number of input patterns, called a learning era (an era is a number of epochs), the weights of nodes unused during the preceding era will be re-initialised to enable them to participate again in the competition for the best winning nodes. In effect, reinitialisation is a neuron pruning algorithm. It removes weight vectors that are redundant.

The following is a summary of the steps that occur in SDNN:

Step 1: Initialise parameters: (α = 1, σ = 0), era = 2000
Step 2: For each epoch (t)
  Test: **Weights re-initialization condition**
  For each input pattern
  Step 2.1: Find the D (D = 10) winning nodes at $F2_1$ with the largest net input
  Step 2.2: Inhibit the $F2_1$ node for weights re-initialization
  Step 2.3: Weights of dSDNN adapted according to the alternative learning procedure: (α,σ) becomes Inverse(α,σ) after every successive epoch
Step 3: Process the output pattern of $F2_1$ as input pattern of $F1_2$
  Step 3.1: Find the node at $F1_2$ with the largest net input
  Step 3.2: Test the threshold condition:
    IF (the net input of the node is greater than the threshold)
    THEN
    Weights of the sSDNN output node adapted according to the alternative learning procedure: (α,σ) becomes inverse (α,σ) after every successive epoch
    ELSE
    An uncommitted sSDNN output node is selected and its weights are adapted according to the alternative learning procedure: (α,σ) becomes Inverse(α,σ) after every successive epoch

Weights re-initialization condition:
After 'era' input patterns
IF ($F2_1$ node not used for the past era input presentations) **THEN**
Re-initialize the $F2_1$ node with randomly selected input pattern
Inhibit the $F2_1$ node for weights re-initialization for the next era input pattern presentation
ELSE
No action taken.

3 The Snap-Drift Algorithm

The learning algorithm combines a modified form of Adaptive Resonance Theory (*snap)* [10] and Learning Vector Quantisation *(drift)* [11]. In general terms, the snap-drift algorithm can be stated as:

$$\textit{Snap-drift} = \alpha(\text{Fast\_Learning\_ART}) + \sigma(\text{LVQ}) \tag{1}$$

The top-down learning of both of the modules in the neural system is as follows:

$$w_{Ji}^{(new)} = \alpha(I \cap w_{Ji}^{(old)}) + \sigma(w_{Ji}^{(old)} + \beta\ (I - w_{Ji}^{(old)})) \tag{2}$$

where w_{Ji} = top-down weights vectors; I = binary input vectors, and β = the drift speed constant = 0.5.

In successive learning epochs, the learning is toggled between the two modes of learning. When $\alpha = 1$, fast, minimalist (snap) learning is invoked, causing the top-down weights to reach their new asymptote on each input presentation. (2) is simplified as:

$$w_{Ji}^{(new)} = I \cap w_{Ji}^{(old)} \tag{3}$$

This learns sub-features of patterns. In contrast, when $\sigma = 1$, (2) simplifies to:

$$w_{Ji}^{(new)} = w_{Ji}^{(old)} + \beta\ (I - w_{Ji}^{(old)}) \tag{4}$$

which causes a simple form of clustering at a speed determined by β.

The bottom-up learning of the neural system is a normalised version of the top-down learning.

$$w_{iJ}^{(new)} = w_{Ji}^{(new)} / |\ w_{Ji}^{(new)}| \tag{5}$$

where $w_{iJ}^{(new)}$= top-down weights of the network after learning.

In SDNN, as described in section 2, snap-drift is toggled between snap and drift on each successive epoch. The effect of this is to capture the strongest clusters (holistic features), sub-features, and combinations of the two.

4 Simulations

The *snap-drift* algorithm is used for learning and discovering the features embedded in the utterances of two speaker groups, non-stammering and stammering. Before any simulations, pre-processing of the utterances is completed. In this research, each point of a speech utterance waveform collected represents 1 ms of speech data. In this research, in order to analyze and recognise the acoustic properties of the speaker with sufficient precision, each utterance is sampled every 10 points for a total of 1000 points, which represents about 1 second of speech information. This is considered sufficient by a phonetics expert. Figure 2 shows the example of sampled utterance used in the simulations. Each of the sampled waveforms is used to generate a number of input patterns for SDNN. The input patterns are generated using a sliding window of size 100 samples points. The sliding window is shifted to the right by 25 sample points to create a new input. This provides some overlapping of features among the

input patterns. Then, each input pattern is converted into a 1400 bit coarse coded binary pattern. 5 utterances are used from 2 speakers, 3 utterances from the non-stammering speaker and 2 from the stammering speaker. Table 1 shows the range and properties of the input set, making the total number of input patterns, 1873 input vectors. These test input patterns are presented in sequence to SDNN. The number of input patterns for each speaker varies because:

1. Each speaker is asked to speak using different types of statements.
2. Non-stammering speaker will produce more fluent speech utterances with shorter or no delay between phrases.
3. Stammering speakers always produce longer utterances due to the delay in the voiceless fricative.

The input patterns, which are also quite noisy, provide a real world test for unsupervised SDNN as a feature discovery and classification system.

For SDNN to act as a viable classifier, and to demonstrate the utility of the features it acquires, it should be able to estimate or predict whether a speaker in a real-time scenario is non-stammering or stammering when a speech utterance is fed into the system. An estimation will be made of how long it takes to be certain that a speaker is non-stammering or stammering.

Table 1. Range and properties of the input set

Speaker group	Total number of Inputs
Non-stammering	256
Stammering	644
Non-stammering	162
Stammering	467
Non-stammering	229

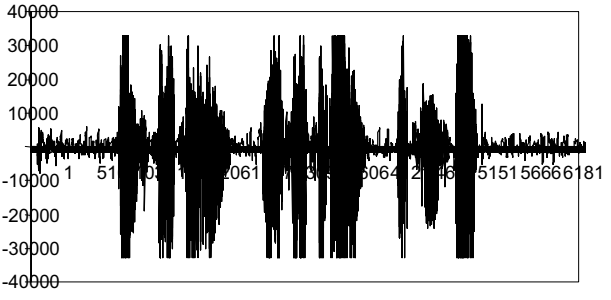


Fig. 2. Example utterance waveform used in simulation

4.1 Results

The results are presented in Table 2 to Table 4; each of the tables shows the example category types formed by the SDNN network with their acoustic properties. The acoustic properties record is obtained from a phonetics expert's annotation of the speech

Table 2. Accoustic properties of example category type 1 (Stammering)

Input	Speaker group	Silent	Periodic	Aperiodic	Transient
195	Non-stammering	✓		✓	✓
211	Non-stammering	✓			✓
377, 456	Stammering		✓		
432, 68	Stammering		✓		✓
473, 485, 575, 585	Stammering	✓			
570	Stammering	✓	✓		
595	Stammering	✓			✓
609	Stammering	✓		✓	

Table 3. Accoustic properties of example category type 2 (Non-Stammering)

Input	Speaker group	Silent	Periodic	Aperiodic	Transient
21, 34, 3699, 142, 175	Non-stammering		✓	✓	
27, 32, 231, 253	Non-stammering		✓		
38, 187	Non-stammering			✓	
48	Non-stammering				✓
56	Non-stammering	✓			✓
200	Non-stammering	✓			
304	Stammering	✓	✓		✓
310	Stammering		✓	✓	

Table 4. Accoustic properties of example category type 3 (Mixture of both type of speakers)

Input	Speaker group	Silent	Periodic	Aperiodic	Transient
45, 108	Non-stammering		✓	✓	
165	Non-stammering	✓			
131, 135	Non-stammering		✓		✓
204, 123	Non-stammering		✓		
283, 504	Stammering		✓		✓
304, 442	Stammering	✓	✓		✓
615, 565, 370, 457	Stammering		✓		
546, 547	Stammering	✓			

waveform corpus. Each of the sampled sequence of the speech utterance is identified with one or more acoustic properties: *Silent, Periodic, Aperiodic* and *Transient.*

By looking at the tables, it is clear that the SDNN has categorised the input patterns into 3 distinctive types, stammering speech, non-stammering speech, and a category type with a mixture of the two speaker types. The three category types were identified since they corresponded to different non-overlapping sets of sSDNN output nodes.

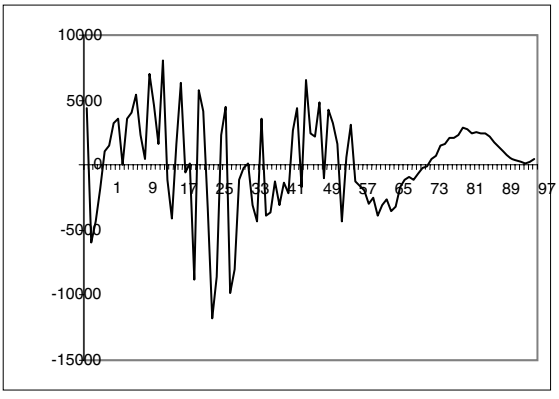


Fig. 3. Example input waveform for category type 1 (Input 377)

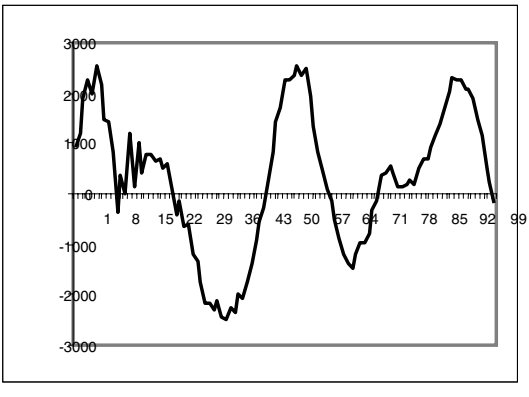


Fig. 4. Example input waveform for category type 1 (Input 595)

Fig. 3 - 5 show the example input waveforms being grouped into the same category, in this case example category type 1 (Stammering). By comparing these waveforms, the similarities can be easily identified. In order to understand the learned features of the speech utterances, a comparison of the input patterns of the system and the learned weight templates is performed.

The input patterns received by the SDNN are binary coarse coded representations of the fragments of speech input utterances, such as those shown in Fig. 3 – 5. Each point in the speech input is represented by a 14 bit binary representation. So, the weights learned are the results of processing these binary input patterns. As a means of visualization, the weights learned are thresholded as a first order approximation to produce a binary representation of the weights learned. Then, the 14 bit coarse binary representation of the weights learned are decoded to show the actual waveform features that have been acquired from the original waveforms.

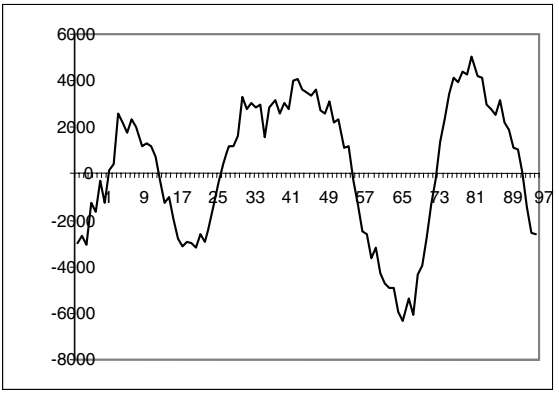


Fig. 5. Example input waveform for category type 1 (Input 456)

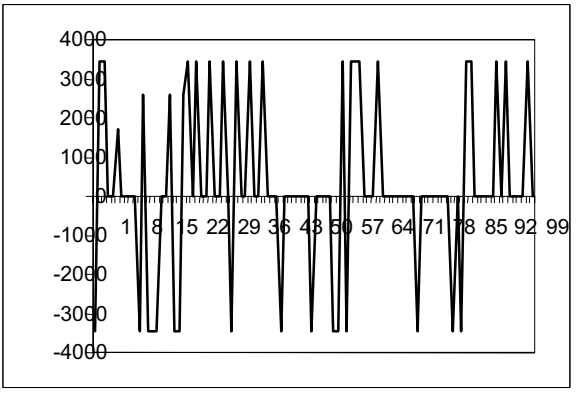


Fig. 6. Example weights learned for category type 1 (Winning node 42)

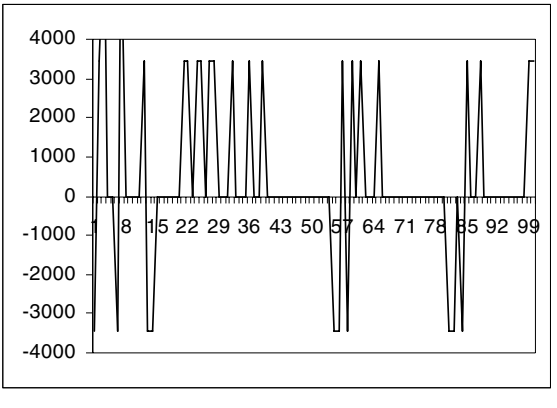


Fig. 7. Example weights learned for category type 1 (Winning node 13)

Fig. 6 and Fig. 7 show the weights learned. Although these weights graphs are drawn using approximation for visualization, the figures clearly show that system has learned the features in the input patterns of the categories. In fig. 6 and 7, the graphs show a noisy sinusoid of about 3 Hz. By comparing with the original waveforms, it has clearly shown that what these waveforms have in common is a sinusoid of approximately 3Hz. The phonetics expert has identified that these parts of the utterances are often associated with silence or pauses or gaps between words where there is some sound perhaps but no clear articulation. This is indeed known to be the case for stammerers.

5 Unique Sequences and Classifications

As mentioned, during each learning epoch, the speech utterances are fed into the system in sequence, one speaker utterance at a time. In order to do the analysis and thus determine the time it takes to identify the speaker type, one epoch after convergence is randomly selected. By randomly selecting one sequence of sSDNN winning nodes to start with, the whole epoch is examined to find any repeated occurrences of the sequence. These repeated occurrences of winning nodes sequences are called *unique sequences* if they are unique to only stammering or non-stammering speakers. Then, the speaker input utterances which caused the *unique sequence*, is examined. With this method of analysis, the length of unique sequence of winning nodes which only occurred in a particular group of speakers, either stammering or non-stammering, will determine the time the system takes to be certain of the speaker group for a particular speech utterance.

Table 4 shows the sequence occurrence of winning nodes for non-stammering or stammering group input patterns. The sequences for analysis are randomly selected. In the table, most of the sequences with the length less than 3 tend to have a mixture of occurrence of both types of speaker groups. By increasing the length of the sequence, some form of bias arises. With the sequence length of more than 5 winning nodes, these sequences only occur in one of the speaker types, either non-stammering or stammering. For example, the sequence {45, 52, 43, 19, 65} only exists in the speech input of the stammering speaker. Obviously, this sequence is *unique* to the stammering speaker. By plotting the average ratio of the speaker type over the sequence length, the length of the sequence which can be labelled as unique can be identified. This is illustrated in Fig. 8. In fig. 8, the average ratio of the speaker group for sequence length of 5 and 6 is the lowest. With this number of randomly selected sequences for consideration, it confidently shows that input patterns for particular speaker groups can be identified when a unique sequence, with the length of 5 winning nodes is used for analysis.

By identifying this *unique sequence*; we mean SDNN is capable of identifying the speaker group of input patterns after system convergence is achieved. As mentioned in section IV, each input pattern roughly represents about 1 second of speech information, thus, SDNN is capable of distinguishing the type of speaker by analysis of about 5 seconds of speech, which is analogous to the a person identifying a speaker as

stammering or non-stammering after hearing several words. Since not all words are stammered by stammerers, this figure is also of the order of 5 seconds of speech for humans.

Thus, SDNN has shown the capability of a classifier, in this case, categorizing the input patterns according to their features and classifying and estimating the time it takes to be certain that a speaker is non-stammering or stammering by using unique sequences of sSDNN winning nodes.

Table 5. Randomly selected sequence occurrence of winning nodes for non-stammering / stammering group input patterns

Sequence	No. of Occurrences	Non-stammering	Stammering
63, 65	21	13	9
1, 36, 31	25	9	6
7,5,3	19	11	8
45,52,43	15	6	9
12,23,34,34	11	6	7
7,5,3,54,39	4	4	0
42,34,46,10,59	3	3	0
45,52,43,19,65	7	0	7
7,7,2,6,49	3	3	0
39,36,56,16,32	4	0	4
6,32,40,4,23,58	6	1	5
69,68,56,68,69	3	0	3
54,69,55,11,46,50	3	0	3
11,63,45,37,56,68	4	4	0
6,32,46,23,4,33	2	0	2

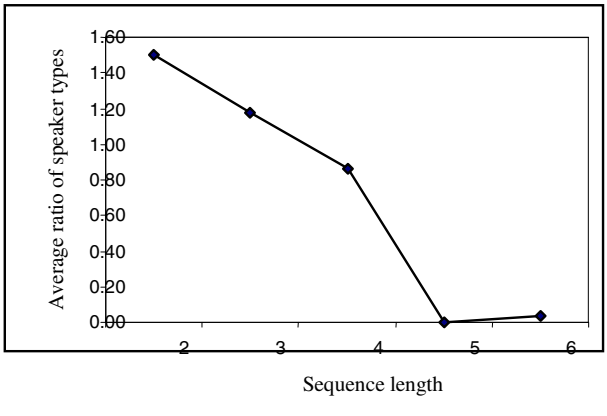


Fig. 8. The average ratio of the speaker type over the length of the winning node sequence

6 Conclusion

This paper presents the new application of feature discovery in phonetics speech using the *snap-drift* algorithm. It also gives the opportunity to test the performance of SDNN without a performance feedback in a purely unsupervised mode. SDNN categorizes the input patterns according to their general and distinct features. By examining the phonetic and waveform properties of the input patterns in each of the categories formed, it has been shown that without any performance feedback, the SDNN modules group the input patterns sensibly and extract properties which are general between non-stammering and stammering speech, as well as distinct features within each of the utterance groups, thus supporting classification.

References

[1] Lee, S. W., Palmer-Brown, D., Roadknight, C. M.: Performance-guided Neural Network for Rapidly Self-Organising Active Network Management. Neurocomputing, Vol. 61C (2004) 5 – 20.
[2] Aylett, M., Turk, A.: Vowel Quality in Spontaneous Speech: What makes a good vowel. Proc. of Int. Conf. of Spoken Language Processing. Sydney, Australia.
[3] Klatt, D. H.: Review of Text-to-Speech Conversion for English. Online collection (1987)
[4] Ladefoged, P.: A Course in Phonetics. 4th ed., Boston, Heinle & Heinle (2001)
[5] Lee, S. W., Palmer-Brown, D., Tepper, J., Roadknight, C. M.: Snap-Drift: Real-time Performance-guided Learning. Proc. of IJCNN, Portland, Oregon, Vo1. 2 (2003) 1412 – 1416.
[6] Lee, S. W., Palmer-Brown, D., Roadknight, C. M.: Reinforced Snap-Drift Learning for Proxylet Selection in Active Computer Networks. Proc. of IJCNN, Budapest, Hungary, Vo1. 2 (2004) 1545 – 1550.
[7] Donelan, H., Pattinson, C., Palmer-Brown, D., Lee, S. W.: The Analysis of Network Manager's Behaviour using a Self-Organising Neural Networks. Proc. of ESM, Magdeburg, Germany, (2004) 111 – 116..
[8] Lee, S. W., Palmer-Brown, D.: Phrase Recognition using Snap-Drift Learning Algorithm. Proc. of IJCNN, Montreal, Canada (2005).
[9] Garside, R., Leech, G., Varadi, T.: Manual of Information to Accompany the Lancaster Parsed Corpus: Department of English, University of Oslo (1987).
[10] Carpenter, G. A., Grossberg, S.: A Massively Parallel Architecture for a Self-Organising Neural Pattern Recognition Machine. Com. Vision, Graphics and Image Proc., Vol. 37 (1987) 54-115.
[11] Kohonen, T.: Improved Versions of Learning Vector Quantization. Proc. of IJCNN, Vol. 1 (1990) 545 – 550.

A New Neuro-Dominance Rule for Single Machine Tardiness Problem with Unequal Release Dates

Tarık Çakar

Sakarya University Engineering Faculty
Department of Industrial Engineering
54187 Adapazarı – Turkey

Abstract. We present a neuro-dominance rule for single machine total weighted tardiness problem with unequal release dates. To obtain the neuro-dominance rule (NDR), backpropagation artificial neural network (BPANN) has been trained using 10000 data and also tested using 10000 another data. The proposed neuro-dominance rule provides a sufficient condition for local optimality. It has been proved that if any sequence violates the neuro-dominance rule then violating jobs are switched according to the total weighted tardiness criterion. The proposed neuro-dominance rule is compared to a number of competing heuristics and meta heuristics for a set of randomly generated problems. Our computational results indicate that the neuro-dominance rule dominates the heuristics and meta heuristics in all runs. Therefore, the neuro-dominance rule can improve the upper and lower bounding schemes.

Keywords: Neuro-dominance rule, weighted tardiness problem, single machine scheduling.

