

Ajith Abraham
Aboul-Ella Hassanien
Václav Snášel (Eds.)

Foundations of Computational Intelligence Volume 5

Function Approximation and Classification



Springer

Ajith Abraham, Aboul-Ella Hassanien, and Václav Snášel (Eds.)

Foundations of Computational Intelligence Volume 5

Studies in Computational Intelligence, Volume 205

Editor-in-Chief

Prof. Janusz Kacprzyk

Systems Research Institute

Polish Academy of Sciences

ul. Newelska 6

01-447 Warsaw

Poland

E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 181. Georgios Miaoulis and Dimitri Plemenos (Eds.)
Intelligent Scene Modelling Information Systems, 2009
ISBN 978-3-540-92901-7

Vol. 185. Anthony Brabazon and Michael O'Neill (Eds.)
Natural Computing in Computational Finance, 2009
ISBN 978-3-540-95973-1

Vol. 186. Chi-Keong Goh and Kay Chen Tan
Evolutionary Multi-objective Optimization in Uncertain Environments, 2009
ISBN 978-3-540-95975-5

Vol. 187. Mitsuo Gen, David Green, Osamu Katai, Bob McKay, Akira Namatame, Ruhul A. Sarker and Byoung-Tak Zhang (Eds.)
Intelligent and Evolutionary Systems, 2009
ISBN 978-3-540-95977-9

Vol. 188. Agustín Gutiérrez and Santiago Marco (Eds.)
Biologically Inspired Signal Processing for Chemical Sensing, 2009
ISBN 978-3-642-00175-8

Vol. 189. Sally McClean, Peter Millard, Elia El-Darzi and Chris Nugent (Eds.)
Intelligent Patient Management, 2009
ISBN 978-3-642-00178-9

Vol. 190. K.R. Venugopal, K.G. Srinivasa and L.M. Patnaik
Soft Computing for Data Mining Applications, 2009
ISBN 978-3-642-00192-5

Vol. 191. Zong Woo Geem (Ed.)
Music-Inspired Harmony Search Algorithm, 2009
ISBN 978-3-642-00184-0

Vol. 192. Agus Budiyono, Bambang Riyanto and Endra Joelianto (Eds.)
Intelligent Unmanned Systems: Theory and Applications, 2009
ISBN 978-3-642-00263-2

Vol. 193. Raymond Chiong (Ed.)
Nature-Inspired Algorithms for Optimisation, 2009
ISBN 978-3-642-00266-3

Vol. 194. Ian Dempsey, Michael O'Neill and Anthony Brabazon (Eds.)
Foundations in Grammatical Evolution for Dynamic Environments, 2009
ISBN 978-3-642-00313-4

Vol. 195. Vivek Bannore and Leszek Swierkowski
Iterative-Interpolation Super-Resolution Image Reconstruction: A Computationally Efficient Technique, 2009
ISBN 978-3-642-00384-4

Vol. 196. Valentina Emilia Balas, János Fodor and Annamária R. Várkonyi-Kóczy (Eds.)
Soft Computing Based Modeling in Intelligent Systems, 2009
ISBN 978-3-642-00447-6

Vol. 197. Mauro Birattari
Tuning Metaheuristics, 2009
ISBN 978-3-642-00482-7

Vol. 198. Efrén Mezura-Montes (Ed.)
Constraint-Handling in Evolutionary Optimization, 2009
ISBN 978-3-642-00618-0

Vol. 199. Kazumi Nakamatsu, Gloria Phillips-Wren, Lakhmi C. Jain, and Robert J. Howlett (Eds.)
New Advances in Intelligent Decision Technologies, 2009
ISBN 978-3-642-00908-2

Vol. 200. Dimitri Plemenos and Georgios Miaoulis
Visual Complexity and Intelligent Computer Graphics Techniques Enhancements, 2009
ISBN 978-3-642-01258-7

Vol. 201. Aboul-Ella Hassanien, Ajith Abraham, Athanasios V. Vasilakos, and Witold Pedrycz (Eds.)
Foundations of Computational Intelligence Volume 1, 2009
ISBN 978-3-642-01081-1

Vol. 202. Aboul-Ella Hassanien, Ajith Abraham, and Francisco Herrera (Eds.)
Foundations of Computational Intelligence Volume 2, 2009
ISBN 978-3-642-01532-8

Vol. 203. Ajith Abraham, Aboul-Ella Hassanien, Patrick Siarry, and Andries Engelbrecht (Eds.)
Foundations of Computational Intelligence Volume 3, 2009
ISBN 978-3-642-01084-2

Vol. 204. Ajith Abraham, Aboul-Ella Hassanien, and André Ponce de Leon F. de Carvalho (Eds.)
Foundations of Computational Intelligence Volume 4, 2009
ISBN 978-3-642-01087-3

Vol. 205. Ajith Abraham, Aboul-Ella Hassanien, and Václav Snášel (Eds.)
Foundations of Computational Intelligence Volume 5, 2009
ISBN 978-3-642-01535-9

Ajith Abraham, Aboul-Ella Hassanien,
and Václav Snášel (Eds.)