1 Introduction

A new neuro-dominance rule which provides a sufficient condition for local optimality for a signle machine total weighted tardiness problem with unequal release dates, $1 \mid r_i \mid \sum w_i T_i$ is presented. Despite the fact that costumer orders can not reach simultaneously in daily life problems, according to the literature and the best of our knowledge we know that there is only one exact approach on the $1 \mid r_i \mid \sum w_i T_i$ problem. Recently, Akturk and Ozdemir [1] proposed a new dominance rule for $1 \mid r_i \mid \sum w_i T_i$ problem that can be used in reducing the number of alternatives in any exact approach. Akturk and Ozdemir used a interchange function, $\Delta_{ij}(t)$, is used to specify the new dominance properties, which gives the cost of interchanging adjacent jobs i and j whose processing starts at time t. Akturk and Ozdemir found seven breakpoints using the cost functions and obtained a number of rules by using the breakpoints. The problem may be described in the following form: There are n jobs independent. Each of them has an integer processing time p_j, a release date r_j, a due date d_j, and a positive weight w_j. Chu and Portmann [2] has stated in their paper that this problem could be simplified using corrected due dates, i.e. if $r_j + p_j > d_j$ then d_j takes the value r_j+p_j. Jobs will be processed without interrupting on a single machine which can process only one job at a time. If job j is completed after due date dj, a

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 963 – 973, 2006.

tardiness penalty is exceeds for each time unit, thus Tj=max{0,(Cj-dj)}, where Cj and Tj are the completetion time and the tardiness of the job j, respectively. The aim is to find a schedule minimizing the total weighted tardiness of all jobs given that any jobs cannot start processing before its release date. It is stated that the total tardiness problem with unequal release dates, $1 \mid r_i \mid \sum w_i T_i$ is NP-hard by Rinnooy Kan [3]. In a paper of Lawler, the total weighted tardiness problem, $1\| \sum w_i T_i$, has been shown strongly NP-hard, therefore the researchers know that unequal release date problems are already strongly NP-hard. Solution methods based on enumeration have been proposed for both weighted and unweighted situations when all jobs are initially present. Several dominance rules for $1\| \sum w_i T_i$, problem that limit the search for an optimal solution has been derived by Emmons. These results have been improved for $1\| \sum w_i T_i$, by Rachamadugu [4] and Rinnooy Kan et al [5]**.** Szwarc and Liu [6] have demonstrated a two-stage decomposition mechanism to $1\| \sum w_i T_i$ problem when there is a proportion between tardiness penalties and the processing times. Akturk and Yildirim [7] proposed more practical application about weighted tardiness problem and computing lower bound. Cakar [8] also proposd a neuro-dominance rule for single machine tardiness problem without release dates. All the optimization approaches mentioned above suppose that the jobs have equal release dates, even though the unequal release dates case has been evaluated for other optimality criteria. Branch and bound (B&B) algorithms has been presented by Chu [9] and Dessouky and Deogun [10] to minimize total flow time, $1 \mid r_i \mid \sum F_i$, whereas Bianco and Ricciardelli [11] and Hariri and Potts [12] take into consideration the total weighted completion time problem, $1 \mid r_i \mid \sum w_i C_i$. Potts and Van Wassenhove [13] has proposed a B&B algorithm for the minimization of the weighted number of late jobs. Erschler and his co-workers has proved a dominance relationship in the set of possible sequences for $1 \mid r_j$ problem independent of the optimality criterion to find a restricted set of schedules. Chu [9] has presented a paper based on the proof of some dominance properties and a lower bound for $1 \mid r_i \mid \sum T_i$ problem. Then, a B&B algorithm is formed by using the previous results of Chu and Portmann [2] and problems with up to 30 jobs can be solved for certain problem samples, even though this approach is limited for larger problems due to the computational requirements. Vepsalainen and Morton [14] has developed and tested efficient dispatching rules. An adequate condition for local optimality is provided by their proposed superior rule, and it generated schedules, which cannot be developed by adjacent job interchanges. In this paper, a trained BPANN to show how the proposed superior rule can be used to develop a sequence given by a dispatching rule. We also gave the proof of that if any sequence disturbs the proposed superior rule, and then switching the disturbing jobs either lowers the total weighted tardiness or leaves it unchanged. Because of the comprehensive computational requirements, according to the literature the weighted tardiness problem is NP-hard and the lower bounds do not have practical applications. Potts and Van Wassenhove [15] based on the linear lower bound is rather a weak lower bound, however the most promising one was presented by Abdul-Razaq et al.[16]. His study is in contradiction with the conjecture about this subject that one should limit the search tree as much as possible with using the sharpest possible bounds. The linear lower bound computations are based on an initial sequence. In this

paper, a solution that has a better upper bound value, which is near to optimal solution, is presented. Out solution also improves the lower bound value obtained from the linear lower bound method. Sabuncuoglu and Gurgun [17] proposed a new neural network approach to solve the single machine mean tardiness scheduling problem and the minimum makespan job shop scheduling problem. The proposed network by Sabuncuoglu and Gurgun combines the characteristics of neural networks and algorithmic approaches.

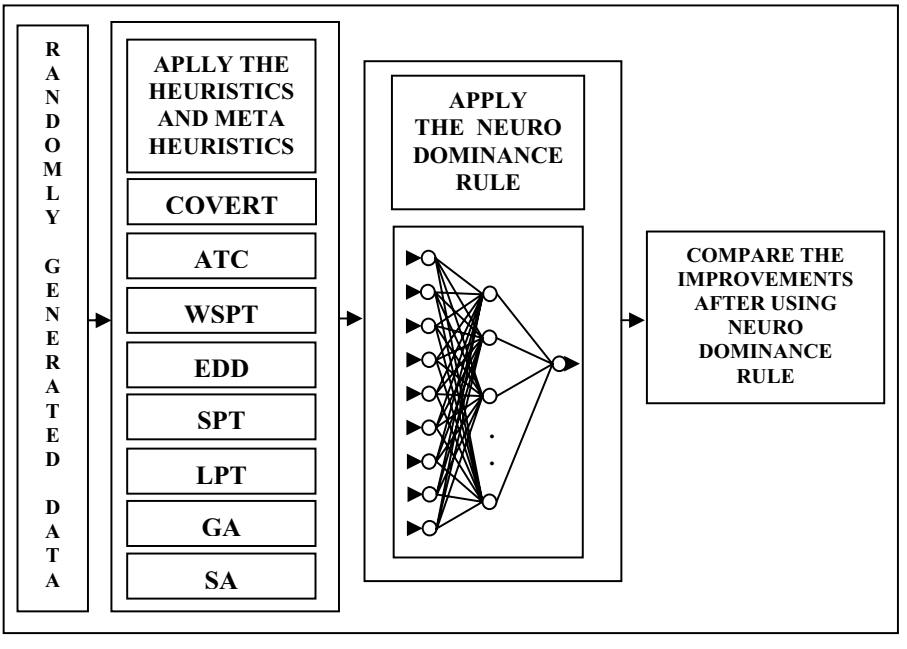


Fig. 1. Steps of the study from obtaining randomly data to comparison of the results

In this study, instead of extracting rules, finding break point using cost functions, an artificial neural network was trained using sufficient number of data which were different from Aktürk and Ozdemir's. When the necessary inputs were given to NDR, according to the total weighted tardiness problem criterion, The NDR decided which job will come first among the adjacent jobs. This paper is organized as follows; In the section 2, used parameters, modeling of the problem and how the proposed NDR works are discussed. In the section 3, used lower and upper bound schemes are explained. In the section 4, all of computational results and analysis are reported.

2 Problem Definition

The single machine problem may be explained as follows. Each job, which is numbered from 1 to n, should be processed with no interruption on a single machine,

which can use only one job at a time. All of the jobs will be available to be processed at time "0". If a job is presented with i, it has parameters as p_i, d_i, w_i, r_i which refer to an integer processing time, a due date, a positive weights and release date, respectively. The problem can be defined as finding a schedule S, which minimizes $f(S)=\sum_{i=1}^{N} w_i T_i$ function. The dominance rule may be introduced by considering schedules, where Q_1 and Q_2 are two disjoint subsequences the rest n-2 jobs, $S_1=Q_1ijQ_2$ and $S_2=Q_1jiQ_2$. $t=\sum_{k\in Q_1} p_k$ is the completion time of Q_1. In this study, it is decided that which job will be done firstly among two adjacent jobs according to the

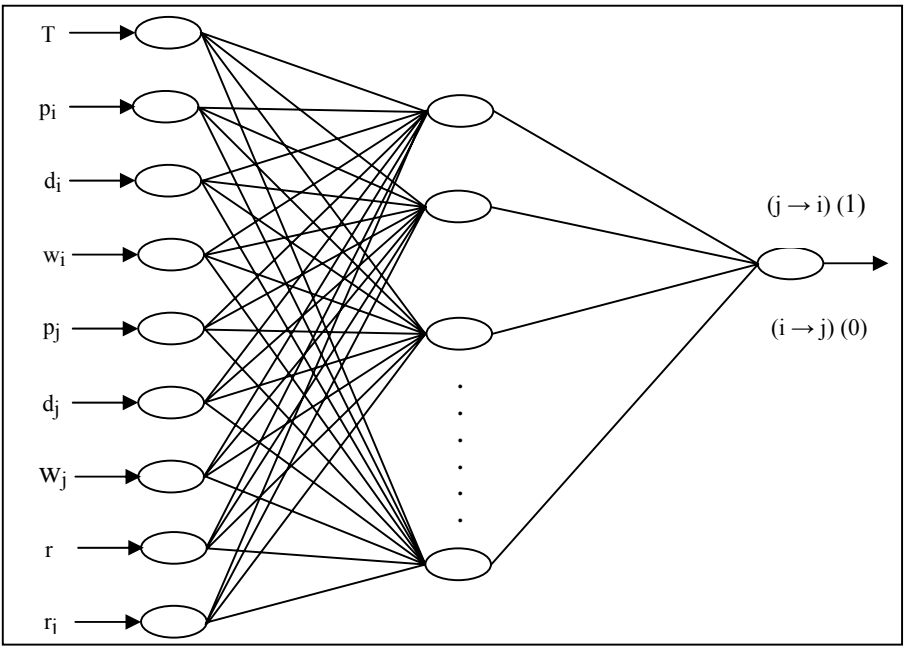


Fig. 2. Structure of the used BPANN. There are 9 input and 1 output

Table 1. Training and test parameters of the BPANN

Sample size and learned sample in training set	10000
Number of test data to test trained network	10000
Achievement rate of the test data (%)	%100
Activation function	Sigmoidal
Iteration number	4.700.000
Learning rate	0.35
Momentum rate	0.75

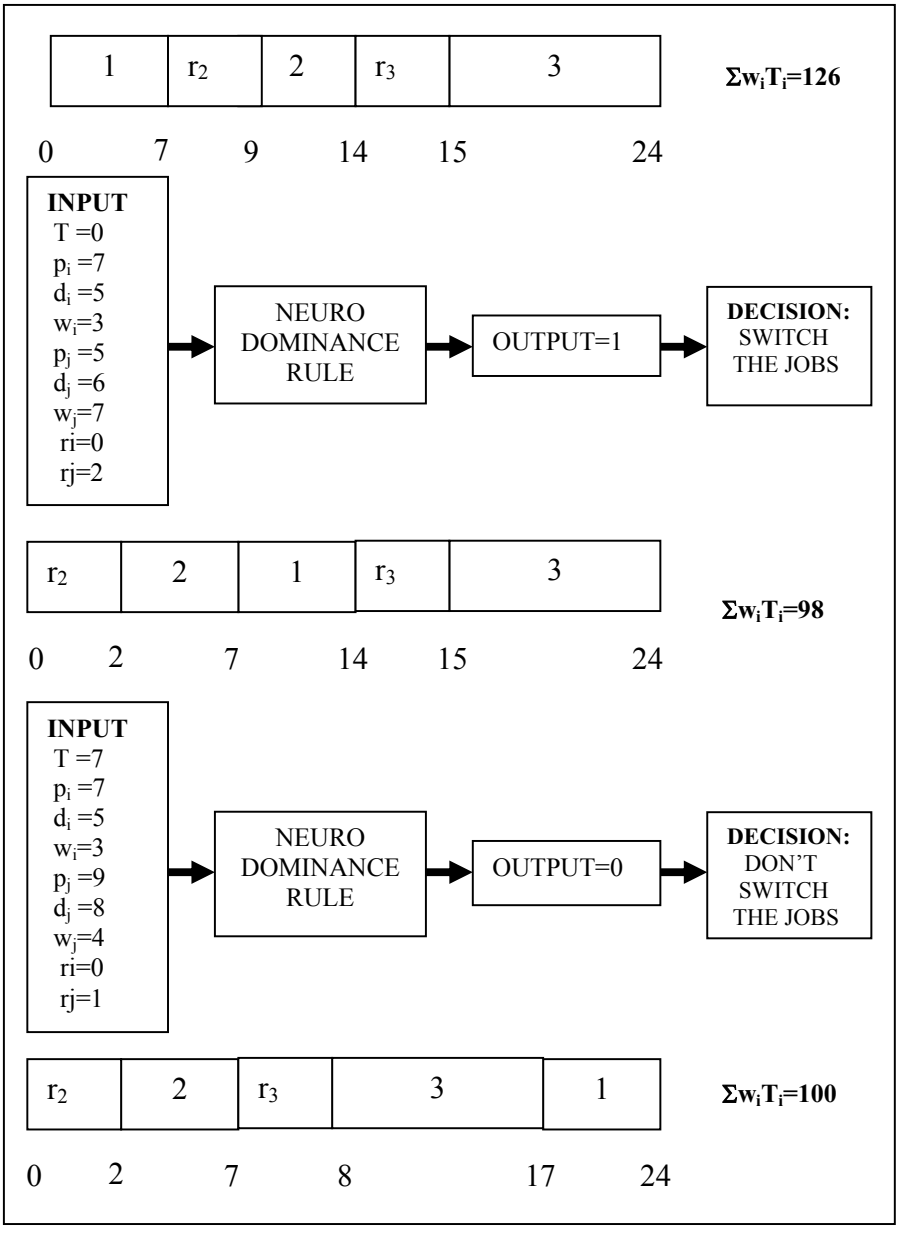


Fig. 3. An example: How the proposed neuro-dominance rule works

total weighted tardiness criterion using a trained BPANN. The first job is taken as i and the second one is taken as j jobs without considering due date or processing time. The used neural network has 9 inputs and 1 output, and there are 30 neurons in the hidden layer. The starting time of job i (T), the processing time of job i (p_i), due date

of job i (d_i), the weight of job i (w_i), the processing time of job j(p_j), the due date of job j (d_j), the weight of job j (w_j), release date of job i (r_i), release date of job j (r_j) are given as inputs to the BPANN. "0"and "1" values are used to determine the precedence of the jobs. If output value of the BPANN is "0", then i should precede j (i•j). If output value of the BPANN is "1" then j should precede i (j•i). Structure of the used BPANN can be seen in Figure 2. The parameters related to the training and test of neural network are given in Table 1. It can be seen that how the NDR works in Figure 3.

In Figure 3, inputs belonging to jobs 1 and 2 are given to NDR and output is obtained as "1". This means that the sequence of jobs 1 and 2 should be changed. The decrease in total weighted cost from 126 to 98 is an indication that NDR made the correct decision. In the following stage the inputs belonging to jobs 2 and 3 are given to NDR and output is obtained as "0". This means that the sequence of jobs 2 and 3 should not be changed. It can be verified that NDR made the correct decision by calculating the total weighted cost before and after switching jobs 2 and 3. If the sequence of job 2 and 3 was changed, the total weighted cost would have increased from 98 to 100.

3 Linear Lower Bound

Potts and Wan Wassenhove [13] have originally obtained the linear lower bound based on using the Lagrangian Relaxation approach with subproblems, which are total weighted completion time problems. Abdul-Razaq and his co-workers have presented additional derivation of it based on reducing the total weighted tardiness criterion to a linear function, i.e. total weighted completion time problem. For the job i, i = 1 to n, $w_i \geq v_i \geq 0$ and C_i is the completion time of job i, we have

$$w_i T_i = w_i \max\{C_i - d_i, 0\} \geq v_i \max\{C_i - d_i, 0\} \geq v_i (C_i - d_i) \quad (1)$$

Suppose that $v=(v_1, \ldots, v_n)$ is a vector of linear weights, i.e. weights for the linear function C_i-d_i, chosen so that $0 \leq v_i \leq w_i$. If so a lower bound can be expressed by given linear function below:

$$LB_{lin}(v) = \sum_{i=1}^{n} v_i (C_i - d_i) \leq \sum_{i=1}^{n} w_i \max\{C_i - d_i, 0\} \quad (2)$$

This situation shows that the total weighted completion time problem solution gets a lower bound on the total weighted tardiness problem. For any given v value, the optimal solution of the total weighted completion problem may be realized by the WSPT rule in which the jobs are sequenced in non-increasing order of v_i/v_p. An initial sequence is needed in the determination of the job completion time C_i to obtain the linear lower bound. Afterwards, v, refers to the vector of linear weights, is chosen to maximize $LB_{Lin}(v)$ with the condition of that $v_i \leq w_i$ for each job i. In Abdul-Razzaq's study, several lower bounding approaches have been compared and according to the their computational results the linear lower bound is found superior to others, which were given in the literature, because of the its quick computability and low memory requirement. In this paper, the impact of an initial sequence on the linear lower bound value will be tested and tried to present having a better, i.e. near optimal, upper bound

value will improve the lower bound value. This linear bound scheme also was used by Akturk and Yildirim.

4 Computational Results

In this study, each lower bounding scheme was tested on a set of randomly generated problems. We have tested the lower bounding scheme on problems with 50, 70 and 100 jobs, which were generated as: for each job i, p_i, and w_i were generated from two uniform distributions, [1, 10] and [1, 100] to create low or high variation, respectively. Here as stated early, p_i and w_i refers to an integer processing time and an integer weight, respectively. The proportional range of due dates (RDD) and average tardiness factor (TFF) were selected from the set {0.1, 0.3, 0.5, 0.7, 0.9}. d_i, an integer due date from the distribution [P(1-TF-RDD/2), P(1-TF+RDD/2)] was produced for each job i, here, P refers to total processing time, $\sum_{i=1}^{n} p_i$. Release dates are generated from a uniform distribution between 0 and $\mu\sum p_j$. As summarized in Table 2, we considered and evaluated 1200 example sets and took 100 replications for each combination resulting among 120.000 randomly generated runs.

Table 2. Experimental design

Factors	Distribution range
Number of jobs	50,70,100
Processing time range	[1-10], [1-100]
Weight range	[1-10], [1-100]
RDD	0.1, 0.3, 0.5, 0.7, 0.9
TF	0.1, 0.3, 0.5, 0.7, 0.9
μ	0.0, 0.5, 1.0, 1.5

To find an initial sequence for the linear lower bound, a number of heuristics were selected and their priority indexes were given as a summary in Table 3. The WSPT, EDD, LPT and SPT can be given as examples of static dispatching rules, where as ATC and COVERT are dynamics ones. Vepsalainen and Morton [14] have mentioned in their paper as: the ATC rule is superior to other sequencing heuristics and they defined it close to the optimal for the $\sum w_i T_i$ problem.

In addition to heuristics, two different meta heuristic, simulated annealing (SA) and genetic algorithms (GA), were used in this study. The parameters and operators used in SA to generate new solution were given. In this study, two different operator have been used to generate new neighborhood solution. Operators are swap and inverse operator. Total weighted tardiness was taken as a fitness function. In SA, the best value, obtained from heuristics, was taken as a starting solution.

Swap operator		Inverse operator	
Old solution	New solution	Old solution	New solution
19**8**4563**7**2	19**7**4563**8**2	19**8456**372	19**3654**872

Table 3. Priority Rules

RULE	RANK AND PRIORITY INDEX
COVERT	$\max\left[\frac{w_i}{p_i}\max\left(0,1-\frac{\max(0,d_i-t-p_i}{kp_i}\right)\right]$
ATC	$\max\left[\frac{w_i}{p_i}\exp\left(-\frac{\max(0,d_i-t-p_i}{k\bar{p}}\right)\right]$
WSPT	$\max\left(\frac{w_i}{p_i}\right)$
EDD	$\min(d_i)$
SPT	$\min(p_i)$
LPT	$\max(p_i)$

SA has some weak points such as long running time and difficulty in selecting cooling parameter when the problem size becomes larger. A geometric ratio was used in SA as $T_{k+1} = T_k$, where T_k and T_{k+1} are the temperature values for k and k+1 steps, respectively. Geometric ratio is used more commonly in practice. In this study, the initial temperature was taken 10000 and 0.95 was used for cooling ratio ( ).

In this study, when preparing initial populations in genetic algorithm, for any given problem, the solutions obtained from COVERT, ATC, EDD, WSPT, LPT; and SA methods, were also used. Others were randomly generated. Total weighted tardiness was taken as a fitness function. The parameters used in genetic algorithm were as given below.

Population size :100 Crossover rate : 100%
Max generation : 200 Mutation rate : 0.05

Linear Order Crossover (LOX) method has been applied to each chromosome independently. LOX works as follows:

1. Select the sublist from chromosomes randomly ;
chromosome #1 : 123**456**789
chromosome #2 : 645**713**298

2.Remove the sublist 2 from chromosome #1;
chromosome #1 : h2h456h89
chromosome #1 : 245hhh689

3. Remove the sublist 1 from chromosome #2;
chromosome #2 : hhh713298
chromosome #2 : 713hhh298

4. Insert sublist into holes to form offspring;
offspring #1 : 245713689
offspring #2 : 713456298

Mutation operator works as follows :

Select the randomly a chromosome and select the randomly two gene and swap the genes:

Selected genes : 3**7**6541**2**98 Mutation : 326541798

Table 4. Computational results for n=70

Heuristics and Meta Heuristics	UPPER BOUND			LINEAR LOWER BOUND		
	Before	After (+NDR)	$\overline{impr}$ (%)	Before	After (+NDR)	$\overline{impr}$ (%)
COVERT	21775908	20847680	4.63	21701505	20771103	4.49
ATC	21724406	20798295	4.83	21651947	20723363	3.65
EDD	18665220	17794675	7.12	18596259	17724392	10.95
WSPT	15632472	14887982	4.93	15480099	14730264	4.25
SPT	18456979	17603705	5.51	18314416	17457135	6.65
LPT	18977064	18115437	4.92	18916071	18051216	18.33
SA	15606987	15559236	3.45	15460244	15456987	3.56
GA	15598745	15589956	3.42	15445681	15432654	3.38

Table 5. Comparison of the linear lower bound (for n=50 and n=70)

	n=50				n=70			
	>	=	<	t-test	>	=	<	t-test
COVERT	38455	1120	425	47.67	36147	1692	2161	48.12
ATC	38940	750	310	47.72	36603	1133	2264	48.55
EDD	37105	2740	65	48.21	34878	4140	982	49.74
WSPT	37565	375	2060	48.78	35311	566	4123	49.57
SPT	39065	430	505	48.08	36721	649	2630	49.18
LPT	39570	525	155	48.29	37195	793	2012	49.46
SA+NDR	39577	362	61	48.86	39040	312	648	49.95
GA+NDR	39580	360	60	48.92	39083	298	619	49.97

Table 6. Comparison of the linear lower bound (for n=100)

	>	=	<	t-test
COVERT	36926	2549	525	48.92
ATC	37655	1707	638	48.69
EDD	33438	6238	324	49.33
WSPT	37012	852	2136	49.76
SPT	38264	978	758	49.16
LPT	37984	1195	821	49.53
SA+NDR	39001	712	287	49.82
GA+NDR	39040	689	271	50.03

If any sequence violates the dominance rule, then the proposed algorithm either lowers the weighted tardiness or leaves it unchanged. Firstly, to find an initial sequence we used one of the dispatching rules, afterwards the algorithm was applied to get the sequence indicated as Heuristic+NDR. The average lower bound value was calculated for each heuristic before and after implementing the algorithm along with the average improvement ($\overline{impr}$) and this situation is summarized in Table 4. ATC, COVERT, and WSPT seem to execute better than other heuristics in the literature when the dominance rule is applied to get the local optimal sequence. But, SA and

GA meta heuristics perform better than the other heuristics. Each heuristic and meta heuristic over 40,000 runs for 50, 70 and 100 job states were tested by us and given in Table 5 and Table 6. As stated above, (>) denotes number of runs in which sequence gotten from Heuristic+NDR gives a higher linear lower bound value than the sequence gotten from the heuristic, where as (=) denotes number of runs in which Heuristic+NDR executes as well as heuristic, and (<) denotes number of runs in which Heuristic+NDR executes worse. For instance, the combination of EDD+NDR executed 37105 times better (>) than EDD rule. According to the large t-test values on the average improvement, the proposed dominance rule provides an important improvement on all rules and the amount of improvement is noteworthy at 99.5% confidence level for all heuristics.

5 Conclusion

In this study, we have developed a neuro-dominance rule for $1 \mid r_i \mid \sum w_i T_i$ problem. A BPANN has been used to obtain the proposed neuro-dominance rule. Inputs of the trained BPANN are starting date of the first job (T), processing times (p_i and p_j), due dates (d_i and d_j), weights of the jobs (w_i and w_j) and release dates of the jobs (r_i and r_j). Output of the BPANN is a decision indicating which job should precede. The proposed neuro-dominance rule provides a sufficient condition for local optimality. Therefore, a sequence obtained by the proposed neuro-dominance rule cannot be improved by adjacent job interchanges. Computational results over 120,000 randomly generated problems indicate that the amount of improvement is significant. For the future research, single machine total weighted tardiness problem with double due dates can be modeled by using artificial neural networks.

References

1. Akturk, M.S., Ozdemir, D., A new dominance rule to minimize total weighted tardiness with unequal release date, European Journal of Operational research, 2001, 135, 394-412.
2. Chu, C. and Portman,M.C., Some new efficient methods to solve the $n \mid 1 \mid r_i \mid \sum w_i T_i$ scheduling problem, European Journal of Operation Research, 1992, 58, 404-413.
3. Rinnooy Kan, A.H.G., Machine sceheduling problems:Classification complexity and omputations , Nijhoff, The Hague, 1976.
4. Rachamadugu, R.M.V., A note on weighted tardiness problem, Operations Research, 1975, 23, 908-927.
5. Rinnooy Kan, A.H.G., Lageweg B.J. and Lenstra, J.K., Minimizing total costs in one machine scheduling, Operations Research, 1975, 23, 908-927.
6. Szwarc,W., and Liu, J.J., Weighted Tardines single machine scheduling with proportional weights, Management Science, 1993, 39, 626-632.
7. Akturk, M.S., Yidirim, M.B., A new lower bounding scheme for the total weighted tardiness problem, Computers and Operational Research, 1998, 25(4), 265-278.
8. Cakar, T., A New Neuro-dominance rule for single machine tardiness problem, Lecture Notes in Computer Science, 2005, 3483, 1241-1250.
9. Chu, C., A Branch-and-bound algorithm to minimize total tardiness with unequal release dates, Naval research logistics, 1992, 39, 265-283.

10. Dessouky, M.I. and Deogun, J.S., Sequencing jobs with unequal ready times to inimize mean flow time, SIAM Journal of Computing, 1981, 10, 192-202.
11. Bianco, L. and Ricciardelli, S., Sceheduling of a single machine to minimize total weighted completion time subject to release dates, Naval Research Logistics, 1982, 29(1), 151-167.
12. Hariri, A.M.A., and Potts, C.N., An algorithm for single machine sequencing with release dates to minimize total weighted completion time, Discrete Applied Mathematics, 1983, 5, 99-109.
13. Potts, C.N. and Van Wassenhove, L.N., A Branch and bound algorithm for total weighted tardiness problem, Operation Research, 1985, 33, 363-377.
14. Vepsalainen, A.P.J., and Morton, T.E., Priority rules for job shops with weighted tardiness cost, management Science, 1987, 33, 1035-1047.
15. Potts, C.N. and Van Wassenhove, L.N., Dynamic programming and decomposition approaches for the single machine total tardiness problem, European Journal of Operation Research, 1987, 32, 405-414.
16. Abdul-Razaq, T.S., Potts, C. N. And Van Wassenhove, L.N., A survey of algorithms for the single machine total weighted tardiness scheduling problem, Discrete Applied Mathematics, 1990, 26, 235-253.
17. Sabuncuoglu, I. And Gurgun, B., A neural network model for scheduling problems, European Journal of Operational research, 1996, 93(2), 288-299.

A Competitive Approach to Neural Device Modeling: Support Vector Machines

Nurhan Türker and Filiz Güneş

Yıldız Technical University, Electrical and Electronics Faculty, Department of Electronics and Communication Engineering, Yıldız, Istanbul, 34349, Turkey
{nturker, gunes} @yildiz.edu.tr

Abstract. Support Vector Machines (SVM) are a system for efficiently training linear learning machines in the kernel induced feature spaces, while respecting the insights provided by the generalization theory and exploiting the optimization theory. In this work, Support Vector Machines are employed for the nonlinear regression. The nonlinear regression ability of the Support Vector Machines has been demonstrated by forming the SVM model of a microwave transistor and it has been compared with its neural model.

1 Introduction

In empirical data modeling, a process of induction is used to build up a model of the system, from which it is hoped to deduce responses of the system that have yet to be observed. Ultimately, the quantity and quality of the observations govern the performance of this empirical model. By its observational nature, data obtained is finite and sampled; typically, this sampling is non-uniform and due to the high dimensional nature of the problem, the data will form only a sparse distribution in the input space [1]. Artificial Neural Networks [ANN] have emerged as a powerful technique for modeling general input/output relationships. The fact that neural networks can be trained with a simple and fast model for totally different applications has resulted in their use in diverse fields such as pattern recognition, speech processing, control, medical applications, and so forth [2]. The recent introduction of neural networks into the RF and microwave fields marks the birth of an unconventional alternative to modeling and design problems in RF and microwave CAD [3-4]. Neural networks can learn and generalize from data, thus allowing model development even when component formulas are unavailable. Neural network models are universal approximators that can be employed for modeling in both linear and nonlinear problems at both device and circuit levels.

Support Vector Machines (SVM) are a system for efficiently training linear learning machines in the kernel-induced feature spaces, while respecting the insights provided by the generalization theory and exploiting the optimization theory. An important feature of these systems is that, while enforcing the learning biases suggested by the generalization theory, they also produce "sparse" dual representations of the hypothesis, resulting in extremely efficient algorithms. This is due to Karush-Kuhn-Tucker conditions [5], which hold for the solution and play a crucial role in the practical implementation and analysis of these machines. Another important feature of

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 974 – 981, 2006.

Support Vector approach is that due to Mercer's conditions [6] on the kernels the optimization problems are convex and hence have no local minima. This fact, and the reduced number of non-zero parameters, marks a clear distinction between these system and neural networks [7]. The foundations of Support Vector Machines (SVM) have been developed by Vapnik [8] and are gaining popularity due to its attractive features, and promising empirical performance. The formulation embodies the Structural Risk Minimization (SRM) principle, which has been shown to be superior [9-10] to traditional Empirical Risk Minimization (ERM) principle, employed by conventional neural networks. SRM minimizes an upper bound on the expected risk, as opposed to ERM that minimizes the error on the training data. It is this difference, which equips SVM with a greater ability to generalize, which is the goal in statistical learning. SVM has been developed to solve the classification problem, but recently they have been proven to apply to the regression problems [11].

This work can mathematically be summarized as an application of the SVM to the nonlinear regression, thus twelve nonlinear real functions are generalized using limited number of data. This process is employed as active device modeling in microwave electronics. Using the manufacturer's data sheet, twelve characterization functions of a microwave transistor are approximated in the operation domain of the device, which are bias conditions of V_{DC}, I_{DS} and frequency *f*. Thus, the SVM model of the transistor is resulted and it is compared with the neural model given in [3].

In the following section, the SVM is briefly introduced and the foundations of the Support Vector Regression Machines (SVRM) are given.

2 Support Vector Regression Machines

The Support Vector method can be applied to the case of regression, maintaining all the main features that characterize the maximal margin algorithm: A non-linear function is learned by a linear learning machine in a kernel induced feature space while the capacity of the system is controlled by a parameter that does not depend on the dimensionality of the space. As in the classification case, the learning algorithm minimizes a convex functional and its solution is sparse. The approach can be summarized as "seeking to optimize the generalization bounds". This is relied on defining a loss function that ignored errors that were within a certain distance of the true value. This type of function is referred to as ε-insensitive loss function. The use of ε-insensitive loss function has the advantage of ensuring the existence of a global minimum and the optimization of a reliable generalization bound.

ε-insensitive Loss Function

The linear ε-insensitive loss function, $L^{\varepsilon} = (x, y, f)$ is defined by

$$L^{\varepsilon} = (x, y, f) = \left| y - f(x) \right|_{\varepsilon} = \max(0, \left| y - f(x) \right| - \varepsilon) , \tag{1}$$

where *f* is a real valued function on a domain *X*, $x \in X$ and $y \in Y$. In order to minimize the sum of the linear ε-insensitive losses the function can be given as

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} L^{\varepsilon}(x_i, y_i, f), \tag{2}$$

to control the size of $\|w\|$ for a fixed training set. C is a parameter to measure the trade-off between the complexity and losses. The primal problem can therefore be defined as follows:

$$\text{minimize } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \hat{\xi}_i), \tag{3}$$

$$\text{subject to } (\langle w.x_i \rangle + b) - y_i \le \varepsilon + \xi_i,$$
$$y_i - (\langle w.x_i \rangle + b) \le \varepsilon + \hat{\xi}_i,$$
$$\xi_i, \hat{\xi}_i \ge 0, \ i=1,2,\ldots,l.$$

where ξ_i and $\hat{\xi}_i$ are two slack variables, one for exceeding the target value by more than ε, and the other for being more than ε below the target. The corresponding dual problem can be derived using the standard techniques:

$$\text{maximize}$$
$$\sum_{i=1}^{l}(\hat{\alpha}_i - \alpha_i)y_i - \varepsilon\sum_{i=1}^{l}(\hat{\alpha}_i + \alpha_i) - \frac{1}{2}\sum_{i,j=1}^{l}(\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j)\langle x_i.x_j \rangle, \tag{4}$$
$$\text{subject to } 0 \le \alpha_i, \hat{\alpha}_i \le C, \ i=1,2,\ldots,l,$$
$$\sum_{i=1}^{l}(\hat{\alpha}_i - \alpha_i) = 0, \ i=1,2,\ldots,l.$$

The corresponding Karush-Kuhn-Tucker complementarity conditions are:

$$\begin{aligned} &\alpha_i(\langle w.x_i \rangle + b - y_i - \varepsilon - \xi_i) = 0, \\ &\hat{\alpha}_i(y_i - \langle w.x_i \rangle - b - \varepsilon - \hat{\xi}_i) = 0, \\ &\xi_i.\hat{\xi}_i = 0, \alpha_i.\hat{\alpha}_i = 0, \\ &(\alpha_i - C).\xi_i = 0, (\hat{\alpha}_i - C).\hat{\xi}_i = 0. \end{aligned} \qquad i=1,2,\ldots,l. \tag{5}$$

By replacing the inner product with an appropriately chosen "kernel" function, one can perform a non-linear mapping to a high dimensional feature space without increasing the number of tunable parameters, provided the kernel computes the inner product of the feature vectors corresponding to the two inputs [7].

In the following section, the signal-noise model of an active device is constructed by both SVM and ANN as an application example and the function approximation performances of each are investigated.

3 Application Example: The Signal-Noise Model of an Active Device

3.1 Determination of Small-Signal and Noise Behaviors of Active Microwave Devices

The signal and noise performance of an active microwave device around a bias point are usually given by the scattering S and noise N parameter vectors at the ω-domain and the measured performance data over the operational band can be arranged in a table form function as follows:

$$\begin{bmatrix} f_1 & S^{(1)} & N^{(1)} \\ f_2 & S^{(2)} & N^{(2)} \\ . & . & . \\ . & . & . \\ f_N & S^{(N)} & N^{(N)} \end{bmatrix} \tag{6}$$

where $S^{(1)}, N^{(1)}; \ldots ; S^{(N)}, N^{(N)}$ are, respectively, the scattering and noise vectors at the $f_1,\ldots,f_N$ sample operation frequencies, and $S^{(N)}$ and $N^{(N)}$ can be given as follows:

$$\left[S^{(N)}\right]^t = [|S_{11}|^{(N)} \varphi_{11}^{(N)} |S_{12}|^{(N)} \varphi_{12}^{(N)} |S_{21}|^{(N)} \varphi_{21}^{(N)} |S_{22}|^{(N)} \varphi_{22}^{(N)}] \tag{7}$$

$$\left[N^{(N)}\right]^t = [F_{opt}^{(N)} |\Gamma_{opt}|^{(N)} \varphi j_{opt}^{(N)} R_N^{(N)}] \tag{8}$$

The functions defined by equations 6-8 are utilized for training the SVM and neural model of the device. Then, the performance vectors $S^{(k)}$ and $N^{(k)}$ at a desired frequency, f_k, can be obtained from the network output by inputting the frequency, f_k. If $S^{(k)}$ and $N^{(k)}$ are unmeasured, they are determined by the generalization process, which can be considered as the ability of the model to give good outputs to inputs it has not been trained on.

N23200A FET is chosen as the active microwave device to be modeled and manufacturer's values for signal and noise parameters of N23200A FET arranged as defined in (6) are used as the training and test data for the SVM and neural model.

3.2 ANN Model for the Signal-Noise Parameters of an Active Device

The multilayer perceptron (MLP), with a single hidden layer having the same number of units as the output layer (Fig. 1), has been found to be sufficient to simulate an active microwave device; Levenberg-Marquardt backpropagation algorithm (BP) algorithm is utilized to train this network [3].

An additive bias is utilised as the second network input to ensure faster convergence which is taken as

$$Bias = \sqrt{\frac{1}{N_S}\sum_{1}^{N_S} f_i} \ , \tag{9}$$

where *Ns* is the sample number.

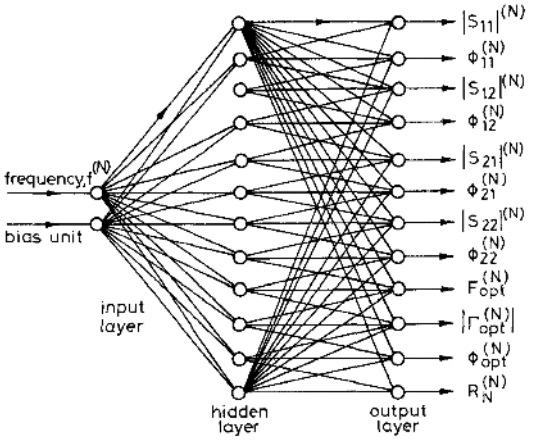


Fig. 1. MLP for an active microwave device

3.3 SVM Model for the Signal-Noise Parameters of an Active Device

In forming SVM model of the microwave transistor, a computer programme using the foundations defined in Section 2 is employed [12]. In the SVM model of the transistor, radial basis function kernel given in (10) with spread value of 0.1 is used.

$$K(x, x') = \exp(-\frac{\left\|x - x'\right\|^2}{2\sigma^2}) \ , \tag{10}$$

If we want to use neural network terminology, SVM model can be considered as the combination of twelve process machines, each of which is for a single characterization function and all the outputs are put together for the simultaneous device characterization.

4 Performance Measure and Results

Same training data are used in the training of SVM and the ANN models. Also, a dataset different from training data is used in testing of the models. Thus, Fig. 2-4 show the performances of the SVM and ANN models trained and tested with the same datasets.

To evaluate the quality of the fit to measured data the following error terms are found to be convenient:

$$E_{S_{ij}} = \frac{1}{n}\sum_{k=1}^{N}\frac{\left|S^k_{ij(meas)} - S^k_{ij(predict)}\right|}{\left|S^k_{ij(meas)}\right|}, \tag{11}$$

$$E_{N_i} = \frac{1}{n}\sum_{k=1}^{N}\frac{\left|N^k_{ij(meas)} - N^k_{ij(predict)}\right|}{\left|N^k_{ij(meas)}\right|}, \tag{12}$$

where S_{ij} and N_i are, respectively the signal and noise parameters and n is the number of discrete frequencies used. Total average error can be defined as the average of the signal and noise errors:

$$E_T = \frac{1}{4}\sum_{i=1}^{4} E_{i(signal)} + \frac{1}{3}\sum_{i=1}^{3} E_{i(noise)}. \tag{13}$$

In Table 1, the error analysis of the transistor N23200A FET is given for SVM and ANN models.

Table 1. The error analysis of the N23200A FET

N23200A FET	**ANN**	**SVRM**
$\mathbf{E_{S11}}$	0.0213	0.0932
$\mathbf{E_{S21}}$	0.0466	0.0616
$\mathbf{E_{S12}}$	0.0978	0.0925
$\mathbf{E_{S22}}$	0.0336	0.0127
$\mathbf{E_{ST}}$	**0.0498**	**0.0650**
$\mathbf{E_{N1}}$	0.0932	0.0177
$\mathbf{E_{N2}}$	0.0444	0.0964
$\mathbf{E_{N3}}$	0.0278	0.0237
$\mathbf{E_{NT}}$	**0.0551**	**0.0459**
E_T	*0.1097*	*0.1109*

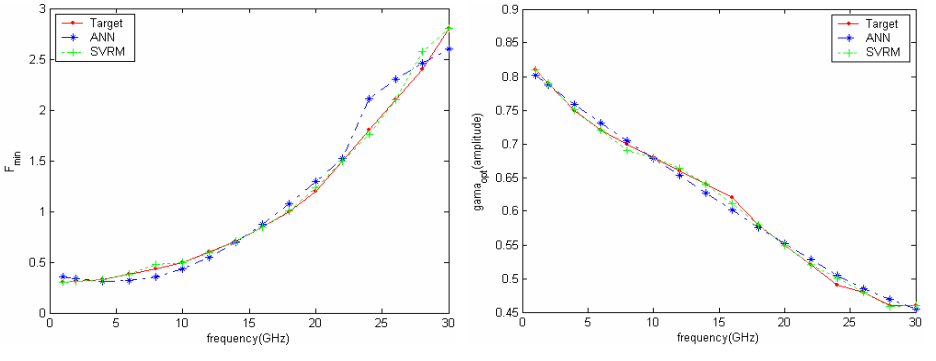


Fig. 2. (a) F_{min}; (b) Amplitude of Γ_{opt} against frequency

In Fig.2, the variations of F_{min} and Γ_{opt} with respect to frequency obtained by using the ANN and SVRM model of the N23200A FET comparable with target values are given.