Foundations of Computational Intelligence Volume 5

Function Approximation and Classification



Springer

Prof. Ajith Abraham
Machine Intelligence Research Labs
(MIR Labs)
Scientific Network for Innovation and
Research Excellence
P.O. Box 2259
Auburn, Washington 98071-2259
USA
E-mail: ajith.abraham@ieee.org

Prof. Vaclav Snášel
Technical University Ostrava
Dept. Computer Science
Tr. 17. Listopadu 15
708 33 Ostrava
Czech Republic
E-mail: vaclav.snasel@vsb.cz

Prof. Aboul-Ella Hassanien
Cairo University
Faculty of Computers and Information
Information Technology Department
5 Ahmed Zewal St.
Orman, Giza
E-mail: Aboitcairo@gmail.com
<http://www.fci.cu.edu.eg/abo/>

ISBN 978-3-642-01535-9

e-ISBN 978-3-642-01536-6

DOI 10.1007/978-3-642-01536-6

Studies in Computational Intelligence

ISSN 1860949X

Library of Congress Control Number: Applied for

© 2009 Springer-Verlag Berlin Heidelberg

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typeset & Cover Design: Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed in acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Preface

Foundations of Computational Intelligence

Volume 5: Function Approximation and Classification

Approximation theory is that area of analysis which is concerned with the ability to approximate functions by simpler and more easily calculated functions. It is an area which, like many other fields of analysis, has its primary roots in the mathematics. The need for function approximation and classification arises in many branches of applied mathematics, computer science and data mining in particular.

This edited volume comprises of 14 chapters, including several overview Chapters, which provides an up-to-date and state-of-the art research covering the theory and algorithms of function approximation and classification. Besides research articles and expository papers on theory and algorithms of function approximation and classification, papers on numerical experiments and real world applications were also encouraged.

The Volume is divided into 2 parts:

Part-I: Function Approximation and Classification – Theoretical Foundations

Part-II: Function Approximation and Classification – Success Stories and Real World Applications

Part I on Function Approximation and Classification – Theoretical Foundations contains six chapters that describe several approaches Feature Selection, the use Decomposition of Correlation Integral, Some Issues on Extensions of Information and Dynamic Information System and a Probabilistic Approach to the Evaluation and Combination of Preferences

Chapter 1 “Feature Selection for Partial Least Square Based Dimension Reduction” by Li and Zeng investigate a systematic feature reduction framework by combining dimension reduction with feature selection. To evaluate the proposed framework authors used four typical data sets. Experimental results illustrate that the proposed method improves the performance on the gene expression microarray data in terms of accuracy.

In Chapter 2, “Classification by the Use of Decomposition of Correlation Integral” Jirina and Jirina Jr. illustrate that the correlation integral can be decomposed into functions each related to a particular point of data space. For these functions,

one can use similar polynomial approximations such as the correlation integral. The essential difference is that the value of the exponent, which would correspond to the correlation dimension, differs in accordance to the position of the point in question.

Chapter 3, “Investigating Neighborhood Graphs for Inducing Density Based Clusters” by Kasahara and Nicoletti investigate the impact of seven different ways of defining a neighborhood region between two points, when identifying clusters in a neighborhood graph, particularly focusing on density based clusters. On the one hand results show that the neighborhood definitions that do not employ parameters are not suitable for inducing density based clusters. On the other hand authors also illustrate that although essential for successfully separating density based clusters, the parameters employed by some of the definitions need to have their values tuned by the user.

In Chapter 4, “Some Issues on Extensions of Information and Dynamic Information Systems” Pancerz discusses some issues on extensions of information and dynamic information systems. Consistent and partially consistent extensions of information and dynamic information systems are helpful in prediction problems. On the basis of those extensions author determine possibility distributions of states and transitions between states over a universe of discourse related to a given system of processes.

Chapter 5, “A Probabilistic Approach to the Evaluation and Combination of Preferences” Parracho and Anna propose a model for the process of evaluating and combining preferences. After determining the preferences according to each criterion, these partial preferences are combined into global ones. Different combination rules are set, in a probabilistic framework. Attention is centered to the case of indicators measured in different levels of aggregation.

In Chapter 6, “Use of the q-Gaussian Function in Radial Basis Function Networks” Tinos and Murta Jr. deploy q-Gaussian function as a radial basis function in RBF Networks for pattern recognition problems. The use of q-Gaussian RBFs allows to modify the shape of the RBF by changing the real parameter q, and to employ radial units with different RBF shapes in a same RBF Network.