The amplitude and angle of S_{11} and S_{22} against frequency obtained by using the ANN and SVRM model of the N23200A FET comparable with target values are given in Fig. 3 and 4, respectively. Table 1 and Fig. 2-4 show that SVRM model of the microwave transistor can be considered as a competitive approach to neural device modeling.

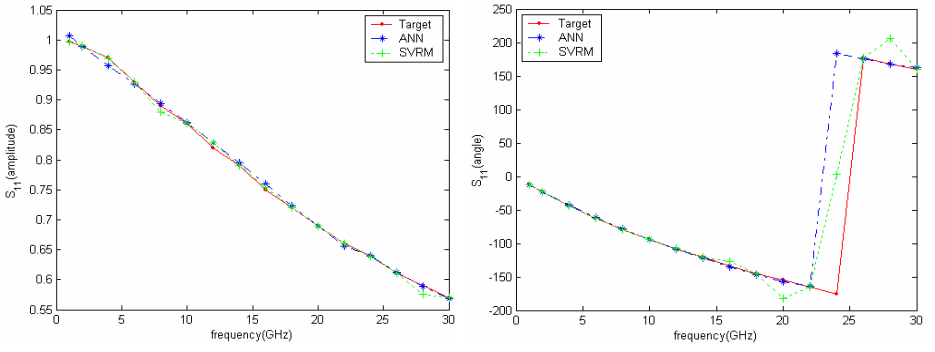


Fig. 3. (a) Amplitude; (b) Angle of S_{11} against frequency

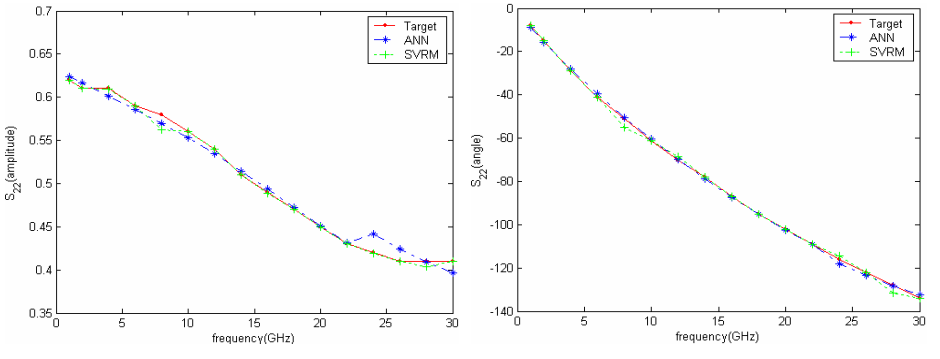


Fig. 4. (a) Amplitude; (b) Angle of S_{22} against frequency

5 Conclusion

In this work, we have experienced that the Support Vector Machines are the competitive machines against neural machines for the nonlinear functions. Each machine is considered as a mathematical module of nonlinear regression for each target function. The nonlinear regression ability of the Support Vector Machines has been demonstrated by forming the SVM model of a microwave transistor and it has been compared with its neural model.

References

1. Poggio, T., Torre, V., and Koch, C.: Computational vision and regularization theory. Nature, Vol.317 (1985) 314–319
2. Zhang, Q.J., Gupta, K.C.: Neural Networks For RF and Microwave Design. Artech House Publishers (2000)
3. Gunes, F., Gurgen, F., Torpi, H.: Signal-Noise Neural Network Model For Active Microwave Devices. IEE P-Circ Dev Syst Vol.143 (1996)
4. Gunes, F., Turker, N.: Artificial Neural Networks In Their Simplest Forms For Analysis And Synthesis Of RF/Microwave Planar Transmission Lines. Int J RF Microw C E Vol.15 (2005) 587-600
5. Kuhn, H., Tucker, A.: Nonlinear programming. In the proceedings of 2nd Berkeley Symposium on Mathematical Statistics and Probabilistics, University of California Press (1951) 481-492
6. Mercer, J.: Functions of positive and negative type and their connection with the theory of integral equations. Philos. Trans. Roy. Soc. London, (1909) 415-446
7. Cristianini, N., Shawe-Taylor, J.: An Introduction To Support Vector Machines And Other Kernel-Based Learning Methods. Cambridge University Press (2000)
8. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, N.Y. (1995) ISBN 0-387-94559-8
9. Gunn, S.R., Brown, M., Bossley, K.M.: Network performance assessment for neurofuzzy data modelling. Intelligent Data Analysis, Vol.1208 of Lecture Notes in Computer Science (1997) 313–323
10. Gunn, S.R., Support Vector Machines for Classification and Regression. Technical Report, University of Southampton (1998)
11. Vapnik, V., Golowich, S., Smola. A.: Support vector method for function approximation, regression estimation, and signal processing. Advances in Neural Information Processing Systems 9, Cambridge, MA, MIT Press. (1997) 281–287
12. Chang, C., Lin, C.: LIBSVM : a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm (2001)

Non-destructive Testing for Assessing Structures by Using Soft-Computing

Luis Eduardo Mujica[1], Josep Vehí[1], and José Rodellar[2]

[1] Department of Electronics, Computer Science and Automatic Control.
University of Girona (UdG), 17071 EPS-PIV Campus Montilivi, Girona, Spain
{lemujica, vehi}@eia.udg.es
[2] Department of Applied Mathematic III.
Technical University of Catalonia (UPC), 08034 Campus Nord, Barcelona, Spain
jose.rodellar@upc.es

Abstract. A hybrid system which combines Self Organizing Maps and Case Based Reasoning is presented and apply to Structural Assessment. Self Organizing Maps are trained as a classification tool in order to organize the old cases in memory with the purpose of speeding up the Case Based Reasoning process. Three real structures have been used: An aluminium beam, a pipe section and a long pipe.

1 Introduction

Many industrial end-users have determined their technical problems and the corresponding high cost related to inspections of several infrastructure installations. The inspections usually consist in detecting corrosion and unwanted sediment accumulation in the bottom of pipes (i.e. oil, gas and water pipelines, pipes in power plants, etc.) and tanks (i.e. oil tanks, ships, etc). Existing inspection techniques commonly use sophisticated equipment, applied in the proximity of the defect or costly robot techniques.

The need for additional global damage detection methods that can be applied to complex structures has led to the development and continued research of methods that examine changes in the vibration characteristics of the structure. Damages which alter the stiffness, mass or energy dissipation properties of a structure should be analyzed using an system that is composed by actuators and sensors and the structure is exposed to known external energy inputs from the actuator (see figure 1). An excitation signal is applied and the dynamic response is examined. The damage will alter the measured dynamic response of the system.

Most of the Non-Destructive Testing (NDT) techniques include Artificial Intelligence and it usually applies: wavelet transformations [HNR00], artificial neural networks [YYJ03], genetic algorithms [CG01], and statistical analysis [SWF02]. However, the use of knowledge-based approaches (as CBR), regardless of being suggested by Natke and Yao in 1993 [NY93], it has not been exploited specifically for damage detection. However, in the field of structural design, some researchers [LM96] have applied CBR to bridge design.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 982–991, 2006.

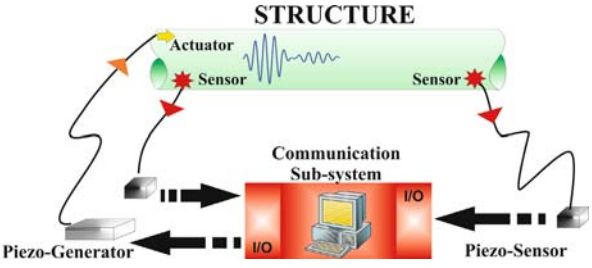


Fig. 1. Active system for vibration-based damage identification

2 CBR Methodology for Structural Assessment

This section describes a methodology for structural assessment (identification of the damage, its location, size and severity) using Case-Based Reasoning (CBR) [MVRK05]. First, in a "learning mode", structural damaged responses are used to generate a set of cases. These responses can be obtained from either simulations using a model of the structure, or experiments that have been previously performed. Using Self Organizing Maps (SOM) as a classification tool [Koh90], an initial casebase is built (see Fig. 2a). To reduce the number of input signals to the SOM, the Wavelet Transform is used to extract features from the measured signal while retaining most of the intrinsic information. When the system is in the "operation mode", similar old cases are retrieved. The localization and severity of the damage are obtained directly from heuristic considerations. Each new experience is retained once the damage has been detected.

This methodology is described using a numerical example of a cantilever truss structure with eight sections (see Fig. 2b). The material and geometric

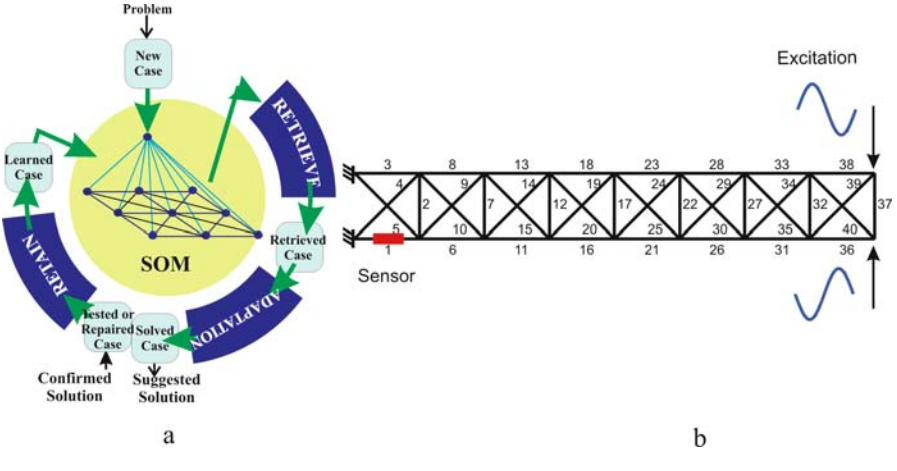


Fig. 2. (a) Proposed CBR cycle. (b) Cantilever truss structure.

specifications have been previously assigned. Two antiphase sine excitation forces are applied to elements 36 and 38. The element 1 was chosen as the sensor receiving the propagated wave.

2.1 Casebase Building

The casebase is an array in memory organizing all the cases to facilitate the search for the case most similar to the current problem. In the proposed methodology, the casebase is a SOM. Each case is defined by the defect of the structure and the minimal representation of its damaged dynamic response. In this situation, the minimal representation is the set of principal features that are extracted from the coefficients of the Wavelet Transform applied to the dynamic response. The wavelet coefficients are computed for each selected case. The coefficients at the same position in different cases are considered as samples of independent random variables. Therefore, bearing in mind the Central Limit Theorem, each variable is approximately normally distributed. The maximal normal numbers and the maximal wavelet coefficients occur at the same positions, which determine the midpoints of the clusters. This pattern of clusters contains relevant signal information. Later, each feature is determined as the square root of the energy of the wavelet coefficients in the corresponding cluster [PK99]. The process of feature extraction and building the casebase can be seen from Fig. 3. After the set of cases is generated (defect and the principal features of the dynamic response) they are organized in memory for recovery at the required time, and an SOM is created and trained. This SOM has l neurons (one for each feature) in the input layer and $m * n$ (according to the number of cases to store) clusters or neurons in the output layer. In the example, the SOM has 65 input neurons and $50 * 50$ output neurons. In each cluster, this network organizes the cases with similar characteristics.

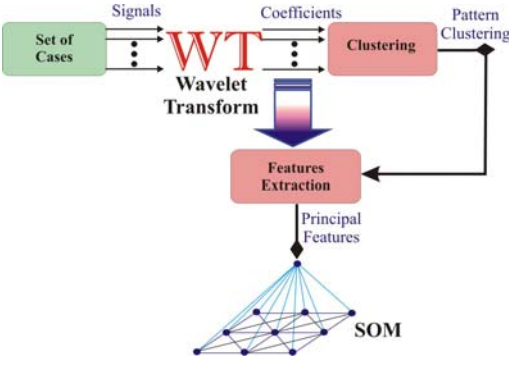


Fig. 3. Casebase building

2.2 Retrieving

Checking the methodology or putting the system in operation mode can be performed by simulation, laboratory testing, and even in normal working conditions for real structures. When a new experiment is carried out, the dynamic response captured by the sensors is obtained. From this signal, the principal features are extracted using the clustering pattern previously defined. From these features the SOM retrieves a set of stored cases with similar characteristics. Table 1 gives some cases (Damaged element and its severity) which are retrieved with their distances. This distance indicates the separation or space between the input vector and the cluster, in other words, it represents the similarity between the new case and the stored cases. The smaller the distance, the more similar the cases.

Table 1. Retrieved cases

Damaged element	Severity	Distance
-15-	-30%-	0.00362
-14-15	-10%-10%-	0.00747
-11-14-15-	-10%-10%-10%-	0.01123
-11-15-	-10%-10%-	0.01483

2.3 Adapting

From the retrieved cases (Table 1), it is noted that the element 15 appears in the first four cases and the element 11 appears three times but not with the least distances. We want to reward (1) elements that are repeated several times—the more frequent the repetition, the higher the probability of being the "winner"; and (2) similar cases—the smaller the distance, the higher the probability of being the "winner". To do this, a factor is calculated for the element, which is the sum of the inverses of the distances in which this element is present. For example, the factor for the elements 11,14 and 15 are:

$$F_{11} = \frac{1}{0.01123} + \frac{1}{0.01483} + \frac{1}{0.01517} = 222.40 \tag{1}$$

$$F_{14} = \frac{1}{0.00747} + \frac{1}{0.01123} + \frac{1}{0.01517} = 288.84 \tag{2}$$

$$F_{15} = \frac{1}{0.00362} + \frac{1}{0.00747} + \frac{1}{0.01123} + \frac{1}{0.01483} = 566.59 \tag{3}$$

By normalizing these factors, the probabilities of damage in each element are obtained (Element 15 -higher factor- has probability 1, Element 14 probability 0.51, and Element 11 probability 0.39, in the presented example). To calculate the dimension and the severity of the defects, a weighted average is computed, using as a weighting coefficients the inverse of the distances (see Eqs. 4,5), where n is the total number of retrieved cases, dim, dam and d are the dimension, the

damage and the distance of each retrieved case, respectively. Note that $d(1)$ is the minimum distance. In this case the dimension is 1.7, which is rounded to two elements (Elements 15 and 14) and the severity is 26.8% (stiffness reduction).

$$Dimension = \sum_{j=1}^{n} dim(j) * \frac{d(1)/d(j)}{\sum_{i=1}^{n} d(1)/d(i)} \tag{4}$$

$$Severity = \sum_{j=1}^{n} dam(j) * \frac{d(1)/d(j)}{\sum_{i=1}^{n} d(1)/d(i)} \tag{5}$$

3 Beam Case Study

The first structure to study is an aluminium cantilever beam. A numerical model using Finite Element Method is considered, as illustrated in Fig. 4a, with a total of 102 elements. It is equipped with a piezoelectric actuator (as can be seen from Fig. 4b) mounted close to the clamped end that induces a sine wave excitation signal of the 142.9 Hz frequency and only one period duration. The piezo-patch sensor (shown in Fig. 4c) close to the free end measures the bending strains (curvature) at the specified location.

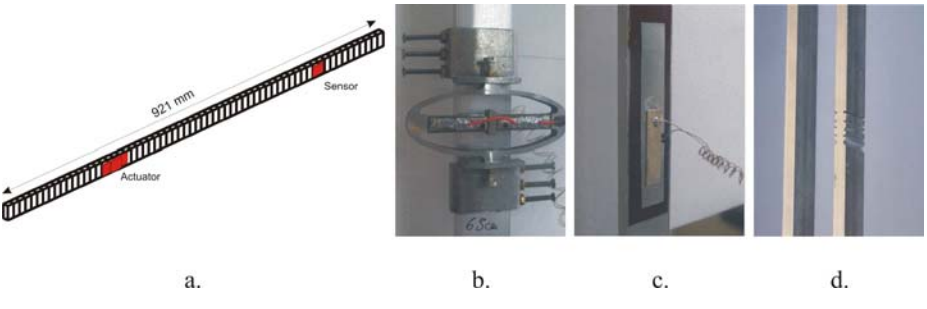


Fig. 4. Beam. (a) Model (b) Actuator (c) Sensor (d) Damage.

Damage identification experiments in this structures using CBR combining with SOM methodology has been reported in [MVRK05]. The results can be summarized as following: A total of 5464 cases of the damaged structure have been simulated (up to 10 consecutive elements with 12 different reductions of mass) using a finite elements model. A total of 57 principal features have been extracted from each response signal. An SOM of 57 input neurons and 50*50 output neurons has been trained in 35 minutes. Three examples are presented, two using numerical simulations and the third one using experimental data from the real structure.

Simulation of one fault in five consecutive elements: Damage was simulated in elements 43-44-45-46-47 with a stiffness reduction of 5%-15%-15%-15%-5%, respectively. The damage has been detected approximately in the assumed elements and a stiffness reduction of 6.7% in each element (Fig. 5a).

Simulation of two faults in three consecutive elements: Two faults have been simulated in elements 28-29-30 and 59-60-61 with stiffness reductions of 20%-30%-20% and 20%-30%-20% respectively. The damages detected at approximately elements 29 and 60, and a stiffness reductions of 25% in each element (Fig. 5b).

Experimental damage in unknown elements: Damage was caused to the real structure in elements 44-45-46 (see Fig. 4d). It can be observed that the damage has been detected in the neighbourhood of the element 46 and a stiffness reduction of 45% (Fig. 5c).

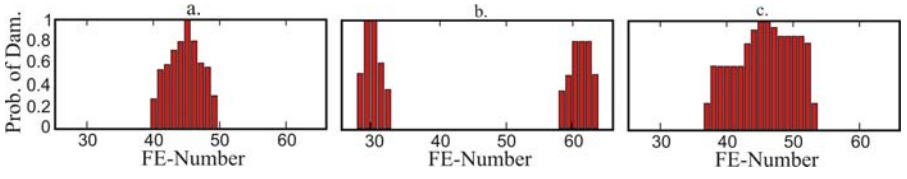


Fig. 5. Elements identified with damage. (a) Simulating one fault of 5 elements (b) Simulating two faults of 3 elements (c) Experimental damage in unknown elements.

4 Pipe Section Case Study

This section presents the result obtained on numerical and experimental tests on a section pipe. Due a calibration problems, two casebases have had to be built and verify by separate. One of them is loaded using only simulations, the other one, using only experiments.

4.1 Experimental Test

A French company provided a pipe section and configured the experimental setup shown in Fig. 6. It has an internal radius of $40mm$, thickness of $2mm$ and its useful length is $5550mm$. It is excited using a 7-cycle Hanning windowed sine pulse with 750 Hz frequency, near to its first radial-axis mode, using generator supported as can be seen from Fig. 6. Four sensors measure the dynamic response of this structure. Reversible defects have been performed. Five masses (M1=50g, M2=100g, M3=300g, M4=200g, M5=250g) have been added in different positions (marked in the Fig. 6 as Position 1 to Position 9). The casebase contains a total of 46 cases (1 undamaged case and 45 damaged cases). Firstly, the methodology has been tested using some defects previously known, these experiments are part of the casebase. Finally, it is tested using 9 unknown tests.

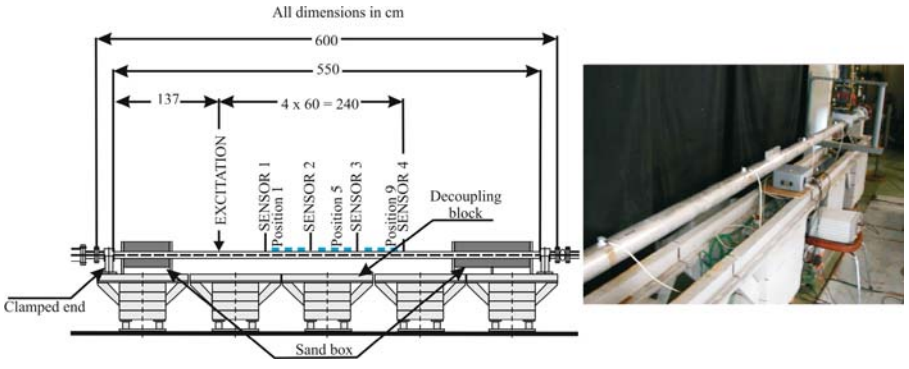


Fig. 6. Experimental setup configuration

Table 2. Defect localization and identification of nine unknown defects

Defect #	Position	Added mass
1	2	204.4
2	5	211.6
3	8	173.5
4	7	223.9
5	3	226.7
6	4	243.8
7	1	241.6
8	9	141.2
9	6	220.2

Mass of 200g added in position 1: The information of the four sensors leads methodology to detect the defect in position 1 with an intensity of 206.44g.

Mass of 250g added in position 5: As the previous test the methodology detects the defect in position 5 and the calculated intensity is 166.43g.

Mass of 50g added in position 9: The methodology detects the defect in position 9 and the calculated intensity is 69.1g.

Unknown defects: A final test with an unknown added masses at unknown positions has been performed. From Table. 2 it can be seen a summary of the results showing the identified position and intensity (added mass). After discussing these results with the company, they confirm that the localization of every experiment is very good, however, the identification of the mass must be improved, because all experiments was performed using a mass of 150g.

4.2 Numerical Test

The numerical model of this pipe section, was not able to be calibrated. The excitation signal is a sine wave with only one period and frequency of 773.75 Hz. Four sensors are used as well. The pipe has been divided in 16 sections (as

can be seen from Fig. 7) and defects in each section by reducing thickness in 20% and 50% around the pipe (axi-symetrical reduction) have been simulated for building the casebase. To test the methodology, several defects have been simulated: Before the actuator (elem. 2 with defect of 20%), between actuator and sensor 1 (elem. 3 with defect of 20%), between sensor 1 and 2 (elem. 6 with defect of 20%), between sensor 2 and 3(elem. 10 with defect of 20%), between sensor 3 and 4 (elem. 14 with defect of 20%), half element (upper half of elem. 9 with defect of 50%) and two elements (elem. 7 and 8 with defect of 50%). The system perfectly locates all these damages and estimates thickness reductions of 30.7%, 29.2%, 28.9%, 30.2%, 29.7%, 33.8%, 38.3% respectively as can be seen from Table. 3.

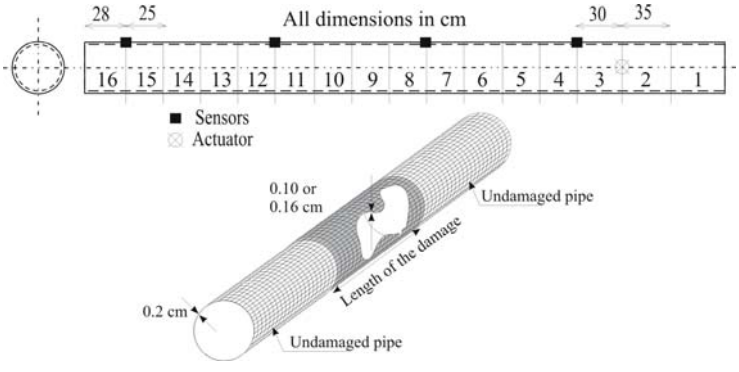


Fig. 7. Pipe section model

Table 3. Defect localization and identification of seven simulated defects

Defect	Simulated damage		Identified damage	
#	Position	Reduced mass	Position	Reduced mass
1	2	20	2	30.7
2	3	20	3	29.2
3	6	20	6	28.6
4	10	20	10	30.2
5	14	20	14	29.7
6	Upper half of 9	50	9	33.8
7	7 & 8	50	6-7-8	38.3

5 Long Pipe Case Study

A long pipe of 80 meters length was provided by a French company of district heating network. The experimental setup is shown in Fig. 8. The material of this pipe is steel AE220, its internal radius is $15cm$, thickness of $0.45cm$ and its length is $7887cm$. It is excited using a 5-cycle Hanning windowed sine pulse

with 474Hz frequency, near to its first radial-axis mode. Just one sensor are measuring dynamic response, it is locate 58 meters far from the actuator. Due to the impossibility of having a efficient model for the pipe, this study has been carried out using only experiments over the real structure. Reversible defects have been performed. Three masses (M1=400g, M2=1000g, M3=5000g) have been added on 58 different positions, almost in every one meter from the actuator to the sensor (58m). In total there are 151 cases to store into the casebase (1 undamaged case and 150 damaged cases). The system has been verified using some experiments which are previously stored into the casebase, the localization and estimation of the magnitude have been successful. However, due to problems of repetitiveness in this structure setup, two identical experiments do not have identical responses.

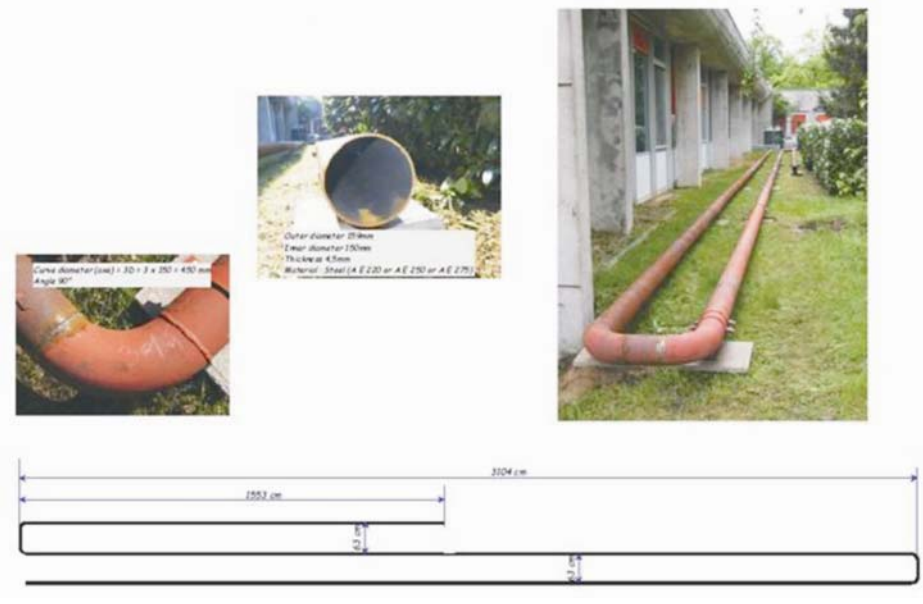


Fig. 8. Experimental setup

6 Conclusions

The feasibility of assessing structures using a knowledge-based reasoning approach has been demonstrated numerically and experimentally. This methodology performs satisfactorily in locating damage and assessing its size and severity for industrial needs. Two of its advantages are: (1) it exploits the model of the structure to preload the casebase in the initial learning mode, and (2) in the operational mode, it incorporates new real damage cases in the casebase, improving the robustness of the methodology against errors in the model.

Acknowledgements

This work has been partially funded by the European Union (European Regional Development Fund) and the Spanish government through the coordinated research projects DPI2003-07146-C02-02 and DPI2004-07167-C02-02 and by the government of Catalonia through 2005SGR-00296.

References

[CG01] J-H. Chou and J. Ghaboussi. Genetic algorithm in structural damage detection. *Computers & Structures*, 79(14):1335–1353, jun 2001.

[HNR00] Z. Huo, M. Noori, and R.S.Amand. Wavelet-based approach for structural damage detection. *Journal of Engineering Mechanics*, 126(7):677–683, jul 2000.

[Koh90] T. Kohonen. The self-organizing map. *Proceeding in IEEE*, 78(9):1464–1480, 1990.

[LM96] M.S. Lehane and C. J. Moore. Applying case-based reasoning in bridge design. In B. Kumar, editor, *Information Processing in Civil and Structural Engineering*, pages 1–5. Inverleith Spottiswoode, 1996.

[MVRK05] L.E. Mujica, J. Vehí, J. Rodellar, and P. Kolakowski. A hybrid approach of knowledge-based reasoning for structural assessment. *Smart Materials and Structures*, 14:1554–1562, nov 2005.

[NY93] H.G. Natke and J.T.P. Yao. Detection and location of damage causing non-linear system behaviour. In H.G. Natke, G.R. Tomlinson, and J.T.P. Yao, editors, *Proc. of the Internat. Workshop on "Safety Evaluation Based on Identification Approaches Related to Time Variant and Nonlinear Structures"*, Lambrecht, 1993. Vieweg, Braunschweig.

[PK99] S. Pittner and S.V. Kamarthi. Feature extraction from wavelet coefficients for pattern recognition tasks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(1):83–88, jan 1999.

[SWF02] H. Sohn, K. Worden, and C.R. Farrar. Statistical damage classification under changing environmental and operational conditions. *Journal of Intelligent Material Systems and Structures*, 13(9):561–574, 2002.

[YYJ03] L.H. Yam, Y.J. Yana, and J.S. Jiang. Vibration-based damage detection for composite structures using wavelet transform and neural network. *Composite Structures*, 60(4):403–412, jun 2003.

Neural Unit Element Application for in Use Microwave Circuitry

M. Fatih Çağlar[1] and Filiz Güneş[2]

[1] Department of Electronics and Communication Engineering,
Süleyman Demirel University, 32260 Isparta, Turkey
mfcaglar@mmf.sdu.edu.tr
[2] Department of Electronics and Communication Engineering,
Yıldız Technical University, 34349 Istanbul, Turkey
gunes@yildiz.edu.tr

Abstract. In this work, a Neural Unit Element (NUE) is defined to be used in the analysis and synthesis of the microwave circuits. For this purpose, analysis of impedance transformation property of a transmission line segment with the parameters ($\beta\ell$, Z_O) is defined as the problem in the forward direction and synthesis of the transmission line to obtain the target impedance is also defined the problem in the reverse direction. This problem is solved using Multilayer Perceptron (MLP) with efficient training algorithm. Finally, NUE driven by 50Ω and complex source which is very common in microwave applications and the short-circuited NUE (Stub) are given as the worked examples.

1 Introduction

Neural networks are universal function approximators allowing reuse of the same modeling technology for both linear and nonlinear problems at both device and circuit levels. Yet neural network models are simple and model evaluation is very fast. Recent works have let to their use for modeling of both active and passive components such as transistors [1], [2], planar transmission line microstrip, coplanar wave (CPW) guides [3], vias, CPW discontinuities, spiral inductors [4]. Furthermore ANNs have found modeling in Smith Chart representation and automatic impedance matching [4].

Neural modeling of the microwave components and circuits has also found applications in solving very important problems of the microwave circuit theory. Design Target Space can be considered as such an important problem of optimization of microwave amplifier and black-box neural modeling of microwave transistors [1], [2] has yet found a good application in solving this problem [5], [6]. Another significant problem of the microwave amplifier is design of the matching circuits to be used as the front- and back-ends of the transistor, to provide the necessary source Z_S and load Z_L impedances. Matching circuits are configured of the distributed-lumped mixed elements and do the impedance transformations between the ports. These transformations are basically very highly nonlinear functions of the distributed-lumped element parameters. In [7], the highly nonlinear impedance transformation equation set of a transmission line segment ($\beta\ell$, Z_O) are solved by the neural networks for the purely resistive driving.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 992 – 1001, 2006.

Here, this work is extended to the complex driving. Briefly, the impedance transformation properties of a transmission line segment with the parameters ($\beta\ell$, Z_O) throughout an operation bandwidth B will be modeled for analysis and synthesis purposes by ANN techniques. In the next section, the problem will basically be defined in both forward and reverse directions. Later the solution is presented and worked examples are given.

2 Definition of Problem

The forward and reverse problems for the impedance transformation properties of the microwave circuits can be defined by means of the two black-boxes: (i) the forward problem: Black-box in analysis (Fig. 1a); (ii) the reverse problem: Black-box in synthesis (Fig. 1b). Calculation mechanism in each black-box is a neural network which is either Multilayer Perceptron (MLP) or Radial Basis Function (RBF) network [7].

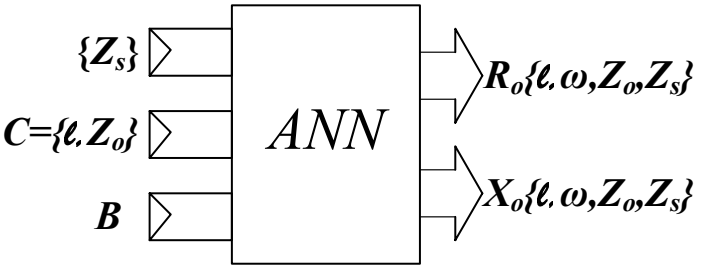


Fig. 1a. Black-Box in Analysis

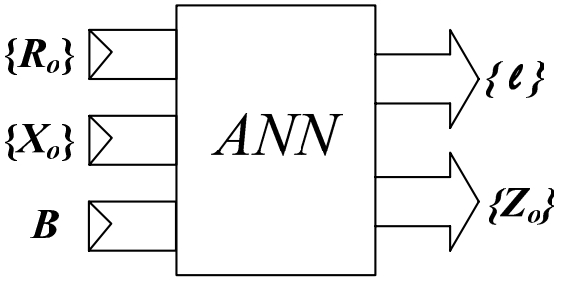


Fig. 1b. Black-Box in Synthesis

2.1 The Forward Problem: Impedance Transformation from the Input Port to the Output Port of the Two-Port

The input quantities to the Analysis Black-Box are the input termination $\boldsymbol{Z_S(\omega)=R_S(\omega)+jX_S(\omega)}$, circuit parameters $\boldsymbol{C}$ and the operation bandwidth $\boldsymbol{B}$ which includes the bandwidths B_1 and B_2 between the frequencies f_{min} and f_{max} for the inductive and capacitive behaviors separately. The corresponding output quantities are

the real $\boldsymbol{R_O(\omega)}$ and the imaginary $\boldsymbol{X_O(\omega)}$ parts of the output impedance throughout the defined operation bandwidth B.

2.2 The Reverse Problem: Synthesis of the Impedance $Z_O(\omega)=R_O(\omega)+jX_O(\omega)$ Using the Circuit Parameters

In the synthesis side of the problem, similar terminology to the analysis mechanism is used. So input quantities are the required $\boldsymbol{R_O(\omega)}$ and $\boldsymbol{X_O(\omega)}$ functions and the operation bandwidth $\boldsymbol{B}$ between the frequencies f_{min} and f_{max}. However, the circuit parameter vector $\boldsymbol{C}$ transforming the input termination $\boldsymbol{Z_S(\omega)}$ into the $\boldsymbol{Z_O(\omega)}$ take place as the output parameters of the synthesis block.

Since functions used in the analysis and synthesis are generally inverse of each other, so for the purpose of determining network parameters of ANNs used in analysis and synthesis, a single program can be prepared using only the analysis formulae of the circuit. Then the output data obtained from this program is arranged with respect to the input-output definition of each ANN so that the data sets can be resulted to train and test ANNs for both the purposes of analysis and synthesis. So we can have inverse of a function in this way provided that always one-to-one mapping relations between the inputs and outputs.

Impedance transformation properties of the unit element (UE) ($\beta\ell$, Z_O) driven by a complex Z_S impedance (Fig. 2) will be considered in the next section. ANN modeling of the unit and stub elements and worked examples will take place in the later sections.

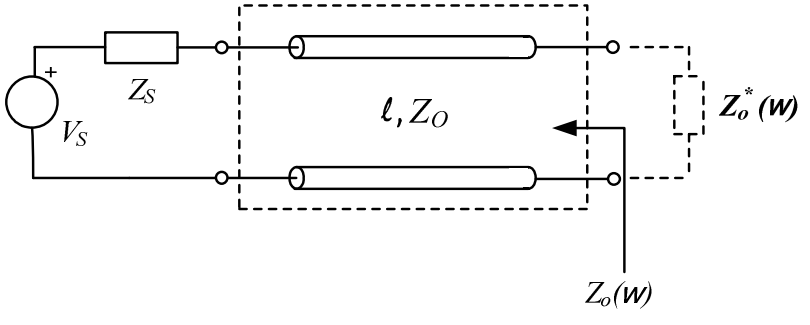


Fig. 2. Unit Element (UE)

3 Impedance Transformation

Output impedance $Z_O=R_O+jX_O$ of the UE with $\{\beta\ell, Z_O\}$ terminated by a complex impedance Z_S is the function of circuit parameters $\beta\ell$, Z_O and Z_S which can be expressed in the closed form as follows:

$$R_O=R_O\{\beta\ell, Z_O, R_S, X_S\} \quad (1a)$$

$$X_O=X_O\{\beta\ell, Z_O, R_S, X_S\} \quad (1b)$$

where

$$R_O(\beta\ell, Z_O, R_S, X_S) = Z_O \frac{R_S\left[Z_O(1+\tan^2\beta\ell) + X_S(1-\tan\beta\ell)\right]}{(Z_O - X_S\tan\beta\ell)^2 + R_S^2\tan^2\beta\ell} \tag{2a}$$

$$X_O(\beta\ell, Z_O, R_S, X_S) = Z_O \frac{(X_S + Z_O\tan\beta\ell)(Z_O - X_S\tan\beta\ell) - R_S^2\tan\beta\ell}{(Z_O - X_S\tan\beta\ell)^2 + R_S^2\tan^2\beta\ell} \tag{2b}$$

In the next section $f\ell$ product will be determined with respect to the reactive behavior of the circuit in the frequency domain.

3.1 $f\ell$ Product for the Reactive Behavior of the UE

$f\ell$ product for the reactive behavior of the UE is determined using the transformation relations given by (1a), (1b), (2a) and (2b) and the well-known mapping relations between the Z-rectangular and Γ-polar planes ↔ Smith chart which are:

$$\Gamma_L = \frac{Z_L - Z_O}{Z_L + Z_O} = \left|\Gamma_L\right| . e^{j\varphi_L} \tag{3a}$$

$$\Gamma(\ell) = \Gamma_L . e^{-j2\beta\ell} = \frac{Z_L(\ell) - Z_O}{Z_L(\ell) + Z_O} \tag{3b}$$

where Γ_L is the load reflection coefficient.

The problem necessitates considering the two cases:

i) Load is within the inductive region ⇔ $0<\varphi_L<\pi$

For this case, the $f\ell$ products have the following intervals for the inductive and capacitive behaviors:

$$0<\beta\ell<\varphi_L/2 \Leftrightarrow 0<f\ell<\varphi_L v/4\pi \Leftrightarrow \text{inductive behavior} \tag{4a}$$

$$\varphi_L/2<\beta\ell<(\varphi_L+\pi)/2 \Leftrightarrow \varphi_L v/4\pi<f\ell<\varphi_L v/4\pi+v/4 \Leftrightarrow \text{capacitive behavior} \tag{4b}$$

where v is the phase velocity within the transmission medium.

ii) Load is within the capacitive region ⇔ $\pi\le\varphi_L\le 2\pi$

In this case, the $f\ell$ products have the following intervals for the inductive and capacitive behaviors:

$$0<\beta\ell<\varphi_L/2-\pi/2 \Leftrightarrow 0<f\ell<\varphi_L v/4\pi-v/4 \Leftrightarrow \text{capacitive behavior} \tag{5a}$$

$$\varphi_L/2-\pi/2<\beta\ell<\varphi_L/2 \Leftrightarrow \varphi_L v/4\pi-v/4<f\ell<\varphi_L v/4\pi \Leftrightarrow \text{inductive behavior} \tag{5b}$$

So, maximum and minimum line lengths for the inductive and capacitive behaviors can be determined using the chosen corresponding limit frequencies f_{min} and f_{max} and the inequalities of (4a), (4b), (5a) and (5b). These will be demonstrated in the worked examples for the purely resistive termination.

4 Properties of Neural Networks

The universal approximation theorem for MLP has been proved by Hornik and Cybenko in 1989. According to their theorem, a 3-layer MLP can approximate a nonlinear, continuous, multi-dimensional function with any desired accuracy [8], [9]. However, here this universal function approximator property of the MLPs is employed in solving the nonlinear equations given by (1a), (1b), (2a),(2b) in both directions. Basically, the training and test data for both the forward and reverse ANNs is obtained from the Smith chart representation of the impedances transformed by a transmission line segment. However, this data is arranged considering one-to-one mapping relations between the input and the output of each type of ANN, given in (4a), (4b), (5a) and (5b).

The Levenberg-Marquardt (LM) back-propagation algorithm for the smallest testing error and three layered network with the minimum number of neuron for faster training are performed with the MLP type of network. These result in the fastest convergence by changing epoch, is seen in the Fig. 3. In fact, performance function of the MLP is shown with the Mean Squared Error (MSE) -Epoch variation in the Fig. 3, where how rapidly results converges can be seen for the reverse problem and similar performance has been obtained for the forward problem too. In the forward ANN, tangent-sigmoid function is used as the activation function of the both input and hidden neurons; however output layer neurons are activated by the linear function. The forward ANN has four input, two output and twenty hidden neurons. However, the reverse ANN has six input, two output and thirty-two hidden neurons. In the reverse ANN, also, both the input and hidden neurons are activated by the tangent-sigmoid function, and the output neurons are activated by linear function.

Table 1. Performance of MLP with different training algorithms

Training Algorithm	Minimum Training Error	Average Test Error	Epoch
Levenberg-Marquart BP	7.30E–04	8.65E–07	1706
Quasi-Newton BP	2.11E–02	8.45E–04	3000
Conjugate Gradient BP with Fletcher-Reeves	2.55E–01	1.96E–03	3000
Gradient Descent with Momentum and adaptive learning rate BP	8.93E–01	1.49E–02	3000
Resilient Backpropagation (BP)	1.81E–01	5.21E–03	3000
Scaled Conjugate Gradient BP	7.94E–02	4.20E–03	3000

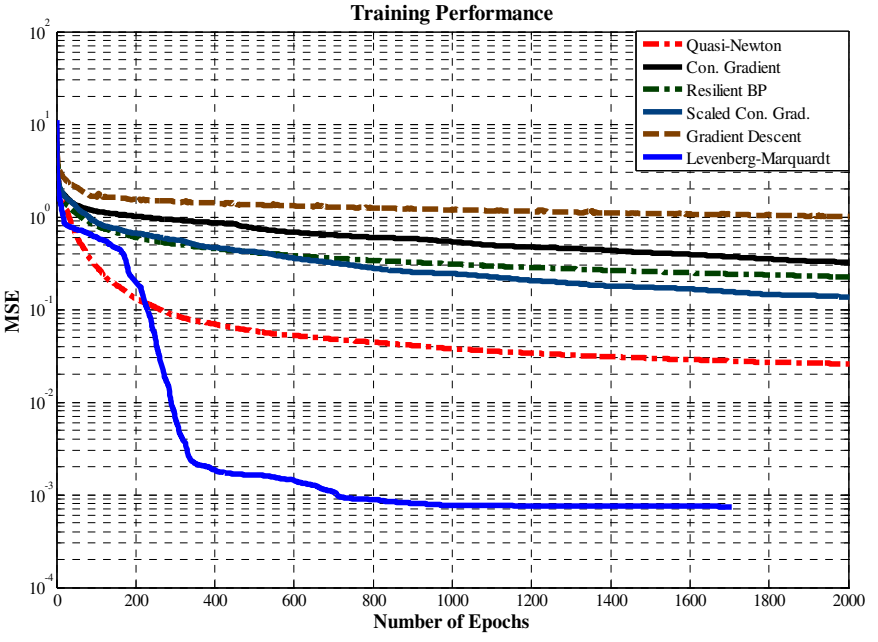
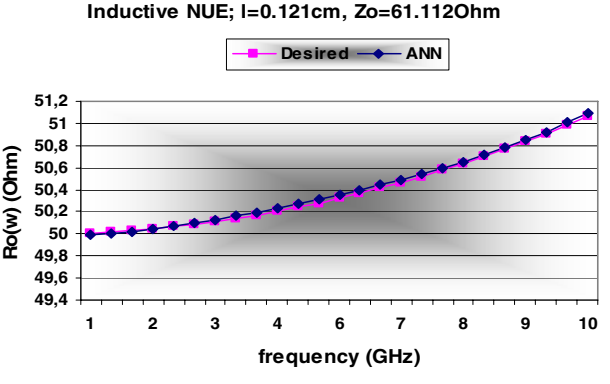


Fig. 3. Training Performance of MLP

5 Worked Examples

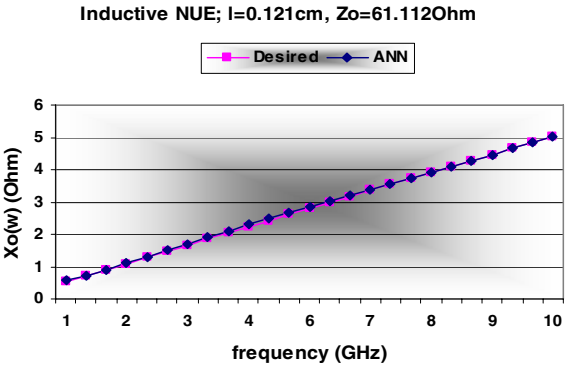
In worked examples, two types of driving of UE are considered:

(i) Z_S=50Ω ⇔ NUE in Fig. 4-5
(ii) Z_S=0 ⇔ NSE in Fig. 6-7.
(iii) Zs=211.78+j95.44Ω in Fig. 8



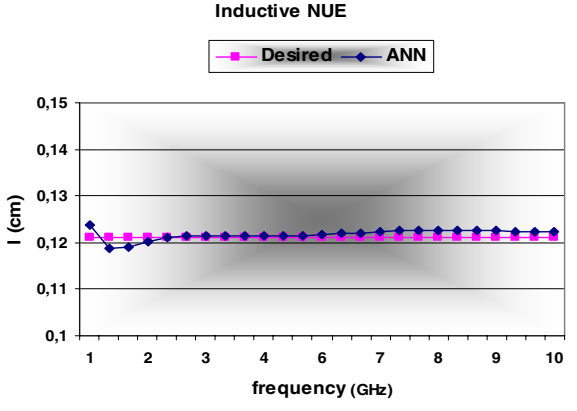
(a) $R_O(\omega)$

Fig. 4. Analysis of the capacitive NUE driven by Z_S=50Ω

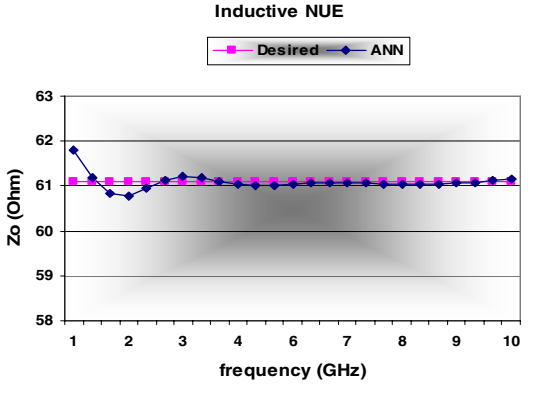


(b) $X_O(\omega)$

Fig. 4. (*continued*)



(a) Physical Length of UE



(b) Characteristic Impedance of UE

Fig. 5. Synthesis of the capacitive NUE

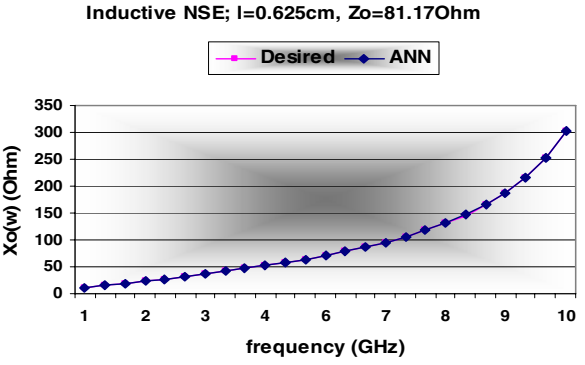
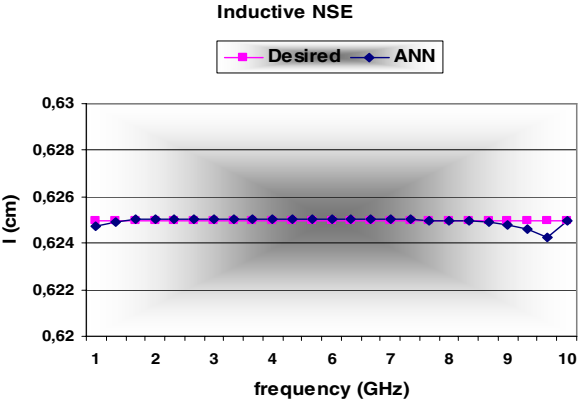
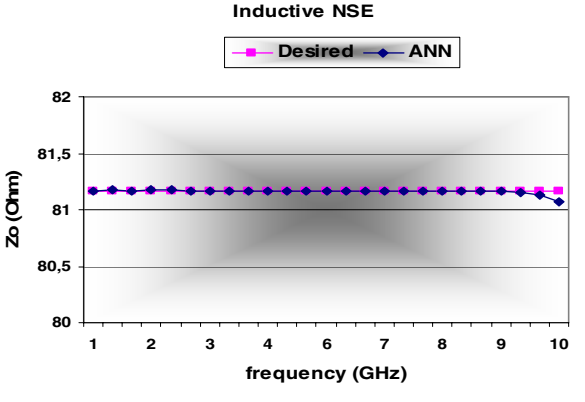


Fig. 6. Analysis of the inductive NSE

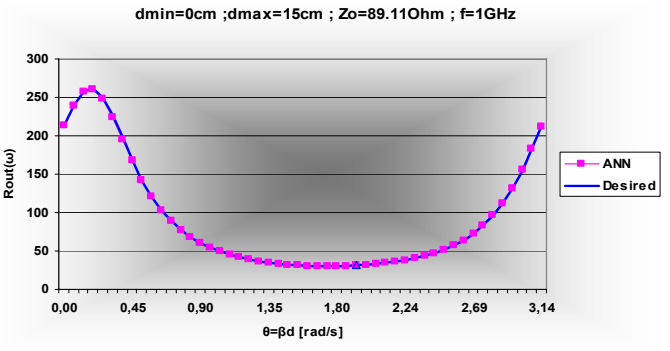


(a) Physical Length of UE

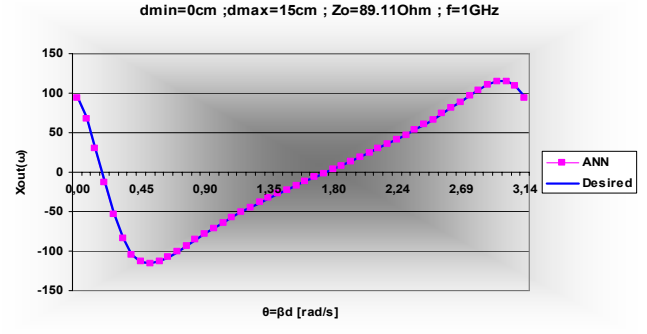


(b) Characteristic Impedance of UE

Fig. 7. Synthesis of the inductive NSE



(a) $R_O(\omega)$



(b) $X_O(\omega)$

Fig. 8. Analysis of the NUE driven by Z_S=211.78+j95.44Ω

6 Conclusions

In this work, Neural Unit Element (NUE) is defined in use analysis and synthesis of microwave circuits. For this aim, highly nonlinear transformation equations of the transmission line are solved by the neural networks. Further step can be analysis and synthesis of the more complicated microwave circuits such as matching circuits to provide necessary terminations of a microwave transistor.

References

1. F. Güneş, F. Gürgen and H. Torpi, "Signal-noise neural network model for active microwave device", IEE Proc-Circuits Devices and Systems; Vol., 143, pp. 1-8, 1996.
2. F.Güneş, H. Torpi and F. Gürgen, "A multidimensional signal-noise naural model for microwave transistor", IEE Proc-Circuits Devices and Systems; vol., 145(2), pp.111-117, 1998.

3. N.Türker, “Analysis and Synthesis of RF/Microwave Planar Transmission Lines with Artificial Neural Networks”, M. Sc. thesis, submitted to Yildiz Technical University, Department of Electronics and Communication Engineering, 2004.
4. Q.J. Zhang and K.C. Gupta, “Models for RF and Microwave Components”, Neural Networks for RF and Microwave Design, Norwood, MA: Artech House, 2000.
5. F. Güneş, C. Tepe, “Gain-Bandwith Limitations of Microwave Transistor”, RF and Microwave Computer-Aided Engineering, vol., 12, pp., 483-495, 2002.
6. Y. Cengiz, “Design of The Microwave Amplifier with The Optimum Performance”, Ph. D. thesis, submitted to Yildiz Technical University, Dep. of Electronics and Communication Eng., 2004.
7. M.F. Çağlar, F. Güneş, “Neural Networks as a Nonlinear Equation Set Solver in Analysis and Synthesis of a Microwave Circuits”, INISTA’2005, Istanbul, pp. 103-107, June 2005.
8. K. Hornik, M. Stinchcombe and H. White, “Multilayer Feedforward Networks are Universal Approximators”, Neural Networks Vol. 2., pp. 359-366, 1989.
9. G. Cybenko, “Approximation by Superpositions of a Sigmoidal Function”, Math. Control Signals Systems, Vol. 2, pp. 303-314, 1989.

An Artificial Neural Network Based Simulation Metamodeling Approach for Dual Resource Constrained Assembly Line

Gokalp Yildiz and Ozgur Eski

Dokuz Eylul University, Faculty of Engineering, Industrial Engineering Dept.,
35100 Bornova-Izmir, Turkey
{gokalp.yildiz, ozgur.eski}@deu.edu.tr

Abstract. The main objective of this study is to find the optimum values of design and operational parameters related to worker flexibility in a Dual Resource Constrained (*DRC*) assembly line considering the performance measures of Hourly Production Rate (*HPR*), Throughput Time (*TT*) and Number of Worker Transfers (*NWT*). We used Artificial Neural Networks (*ANN*) as a simulation metamodel to estimate *DRC* assembly line performances for all possible alternatives. All alternatives were evaluated with respect to a utility function which consists of weighted sum of normalized performance measures.

1 Introduction

Fast and dramatic changes in customer expectations, competition, and technology cause uncertain environments in the global markets of the twenty first century. The success of a manufacturing firm in an uncertain environment depends on how effectively the firm responds to these changes in the markets. In general, the researchers and the manufacturing managers contend that the most important concept to cope with this uncertainty is the manufacturing flexibility. The manufacturing flexibility is the ability of the firm to manage production resources and uncertainty to meet customer requests [16]. Especially, in *DRC* Systems, the key element is the worker flexibility which is one of the dimensions of manufacturing flexibility. A *DRC* production system is one in which all equipment in the shop is not fully staffed and, furthermore, the workers can be transferred from one piece of equipment to another as needed [12].

Two important decisions should be made when the system is *DRC*; (1) when to move workers, (2) Where to move workers. This is called when/where rule pairs in *DRC* systems. Typical "when" rules in *DRC* literature are Decentralized and Centralized. The centralized rule provides the maximum worker flexibility because this rule implies that the worker is available for transfer whenever a job is completed. However, if worker transfer delays are significant, effective worker capacity might decrease dramatically due to the increased number of transfers. The decentralized rule implies that the workers are eligible for transfer when there is no job waiting to be processed at their current workstations. This rule is used especially when the transfer

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 1002 – 1011, 2006.

times are significant. Due to the ease in their implementation, these rules have received a lot of attention from researchers [12,14]. Fryer extended these rules to a parametric version [4]. In his parametric version of "when" rule, the workers are eligible for transfer to another workstation only when there is maximum of "*q*" jobs in their queues. Other types of "when" rules can be found in [5,9,13]. Typical "where" rules in *DRC* literature are First in System First Serve (*FISFS*), Largest Number in Queue *(LNQ*) and Earliest Due Date (*EDD*), etc. However, their effects on the shop performance have not been reported as extensively as "when" rules [14].

In the most of these studies, simulation has been used to analyze the effects of when/where rule pairs on the performance measures since it is a very flexible tool in modeling and analysis of such complex systems. In simulation modeling, a set of inputs are used to estimate a set of output performance measures. This process is repeated until a satisfactory level of performance measures is obtained. Hence, the simulation modeling becomes a trial and error process [2]. The iterative nature of this process may result in high computational costs and difficulties in prediction of the performance measures. Simulation metamodels are generally used in order to overcome these problems. The main objective of a simulation metamodel is to accurately represent the input and output relationships. In metamodeling, simulation is used to generate data set for construction of metamodels. In general, regression analysis has been combined with simulation for building simulation metamodels. The use of artificial neural networks is another approach for metamodeling. A neural network is a proven tool in providing excellent response predictions in a wide range of application areas and it outperforms regression analysis [6].

A neural network based simulation metamodel is a neural network whose training set (*i.e.*, input-output pairs) is provided by a simulation model. Pierreval [10] used neural networks to model simulation of manufacturing shops. Kilmer et al. [8] described the use of supervised neural networks as a metamodeling technique for discrete, stochastic simulation. Hurrion [7] developed a neural network metamodel to search for the optimal kanban combination for a two-station pull system. Savsar and Choueiki [11] extended the study of Hurrion and proposed a Generalized Systematic Procedure that integrates experimental design concepts with simulation and neural networks for solving kanban allocation problem. Araz et al. [1] proposed a multi-criteria decision making methodology based on neural networks for kanban allocation problem. Fonseca et al. [3] discussed the importance of simulation metamodeling through artificial neural networks and provided general guidelines for the development of *ANN* based simulation metamodels. The results of these studies indicated that simulation metamodels with neural networks can be effectively used for estimation of the system performance. However, constructing a neural network is also time consuming since the process requires generating a training set.

The objective of this study is threefold: (1) *To apply Fryer's parametric "when" rule to a special assembly line which produces electrical motors.* Since the manufacturing system under consideration is *DRC*, how to move the workers is important. Hence, the Fryer's parametric "when" rule was modified and applied to the special assembly line. (2) *To find the optimum values of design and operational parameters related to worker flexibility*. Since the worker transfer decisions directly

affects the system performance, it is critical to find the optimum values of design and operational parameters related to worker flexibility. In this study, a neural network based simulation metamodeling approach was used in order to obtain optimum parameter configurations with respect to a utility function which consists of weighted sum of normalized performance measures. (3) *To propose and to implement an integrated framework to prepare training set for ANN based simulation metamodeling*. As stated above, in *ANN* based simulation metamodeling, the construction of the training set is a time consuming activity since the training set is provided by simulation models. Furthermore, making required modifications on simulation models becomes impractical when the size of the training set gets larger. To deal with this problem, we developed an integrated framework which consists of a parametric simulation model and a Training Set Generator. This integrated framework automates the process of generating the training set and building simulation models. It also gives flexibility to increase the size of the training set.

The paper is structured as follows; in Section 2, the overview of the assembly line is explained. In Section 3, the solution methodology is given and in the final section, the conclusions are presented.

2 System Overview

As it is depicted in Figure 1, the assembly line which produces electrical motors consists of five workstations (*i.e.*, S_1, S_2, S_3, S_4 and S_5). The parts visit the workstations sequentially and are transferred from one workstation to another on pallets by the loop-conveyor, c_L. The total number of pallets in this system is denoted by K. It must be noted that the level of K limits the level of Work-in-Process (*WIP*). The transfers between c_L and the workstations are provided by the sub-conveyors $c_1,c_2,\ldots,c_8$. Each workstation has its own input/output buffers except for S_5. The workstation S_5 has only its input buffer which is same as the output buffer of S_4. The capacities of input/output buffers are denoted as i_1, o_1, i_2, o_2, i_3, o_3, i_4, and o_4=i_5. The worker capacities of workstations are presented by w_1, w_2, w_3, w_4, and w_5. For example, if the worker capacity of S_1 is equal to 2 (*i.e.*, w_1=2), it means that two workers can work together at workstation S_1.

In this assembly line, each pallet may be in one of four possible states (*i.e.*, empty pallet (*STATE*1), the pallets carrying the parts processed at S_1 (*STATE*2), S_2 (*STATE*3), and S_3 (*STATE*4)). The parts processed at S_4 are removed from their pallets, and then send to the input buffer of S_5. In addition to this, the empty pallets which are removed from the parts try to access c_8 to turn back to c_L. The states of pallets are changed when the worker at a particular workstation finishes the related operation on a part. So, as it is illustrated in Figure 1, the pallets in *STATE*1, *STATE*2, *STATE*3, and *STATE*4 try to enter the related workstations from the points *A*, *C*, *E*, and *G*, respectively. Then, following the state changes, they try to access to c_L from the related points (*i.e.*, *B*, *D*, *F*, and *H*) to be transferred to workstations for consecutive operations. If there is not available space at the input buffer of the workstation, the pallet keeps moving through the loop-conveyor until there is available space at related workstation.

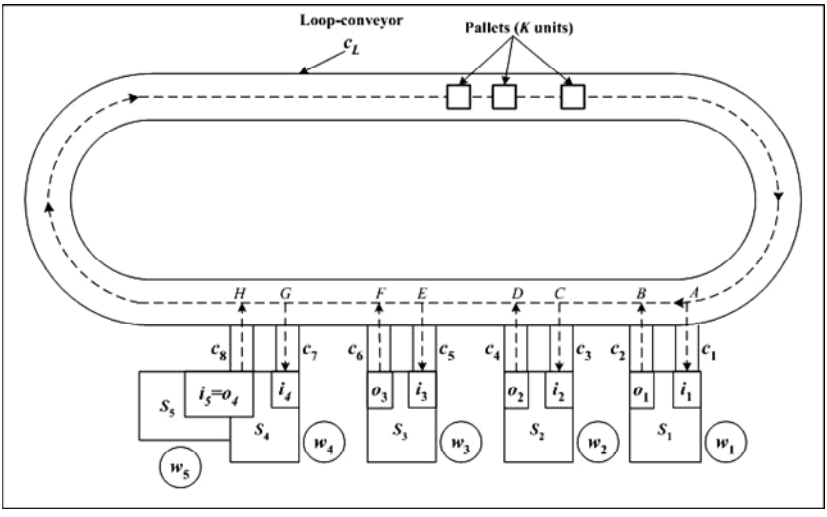


Fig. 1. The overview of the assembly line

The explanations given so far are related to the movements of parts in the system. Due to the reason that the system is *DRC*, how to move the workers (*i.e.*, typical when/where rule pairs) on this kind of system is also important. So, as mentioned before, the following "when" rule is applied in this system; "the workers are eligible for transfer to another workstation only when there is maximum of q jobs in their queues" as Fryer stated [4]. The q is a threshold value that triggers the worker to move to another workstation if it is needed. This rule represents the general case of decentralized and centralized rules. It is general because when q is set to zero, the resulted rule will be decentralized, when q is set to K (*i.e.*, the number of pallets), the resulted rule will be centralized. Increasing the value of q from 0 to K, increases the degree of worker flexibility (*i.e.*, the number of worker transfers). To be able to apply Fryer's rule in this system, the threshold value q was applied to the number of pallets in *STATE*1, *STATE*2, *STATE*3, and *STATE*4. Since the number of pallets in different states represents the total workload of the related workstations, the answer of the "where" question becomes "to the workstation with the greatest workload". For example, if the number of pallets in *STATE*1 (*i.e.*, empty pallets) is the greatest, it means that the workload of S_1 is greatest, so the worker who is eligible for transfer at that time should move to the workstation S_1 if it is not fully staffed.

When the worker is eligible for transfer, he can move from his current workstation to another workstation under the following conditions; (1) if the workload of the current workstation is lower than the workload of the target workstation. (2) If the target workstation is not fully staffed.

The processing times at workstations S_1, S_2, S_3, S_4 and S_5 are uniformly distributed with the minimum and maximum of 29.4-37.8, 81.6-93.6, 62.4-108.3, 62.4-80.4 and 36.6-64.25 seconds, respectively. The velocities of conveyors are 48 pallets/minute. The worker capacity of S_5 (*i.e.*, w_5) is assumed constant, and it is equal to one worker. This worker remains at S_5 while the system is operating. The total number of workers in this system is five.

3 Solution Methodology

We used a solution methodology which consists of five steps to find the optimum values of design and operational parameters related to worker flexibility in a *DRC* assembly line. In the first step, the parametric simulation model of the system was developed. In the second step, the decision variables and important performance measures were identified. In the third step, training set was selected and their related simulation models were constructed. Then, in the fourth step, the design parameters of *ANN* model for each performance measure was identified and the *ANN* models were built, trained, validated and tested. In the last step, using the trained *ANN* models, the performance measures of all possible alternatives were predicted and evaluated.

3.1 Building the Parametric Simulation Model

The *DRC* assembly line given in Figure 1 was modeled by using simulation software, ARENA 10.0, which is highly flexible tool in modeling this type of systems consisting complex interactions. Since the generation of simulation models related to selected alternatives is time consuming when the size of the training set is higher, the simulation model was built parametrically in order to accomplish the related modifications easily.

3.2 Identification of Decision Variables and Performance Measures

There are six decision variables to be considered in this study. These are K, w, P_1, P_2, P_3, P_4. As mentioned before, the K and w represent the total number of pallets on the conveyor system and the worker capacities of workstations, respectively. In our system, it is assumed that the values of w_1,w_2,w_3, and w_4 are equal and presented by w. The decision variable P_i is a threshold value related to "when" rule for workstation S_i.

In this study, each alternative consists of different levels of decision variables K, w, P_1, P_2, P_3, and P_4. For example, the alternative (28, 3, 8, 10, 6, 4) represents the following configuration; there are 28 pallets in the system, maximum 3 workers can work at a workstation simultaneously, and at the workstations S_1,…,S_4, workers will be eligible for transfer when their workloads are less than or equal to 8, 10, 6 and 4, respectively. Based on the results of pilot studies and our previous study [15], the lower and the upper levels of decision variables were defined as shown in Table 1.

Table 1. The levels of decision variables

Decision Variable	Low Level	High Level	Step Size
$\boldsymbol{K}$	10	32	2
$\boldsymbol{w}$	2	4	1
$\boldsymbol{P_i}$	0	$K/2$	2

The performance measures considered in this study are hourly production rate of the line (*HPR*), throughput time (*TT*), and the number of worker transfers (*NWT*). The *HPR* is defined as the average number of electrical motors produced in an hour. The

TT represents the average total time (minutes) that a part spends between the workstations S_1 and S_5. The *NWT* shows the total number of worker transfers realized in a given period. Although, it is assumed that the worker transfer times are negligible in this system, the *NWT* can be used to evaluate the alternatives according to the possible losses in effective worker capacity.

According to the results of our previous study [15], the parameter *K* which also represents the *WIP* level has the greatest effect on both *HPR* and *TT*. So, increasing the number of pallets in this kind of *DRC* system increases the *HPR* but causes higher throughput times and *WIP* levels. Increasing the level of *w* causes an increase in *NWT*. It is clear that it would be costly if the worker transfer times are significant. Hence, it is desired to achieve a compromise solution by maximizing *HPR* while minimizing *TT* and *NWT*.

Considering Table 1, the search space includes 91410 alternatives. Running all 91410 simulations would approximately require 3050 hours (each simulation run requires approximately 2 minutes of input/output and computation time). Instead of simulating all configurations, we used an *ANN* based simulation metamodeling approach for estimating the performance measures of all alternatives.

3.3 Construction of Training Set

Typical *ANN* based simulation metamodeling study requires three main steps for data preparation process; the first step is to select the alternatives (*i.e.*, inputs) to be used for the training set. The second step is to build the simulation models of these selected alternatives. The last step is to run these simulation models to obtain the performance measures (*i.e.*, outputs). We automated these steps by developing an integrated environment which we named as Training Set Generator (*TSG*). *TSG* consists of three main modules named Configuration Generator, Simulation Model Generator, and Output Data Processor. All modules of *TSG* were coded in *C*. The framework for *TSG* can be seen in Figure 2.

In the first step of data preparation process, our Configuration Generator Module divides the search space into 36 regions according to the levels of *K* and *w* (*i.e.*, $K \in \{10,12,\ldots,32\}$, $w \in \{2,3,4\}$, the total number of region is 12*3= 36). The number of alternatives to be taken from a region is calculated by multiplying the total size of the training set by the percentage of alternatives in that region to total number of alternatives. For example, if the levels of *K* and *w* are fixed at 10 and 2, the total alternatives in this region will be 4^4=256 (*i.e.*, $P_i \in \{0, 2, 4, 6\}$, i:1,...,4). Since we have 91410 total alternatives, setting the size of the training set to 5000, the number of total samples to be taken from this region is 5000*256/91410≈14.

In the second step of data preparation process, the list of selected alternatives to be used for training the *ANN* is sent to Simulation Model Generator Module. This module automatically makes the required changes on Default Simulation Model of the system. The output of this generator is the simulation models of the selected alternatives. Handling this process manually is very time consuming and impractical when the size of the training set gets larger. The Simulation Model Generator Module allows us to increase the size of the training set.

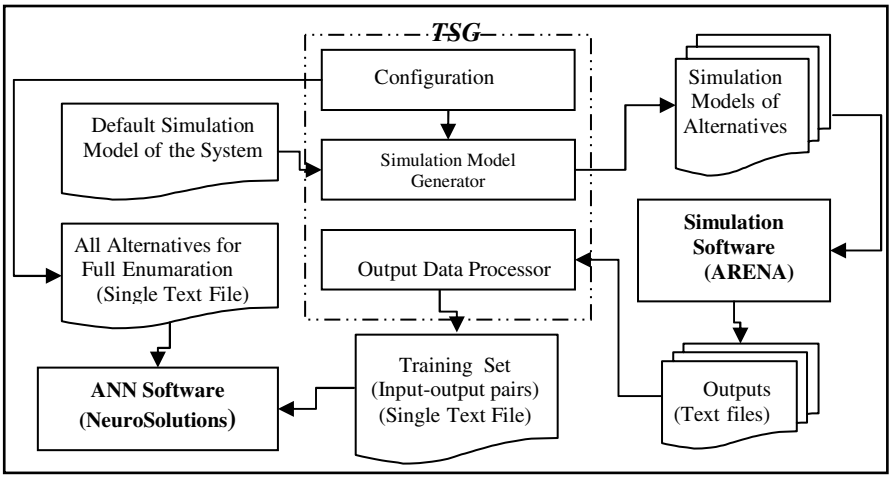


Fig. 2. Framework for Training Set Generator (*TSG*)

In the third step of data preparation process, the simulation models generated by the Simulation Model Generator Module are executed by ARENA 10.0 Simulation Software and the replication results for the performance measures are recorded in a text file for each alternative. The Output Data Processor Module processes these text files and constitutes a single text file including inputs and related average performance measures. This text file is the input of *ANN* software.

For simulation models, the total replication length is equal to 30 hours including 3 hours of warm up period. The number of independent replications is 5 for each alternative. The number of alternatives to be generated among all 91410 alternatives by the *TSG* is set as 5000. This number includes data for training, validating, and testing. Running 5000 simulations would approximately require 170 hours (each simulation run requires approximately 2 minutes of input/output and computation time) without using *TSG*. We obtained input-output pairs for 5000 alternatives in approximately 16 hours with *TSG*.

3.4 Building, Training and Testing of Neural Network Models

A feed-forward architecture including one layer of hidden units has been employed for each performance measure (*HPR*, *TT* and *NWT*).The assembly line parameters *K*, *w*, P_1, P_2, P_3 and P_4 were introduced as input nodes for each network. In each *ANN*, output layer consists of one node that estimates an output measure.

The performance of an *ANN* depends on several design parameters such as number of hidden layers (*HL*), the number of processing elements (*PE*) in hidden layers, the type of transfer function, learning rate and momentum rate etc. Selection of these parameters is generally based on trial-and-error. Using trial-and-error, the design parameters of *ANN*s were selected as in Table 2. The *ANN* models were developed using NeuroSolutions 5.0 software.

Training is an important feature of *ANN*s. After the network reaches a satisfactory level of performance, it learns the relationship between input factors and simulation

Table 2. Design parameters of *ANN* models

ANN Model	# of Input Node	Output Nodes	# of *HL*	# of *PE* in *HL*	Type of Transfer Function	Learning Rate	Momentum Rate
1	6	1(*HPR*)	1	52	*Tanh*	0.1	0.80
2	6	1(*TT*)	1	56	*Tanh*	0.1	0.90
3	6	1(*NWT*)	1	54	*Tanh*	0.1	0.85

responses. Then the trained network can be used to estimate the simulation outputs. Back-propagation (*BP*) learning algorithm is used in this study. *BP* is the most widely used network learning algorithm. The *BP* learning involves three stages: the feed-forward of the input training pattern, the calculation of the associated error, and the adjustment of the weights. For our *ANN* models, 4250 input-output pairs are used for training. 250 different pairs are used for cross validation and 500 are used for testing. The networks were trained to an error tolerance 0.001 or to a maximum of 3000 epochs.

An essential aspect of metamodeling is to evaluate the quality of the performance measures produced by *ANN*s as compared with the performance measures produced by simulation model (*i.e.*, true value). The mean square error and the percentage error were used as the measure of accuracy of the neural networks. As seen in Table 3, *ANN*s provide estimates for *HPR*, *TT* and *NWT* accurately.

Table 3. Measures of accuracy for *HPR*, *TT*, and *NWT*

Measure of Accuracy	*HPR*	*TT*	*NWT*
MSE	0.016	0.046	0.080
% Error	4.14	4.63	9.69

3.5 The Full Enumeration and Evaluation of the Alternatives

The *HPR*, *TT* and *NWT* for all alternatives are obtained by using three neural network models given in previous section. As mentioned before, our objective is to achieve a compromise solution by maximizing *HPR* while minimizing *TT* and *NWT*. Since we have more than one objective, the Simple Additive Weighted (*SAW*) method, which is one of the simplest and the most popular Multi-Objective Decision Making method, was used for selecting the best alternative that satisfies the decision maker's requirements. The *SAW* method uses the following equation to evaluate the utility value of the x^{th} alternative (U_x);

$$U_x = \sum_{i=1}^{m} w_i Y_{ix} \tag{1}$$

where,
U_x: The utility value of x^{th} alternative, x:1,…,v
v: The total number of alternatives

m: The number of objectives
w_i: The weight for i^{th} objective
Y_{ix}: The normalized value of i^{th} objective for x^{th} alternative

The performance measures were normalized between -1 and 1. Using Equation 1 with a weight set of w_{HPR}=0.5, w_{TT}=0.30, w_{NWT}=0.20, the utility value of each alternative was obtained. Then, the alternatives were ranked according to their utility values. The five alternatives with the best utility values were presented in Table 4.

Table 4. The best alternatives for the weight set w_{HPR}=0.5, w_{TT}=0.30, w_{NWT}=0.20

Alternative No	(K,w,P_1,P_2,P_3,P_4)	*HPR*	*TT*	*NWT*	*U*
1	(18,2,8,10,8,10)	50.329	20.472	764.28	0.7506
2	(18,2,10,10,8,10)	50.081	20.907	702.62	0.7490
3	(18,2,8,10,8,8)	50.188	20.419	764.41	0.7471
4	(18,2,10,10,8,8)	49.812	20.903	704.17	0.7411
5	(18,2,8,10,10,10)	50.184	20.701	789.94	0.7402

As seen in Table 4, for the best alternative, the values of parameters K, w, P_1, P_2, P_3, and P_4 are 18, 2, 8, 10, 8, and 10, respectively. According to this alternative; (1) the number of pallets in this *DRC* assembly line should be 18. (2) The worker capacity of each workstation should be 2, which means that maximum two workers can work simultaneously at a workstation. (3) The threshold values of Fryer's parameters for workstations S_1, S_2, S_3, S_4 are 8, 10, 8, 10, respectively. This implies that a worker becomes eligible for transfer to another workstation when the workload of that worker is equal or less than the corresponding threshold values. It must be noted that the threshold value of the second workstation is at its maximum level since it is the bottleneck workstation of this *DRC* assembly line. We simulated this assembly line for the static case where the worker movement is not allowed (*i.e.*, w=1) for 10 replications. The *HPR* and the *TT* levels were obtained as 41.107 units/hour and 25.593 minutes, respectively. Hence, averagely 22.43% and 20% improvement in *HPR* and *TT* may be obtained when the best alternative is applied. It is obvious that a decision maker can find different solutions by using different weight structures that reflect her/his preferences.

4 Conclusions

In *DRC* Systems, the key element is the worker flexibility which is one of the dimensions of manufacturing flexibility. In this type of manufacturing systems, two important decisions should be made; "when" and "where" to move workers. In this study, we modified the Fryer's parametric "when" rule and applied it to a special *DRC* assembly line. To optimize the design and operational parameters related to worker flexibility, we used *ANN* based simulation metamodeling which allows us to estimate the performance measures of all alternatives in a reasonable time. All alternatives were evaluated with respect to a utility function which consists of

weighted sum of normalized performance measures. We also proposed and implemented an integrated framework which automates the process of generating the training set and building simulation models.

References

1. Araz, U. O., Eski, O., Araz, C.: A Multi-Criteria Decision Making Procedure Based on Neural Networks for Kanban Allocation. In: ISSN 2006. Lecture Notes in Computer Science **3973** (2006) 898–905
2. Fasihul, M. A., Ken, R. M., Trevor, J. R.: A Comparison of Experimental Designs in the Development of a Neural Network Simulation Metamodel. Simulation Modeling Practice and Theory **12** (2004) 559–578
3. Fonseca, D. J., Navaresse, D.O., Moynihan, G.P.: Simulation Metamodeling Through Artificial Neural Networks. Engineering Applications of Artificial Intelligence **16** (2003) 177–183
4. Fryer, J. S.: Labor Flexibility in Multiechelon Dual Constrained Job Shops. Management Science **20** (1974) 1073-1080
5. Gunther, R. E.: Server Transfer Delays in A Dual Resource Constrained Parallel Queuing System. Management Science **25** (12) (1979) 1245-1257
6. Hornik, K., Stinchcombe, M.W.H.: Multilayer Feedforward Networks are Universal Approximators. Neural Networks **2** (1989) 359–366
7. Hurrion, R.D.: An Example of Simulation Optimization Using A Neural Network Metamodel: Finding the Optimum Number of Kanbans in Manufacturing System. Journal of the Operation Research Society **48** (1997) 1105-1112
8. Kilmer, R., Smith, A., Shuman, L.: An Emergency Department Simulation and a Neural Network Metamodel. Journal of the Society for Health Systems **5** (3) (1997) 63–79
9. Nelson, R.T.: Labor and Machine Limited Production Systems. Management Science **13** (9) (1967) 648-671
10. Pierreval, H.: Training a Neural Network by Simulation for Dispatching Problems. In: Proceedings of the Third Rensselaer International Conference on Computer Integrated Engineering, (1992) 332–336
11. Savsar, M., Choueiki, M.H.: A Neural Network Procedure for Kanban Allocation in JIT Production Control Systems. International Journal of Production Research **38** (2000) 3247-3265
12. Treleven, M. D.: A Review of the Dual Resource Constrained System Research. IIE Transactions **21** (3) (1989) 279-287
13. Treleven, M. D.: The Timing of Labor Transfers in Dual Resource-Constrained Systems: Push vs. Pull Rules. Decision Sciences **18** (1) (1987) 73-88
14. Treleven, M. D., Elvers, D. A.: An Investigation of Labor Assignment Rules in a Dual-Constrained Job Shop. Journal of Operations Management **6** (1) (1985) 51-67
15. Yıldız, G.: An Application of Typical When/Where Rule Pairs in Dual Resource Constrained (DRC) Assembly Line with Closed-Loop Conveyor. In: Proceedings of 35th International Conference of Computers and Industrial Engineering (2005) 2185-2190
16. Zhang, Q., Vonderembse, M.A. Lim, J.S.: Manufacturing Flexibility: Defining and Analyzing Relationships among Competence, Capability, and Customer Satisfaction. Journal of Operations Management **21** (2) (2003) 173-191

A Packet Routing Method Using Chaotic Neurodynamics for Complex Networks

Takayuki Kimura[1,2] and Tohru Ikeguchi[1]

[1] Graduate school of Science and Engineering, Saitama University,
255 Shimo-Ohkubo Saitama 338–8570, Japan
[2] kimura@nls.ics.saitama-u.ac.jp

Abstract. We propose a new packet routing method for a computer network using chaotic neurodynamics. We first compose a basic neural network which routes packets using information of shortest path lengths from a node to the other nodes. When the computer network topology is regular, the routing method works well, however, when the computer network topology becomes irregular, the basic routing method doesn't work well. The reason is that most of packets cannot be transmitted to their destinations because of packet congestion in the computer network. To avoid such an undesirable problem, we extended the basic method to employ chaotic neurodynamics. We confirm that our proposed method exhibits good performance for computer networks with various topologies. Furthermore, we analyze why the proposed routing method is effective: we introduce the method of surrogate data which is often used in the field of nonlinear time-series analysis. In consequence of introducing such a statistical control, we confirm that using chaotic neurodynamics is the most effective policy to decentralize the congestion of the packets in the computer network.

1 Introduction

Recently, information-technologies have been developed exponentially. One of typical examples is the Internet. We can get a wide variety of information instantaneously using the Internet. In a computer network such as the Internet, large amounts of packets of various sizes are flowing. However, they are often delayed or lost. To avoid such undesirable situation, it is very important to route the packets in the computer network.

The packet routing strategies can be generally classified into two categories. The first one is a centralized control. The centralized control is a strategy that a centralized unit controls all packet routing in the network. When the network size not so large, the centralized unit can accumulate a state of the whole computer network and decide which node is most appropriate to transmit packets among all adjacent nodes. However, in a large-scale network, the centralized control often fails to work well because the central unit has huge computational load. Therefore, it is impossible to route most of the packets to transmit their destinations realistically using the centralized control.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 1012–1021, 2006.

The second strategy is a decentralized control. The decentralized control is a more realistic strategy than the centralized control for large-scale networks because each unit transmits packet autonomously and adaptively.

A computer network comprises nodes and links. A packet is transmitted from one node to another through the links. A packet can be transmitted from the nodes and multiple packets can be received simultaneously. Every node stores some amounts of packets in a buffer and all packets are transmitted according to First-In-First-Out basis. Then, when a buffer of the node is full, the packet transmitted to the node will be removed. In addition, the packet flow is regulated by an upper limit. Thus, every packet is also removed if it exceeds this limit. When a packet is removed, the packet is resent from its source until it will be transmitted to the destination of the packet.

In an ideal computer network, every node has an infinite buffer size and throughput. In such a network, the Dijkstra algorithm[1], one of the basic strategies to find a shortest path of the network, may work well[1]. However, under a real situation, the buffer sizes are finite and the throughputs at each node are different, which eventually leads to congest the route to transmit packets. Then it is inevitable to consider how to avoid such congested routes. It means that an ideal packet routing problem is easy to be solved, but real packet routing problems probably become very difficult and possibly may belong to a hard class.

As for solving $\mathcal{NP}$ hard class combinatorial optimization problems, for example, the traveling salesman problems (TSP) or the quadratic assignment problems (QAP), it is widely acknowledged that a method using chaotic neurodynamics is very effective[2, 3, 4, 5]. The method[2, 3, 4, 5] extends a tabu search strategy[6, 7], which avoids a solution that has already been searched for a while. If the strategy is modified to involve chaotic neurodynamics[2, 3, 4, 5], the algorithm exhibits better performance not only for bench mark problems of TSP or QAP, but also for real life problems such as bipartitioning problems[8], motif extraction problems from DNA sequences[9] and time tabling problems[10].

In the present paper, we propose a new routing-packets method to introduce such techniques[2, 3, 4, 5, 8, 9, 10]. We confirm that our proposed method is very effective not only for regular networks but also for randomized networks[11] and scale-free networks[12] in comparison with the Dijkstra algorithm. Furthermore, we analyze the effectiveness of the proposed routing method, applying the method of surrogate data: a statistical hypothesis testing frequently used in nonlinear time-series analysis. In our analysis, we use the method of surrogate data as a statistical control to produce a surrogate time-series which has the same statistics as internal states which correspond to the origin of chaotic neurodynamics. We confirm that using chaotic neurodynamics is the most effective policy to decentralize packet congestion in the computer networks.

2 Routing Method Using Chaotic Neurodynamics

In order to realize chaotic neurodynamics, we introduced a chaotic neural network[13]. First, we consider a computer network model which has N nodes.

In the computer network, the i-th node has N_i adjacent nodes ($i = 1, \ldots, N$). In this framework, each node has its own neural network, and N_i adjacent neurons are assigned to each node. The ij-th neuron corresponds to the connection between the i-th node and its j-th adjacent node. We first compose a basic neural network which operates to minimize a distance of transmitting packets from the i-th node to the destinations. To realize this method, we consider the following internal state of the ij-th neuron:

$$\xi_{ij}(t+1) = \beta \left(1 - \frac{d_{ij} + d_{jo}}{d_c} \right), \tag{1}$$

where d_{ij} is the distance from the i-th node to the j-th adjacent node; d_{jo} is the distance from the j-th adjacent node to the destination of the i-th node; d_c is a control parameter which expresses the size of the computer network; β is a normalization parameter. If $\xi_{ij}(t+1)$ is the largest value in the neurons of the i-th node, the ij-th neuron fires, which means that the j-th adjacent node is selected to transmit a packet from the i-th node. The decent down-hill dynamics of Eq.(1) corresponds to the basic Dijkstra algorithm[1] and works well for the ideal case.

However, under real circumstances we have to consider both network topologies and packet congestion at nodes. If the network topology is not regular, the number of links of each node is biased. In addition, the number of routes through which the packets are transmitted to the destinations also increases. When we conduct a packet routing for an irregular network, if we only consider to minimize the shortest distance, many packets might be transmitted to the nodes which are connecting many adjacent nodes. This behavior leads to delay or lost packets. To avoid such an undesirable situation, we introduce a refractory effect of a chaotic neuron model[13] described as follows:

$$\zeta_{ij}(t+1) = -\alpha \sum_{d=0}^{t} k_r^d x_{ij}(t-d) + \theta, \tag{2}$$

where α is a control parameter of the refractoriness; k_r is a decay parameter of the refractoriness; $x_{ij}(t)$ is the output of the ij-th neuron at time t; θ is a threshold.

The refractory effect plays an important role for how to decentralize the packets in the adjacent nodes. Because the refractory effect is related to the information of a past routing history, we expect that the packets are transmitted to their destination by avoiding the nodes which packets have just been transmitted to and which possibly have already stored many packets.

In addition, we use a mutual connection to control firing rates of neurons, because too frequent firing often leads to a fatal situation of the packet routing. The mutual connection is defined as follows:

$$\eta_{ij}(t+1) = W - W \sum_{j=1}^{N_i} x_{ij}(t), \tag{3}$$

where W is a positive parameter.

Then, the output of the ij-th neuron is defined as follows;

$$x_{ij}(t+1) = f\{\xi_{ij}(t+1) + \zeta_{ij}(t+1) + \eta_{ij}(t+1)\}, \tag{4}$$

where $f(y) = 1/(1+e^{-y/\epsilon})$. In this algorithm, if $x_{ij}(t+1) > 1/2$, the ij-th neuron fires; the packet at the i-th node is transmitted to the j-th node. If the outputs of multiple neurons exceed $1/2$, we defined that the neuron whose output is the largest only fires.

3 Evaluation of the Proposed Method in Some Networks with Various Topology

We compared the proposed method with the Dijkstra algorithm and a packet routing method using a neural network (the NN method) for the randomized networks[11] and the scale-free networks[12]. The NN method is a routing method that each neuron has only the gain effect which is defined by Eq.(1).

We conducted computer simulations of the packet routing by the following procedures. First, we assigned random values from one to five, which correspond to throughputs at all nodes. In addition, each node calculates the shortest distance from the node to the other nodes. In other words, each node has a routing table which contains information of shortest distances. Next, we generated packets randomly at all nodes in the computer network using uniformly distributed random numbers; each packet has a destination and the destinations are assigned randomly using uniformly distributed random numbers. Then, the link selection is simultaneously conducted at every node. We set the buffer size of the i-th node to $1,000$ times of the number of adjacent nodes of the i-th node. We also set the upper limit of the packet movement to 20. A packet is removed when the buffer is full and the packet exceeds the limit. The packet is resent from a source to its destination until it will be delivered to the destination.

We repeated the link selection and packet transmitting for $2,048$ iterations. We fixed the total number of packets in the network. When the packet arrived at its destination, we added a new packet. Then, a source and its destination of the new packet are randomly decided again using uniformly distributed random numbers. We set the parameters of Eqs.(1)–(3) as follows: $\beta = 1.5$, $\alpha = 0.045$, $k_r = 0.98$, $\epsilon = 0.05$ $W = 0.05$ and $\theta = 0.5$. We also set d_c as the longest path length in the network.

To evaluate performance of the proposed method, we introduced the following quantities:

N_p : an average number of packets at each node,
N_a : the number of packets arriving at their destinations,
T : an average arrival times of packets arriving at their destinations.

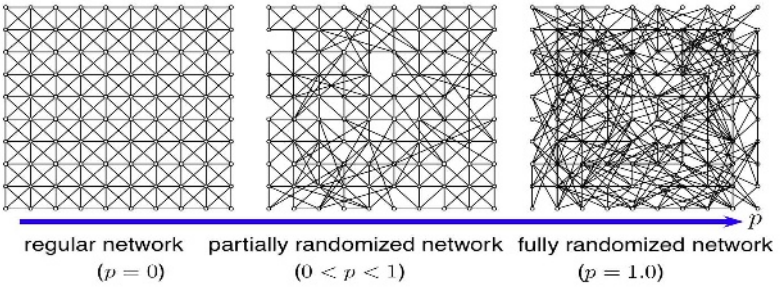


Fig. 1. The regular network (left) and a fully randomized network(right), and a partially randomized network. We rewired each link in the regular network based on the rewiring probability(p). We can produce some different types of the networks; the regular network corresponds to $p = 0$, a fully randomized network corresponds to $p = 1.0$. In the above example of the partially randomized network, $p = 0.2$.

$$N_p = \frac{N_{tp}}{N}, T = \frac{T_{tp}}{N_a},$$

where N_{tp} is the total number of packets in the network; N_g is the number of all packets generated in the network; T_{tp} is the accumulated arrival times of packets at their destinations.

The randomized networks are generated in a similar way as Watts and Strogatz[11]. Starting from the regular network as shown in Fig.1(left), we rewired each link at random with a probability p ($0 \leq p \leq 1$). We also introduced a constraint that each link cannot be connected to a further node beyond three links. This construction allows us to tune the network between the regular network ($p = 0$), partially randomized networks ($0 < p < 1$), and a fully randomized network ($p = 1$). In this simulation, we fixed an average number of packets at each node (N_p) to 50.

Results for the randomized networks are shown in Fig.2. In Fig.2(a), the proposed method transmits many packets to their destinations compared with the NN method and the Dijkstra algorithm for every p values. In addition, in Fig.2(b), the proposed method keeps the average arrival times of packets arriving at their destinations (T) shorter than the NN method and the Dijkstra algorithm.

Next, we conducted computer simulations on the scale-free networks. The scale-free networks are generated in the same way as Barabasi and Albert[12]. This network is constructed by the following procedure: First, we made a complete graph which has four nodes, then we put a new node with three links at every time step. Next, we connected three links of the newly added node to the nodes already existing in the computer network with the probability $\mathit{\Pi}(k_i) = \frac{k_i}{\sum_{j=1}^{n} k_j}$, where k_i is the degree of the i-th node ($i = 1, \ldots, n$); n is the number of nodes at a current iteration. In this simulation, the scale-free networks comprise 100 nodes.

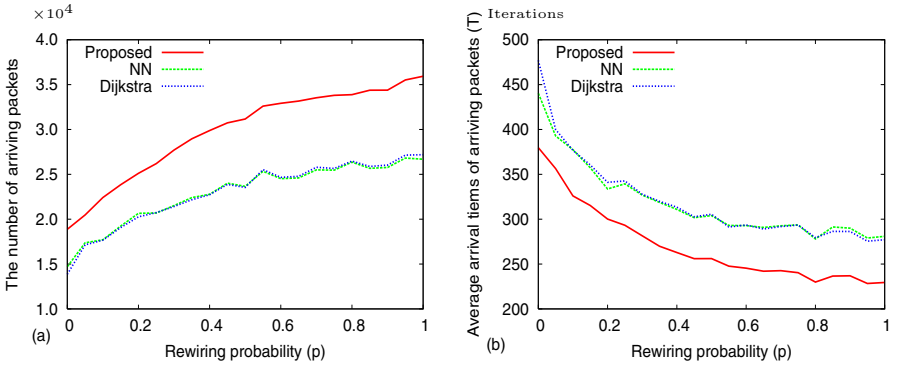


Fig. 2. Relationship between rewiring probability (p) and (a) the number of packets arriving at their destinations (N_a), and (b) an average arrival times of packets arriving at their destinations (T) for the randomized networks (Fig.1). In these figures, NN is the routing method that each neuron has only the gain effect(Eq.(1)).

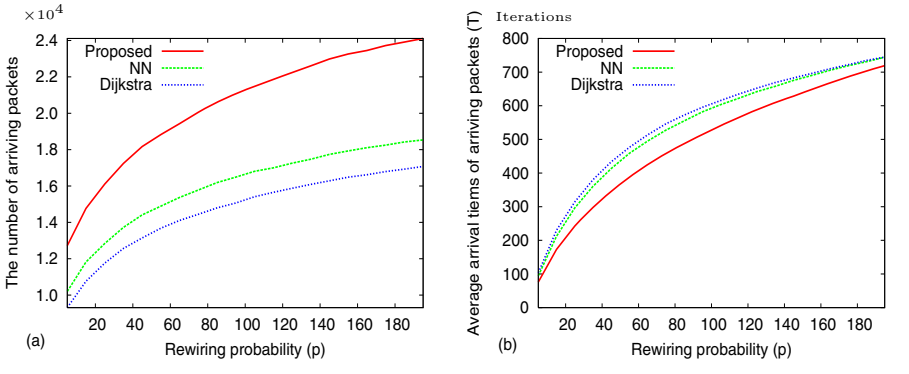


Fig. 3. Relationship between average number of packets at each node (N_p) and (a) an average number of packets arriving at their destinations (A), and (b) an average arrival times of packets arriving at their destinations (T) for the scale-free networks. NN is the routing method that each neuron has only the gain effect(Eq.(1)).

Results for the scale-free networks are shown in Fig.3. In Fig.3(a), the proposed method transmits many packets to their destinations in comparison with the NN method and the Dijkstra algorithm. In addition, in Fig.3(b), the proposed method reduces an average arrival times of packets arriving at their destinations (T) in comparison with the NN method and the Dijkstra algorithm for every N_p.

From Figs.2 and 3, we confirmed that the proposed method can select better adjacent nodes to transmit packets to their destinations faster without discarding them not only for the regular networks but also for the randomized networks and the scale-free networks.

4 Surrogate Analysis of the Effective of the Proposed Method

In previous section, we confirmed that our routing method using the chaotic neurodynamics exhibits good performance for the randomized networks and the scale-free networks. Based on the packet history of transmitting packets, the chaotic neurodynamics defined by Eq.(2) decentralizes the packets in the computer network. However, it is very important issue to consider whether the chaotic neurodynamics is really effective for the various topological networks. It might be sufficient to use only a stochastic fluctuation, which has produced the same first-order or the same second-order statistics of chaotic dynamics for avoiding the congested routes. From this point of view, we compared the performance of the proposed method with the following five routing methods:

1. Random neuron (RN),
2. Random shuffle (RS),
3. Fourier transform (FT),
4. Amplitude adjusted Fourier transform (AAFT), and
5. Fourier shuffle (FS).

These five methods produce a surrogate fluctuation of chaotic dynamics which preserves the same statistics.

In the RN method, Eq.(2) in the proposed method is replaced by $\zeta_{ij}(t+1) = -U(t)$, where $U(t)$ is a uniformly distributed random number[3, 14]. This method can also decentralize the packets in a stochastic way.

The RS method simply replaces the time-series of Eq.(2) by a time-series which has the same first order statistics as the time-series of Eq.(2)[15]; average, variance and standard deviation. The FT method uses a time-series which has the same correlation function as the time-series of Eq.(2), but destroys the first-order statistics. The AAFT method makes a time-series which preserves both first order and second order statistics of the time series of Eq.(2). The FS method also produces a time-series that preserves both first- and second-order statistics of the time-series of Eq.(2). However, the preservation of the correlation structure (the second-order statistics) using the FS method is superior to the AAFT method[15].

We analyze the proposed method and the other routing methods using the method of surrogate data for the randomized networks as shown in Fig.1 and the scale-free networks. In these simulations, we used the same experimental assumption as those in Section 3.

Results for the randomized networks are shown in Fig.4. In Fig.4(a), the proposed method transmits many packets to their destinations for every rewiring probability (p) in comparison with the other routing methods. In addition, in Fig.4(b), the proposed method reduces an average arrival times of packets arriving at their destinations (T) in comparison with the other routing methods

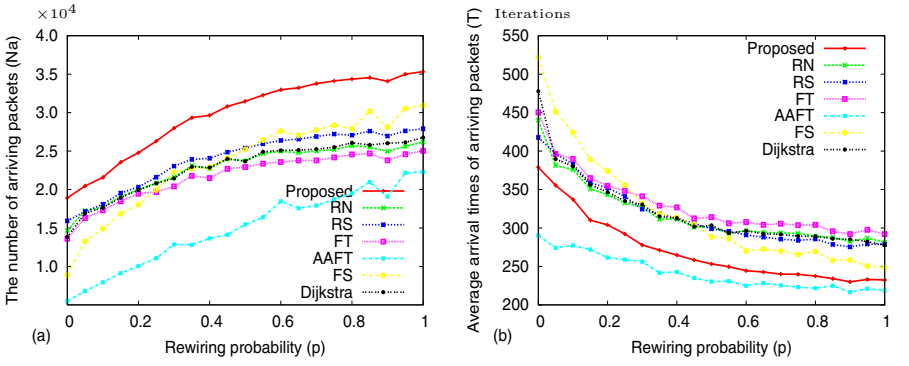


Fig. 4. Relationship between rewiring probability (p) and (a) the number of packets arriving at their destinations (N_a), and (b) an average arrival times of packets arriving at their destinations (T) for the randomized networks (Fig.1)

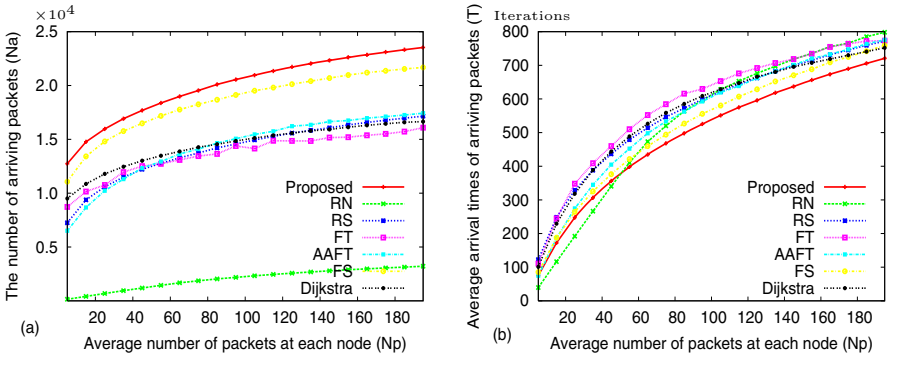


Fig. 5. Relationship between average number of packets at each node (N_p) and (a) the number of packets arriving at their destinations (N_a), and (b) an average arrival times of packets arriving at their destinations (T) for the scale-free networks

for every p values. Average arrival times of packets arriving at their destinations (T) for randomized networks in the AAFT method is the shortest. However, we cannot say that the AAFT method has good performance. This is because the number of packets arriving at their destinations (N_a) in the AAFT method is the smallest in comparison with the other routing methods.

Results for the scale-free networks are shown in Fig.5. In Fig.5(a), the proposed method transmits more packets to their destinations than the other routing methods. Furthermore, in Fig.5(b), the proposed method reduces an average arrival times of packets arriving at their destinations (T) in comparison with the other routing methods for almost N_p values. In Fig.5(b), average arrival times of packets arriving at their destinations (T) in the RN method is the shortest when the average number of packets at each node (N_p) is smaller than 45.

However, we cannot say that the RN method has good performance for the scale-free networks because the number of packets arriving at their destinations (N_a) in the RN method is smaller than the proposed method.

In Figs.4 and 5, the proposed method transmits large number of packets (N_a) to their destinations compared with the other routing methods for the randomized and the scale-free networks. These results indicate that the proposed method selects better adjacent nodes to transmit the packets to their destinations. Furthermore, selection of better adjacent nodes using a past-routing history shorten average arrival times of arriving packets (T) for the randomized networks and the scale-free networks.

5 Conclusion

In the present paper, we proposed a new algorithm for packet routing using chaotic dynamics. By introducing refractory effect, which is an essential characteristic of real nerve cells, the proposed method shows the highest performance for the randomized networks and the scale-free networks in comparison with the Dijkstra algorithm. Furthermore, we analyzed the proposed method using chaotic neurodynamics with the method of surrogate data, which is an important analysis technique in the field of nonlinear time-series analysis[15]. From the results, we also confirmed that it is an effective method to use chaotic neurodynamics in order to decentralize packets in the computer networks in comparison with the routing methods using the surrogate data making algorithms.

It has been shown that a meta-heuristic algorithm by the chaotic neural network[13] is effective for solving traveling salesman problems (TSP) and quadratic assignment problems (QAP)[4, 5]. Although we used almost the same strategy to employ chaotic neurodynamics as in Refs.[2, 3, 4, 5], the packet routing problem has a different property from TSP and QAP. TSP and QAP are usually static because the state of the problem is fixed, while the computer network always changes its state because of the flowing of the packets. Namely, the packet routing problem is dynamical combinatorial optimization. Therefore, the results shown in this paper is a good evidence that the chaotic neurodynamics could also be effective for solving the dynamical combinatorial optimization problems whose constraints are always changed, or have nonstationarity.

Many schemes of packet routing methods which are aimed at decentralizing packets in the computer network have also been proposed. In this paper, we do not compare the performance of the proposed method with such routing methods. Thus, it is an important task to compare performance of the proposed routing method with such routing methods.

We are grateful to H. Nakajima, Y. Horio, M. Adachi, M. Hasegawa, and H. Sekiya for their valuable comments and discussions. The research of TI is partially supported by Grant-in-Aid for Scientific Research (B) from JSPS (No.16300072).

Bibliography

[1] D.Bertsekas, R.Gallager: Data Networks. Prenticehall (1987)

[2] M.Hasegawa, T.Ikeguchi, K.Aihara: Exponential and chaotic neurodynamics tabu searches for quadratic assignment problems. Control and Cybernetics **29** (2000) 773–788

[3] M.Hasegawa, T.Ikeguchi, K.Aihara: Combination of chaotic neurodynamics with the 2-opt algorithm to solve traveling salesman problems. Physical Reveiw Letters **79** (1997) 2344–2347

[4] M.Hasegawa, T.Ikeguchi, K.Aihara: Solving large scale traveling salesman problems by chaotic neurodynamics. Neural Networks **15** (2002) 271–283

[5] M.Hasegawa, T.Ikeguchi, K.Aihara, K.Itoh: A novel chaotic search for quadratic assignment problems. European J. Oper. Res. **139** (2002) 543–556

[6] F.Glover: Tabu search I. ORSA Journal on Computing **1** (1989) 190–206

[7] F.Glover: Tabu search II. ORSA Journal on Computing **2** (1990) 4–32

[8] E.Mardhana, T.Ikeguchi: Neurosearch: A program library for neural network driven search meta-heuristics. Proceedings of 2003 IEEE International Symposium on Circuits and Systems **V** (2003) 697–700

[9] T.Matsuura, T.Ikeguchi, Y.Horio: Tabu search and chaotic search for extracting motifs from DNA sequences. Proceedings of the 6th Metaheuristics International Conference (2005) 677–682

[10] T.Ikeguchi: Combinatorial optimization with chaotic dynamics. Proceedings of 2005 RISP International Workshop on Nonlinear Circuits and Signal Processing (2005) 263–266

[11] D.J.Watts, S.H.Strogatz: Collective dynamics of small-world networks. Nature **393** (1998) 440–442

[12] A.-L.Barábsi, R.Albert: Emergence of scaling in random networks. Science **286** (1999) 509–512

[13] K.Aihara, T.Tanabe, M.Toyoda: Chaotic neural network. Physics Letters A **144** (1990) 333–340

[14] T.Yamada, K.Aihara: Nonlinear neurodynamics and combinatorial optimization in chaotic neural networks. Journal of Intelligent and Fuzzy Systems **5** (1997) 53–68

[15] H.Kantz, T.Schreiber: Nonlinear time series analysis. Cambridge university press (2003)

Author Index