Part II on Function Approximation and Classification – Success Stories and Real World Applications contains six chapters that describe several success stories and real world applications on Function Approximation and Classification

Chapter 7, “Novel biomarkers for prostate cancer revealed by (α, β) - k-feature sets” by Ravetti et al. present a method based on the (α, β) - k- feature set problem for identifying relevant attributes in high-dimensional datasets for classification purposes. Using the gene expression of thousands of genes, authors illustrate that the method can give a reduced set that can identify samples as belonging to prostate cancer tumors or not.

Most classification problems associate a single class to each example or instance. However, there are many classification tasks where each instance can be associated with one or more classes. This group of problems represents an area

known as multi-label classification. In Chapter 8, “A Tutorial on Multi-Label Classification Techniques” Carvalho and Freitas present the most frequently used techniques to deal with these problems in a pedagogical manner, with examples illustrating the main techniques and proposing a taxonomy of multi-label techniques that highlights the similarities and differences between these techniques.

In Chapter 9, “Computational Intelligence in Biomedical Image Processing” Bollenbeck and Seiffert show that the segmentation of biological images, characterized by non-uniform image features, significantly benefits from combining global physical models and local feature-based supervised classification. Authors used an entropy-based voting of optimal feed-forward networks by cross-validation architecture selection and global registration-based segmentation.

Chapter 10, “A Comparative Study of Three Graph Edit Distance Algorithms” by Gao et al. propose two cost function-free GED algorithms. In the edge direction histogram (EDH)-based method, edit operations are involved in graph structure difference characterized by EDH and GED is converted into earth mover’s distance (EMD) of EDHs, while edit operations are involved in node distribution difference characterized by HMM and GED is converted into KLD of HMMs in the HMM-based method. With respect to two cost function free algorithms, HMM-based method excels EDH-based method in classification and clustering rate, and efficiency.

In Chapter 11, “Classification of Complex Molecules” by Torrens and Castellano introduce algorithms for classification and taxonomy based on criteria, e.g., information entropy and its production. In molecular classification, the feasibility of replacing a given molecule by similar ones in the composition of a complex drug is studied.

In Chapter 12 “Intelligent finite element method and application to simulation of behavior of soils under cyclic loading” Javad et al. present a neural network-based finite element method for modeling of the behavior of soils under cyclic loading. The methodology is based on the integration of a neural network in a finite element framework. In this method, a neural network is trained using experimental data representing the mechanical response of material to applied load. The trained network is then incorporated in the finite element analysis to predict the constitutive relationships for the material.

Chapter 13, “An Empirical Evaluation of the Effectiveness of Different Types of Predictor Attributes in Protein Function Prediction” Otero et al. present an empirical evaluation of different protein representations for protein function prediction in terms of maximizing predictive accuracy, investigating which type of representation is more suitable for different levels of hierarchy.

In the last Chapter, “Genetic Selection Algorithm and Cloning for Data Mining with GMDH Method” Jirina and Jirina Jr. generalize the idea of the genetically modified GMDH neural network for processing multivariate data appearing in data mining problems and to extend this type of network by cloning. Clones are close, but not identical copies of original individuals. The new genetically modified GMDH method with cloning (GMC GMDH) has no tough layered structure.

We are very much grateful to the authors of this volume and to the reviewers for their great efforts by reviewing and providing interesting feedback to authors

of the chapter. The editors would like to thank Dr. Thomas Ditzinger (Springer Engineering Inhouse Editor, Studies in Computational Intelligence Series), Professor Janusz Kacprzyk (Editor-in-Chief, Springer Studies in Computational Intelligence Series) and Ms. Heather King (Editorial Assistant, Springer Verlag, Heidelberg) for the editorial assistance and excellent cooperative collaboration to produce this important scientific work. We hope that the reader will share our joy and will find it useful!

December 2008

Ajith Abraham, Trondheim, Norway

Aboul Ella Hassanien, Cairo, Egypt

Václav Snášel, Ostrava, Czech Republic

Contents

Part I: Function Approximation and Classification: Theoretical Foundations

Feature Selection for Partial Least Square Based Dimension Reduction <i>Guo-Zheng Li, Xue-Qiang Zeng</i>	3
Classification by the Use of Decomposition of Correlation Integral <i>Marcel Jirína, Marcel Jirína Jr.</i>	39
Investigating Neighborhood Graphs for Inducing Density Based Clusters <i>Viviani Akemi Kasahara, Maria do Carmo Nicoletti</i>	57
Some Issues on Extensions of Information and Dynamic Information Systems <i>Krzysztof Pancerz</i>	79
A Probabilistic Approach to the Evaluation and Combination of Preferences <i>Annibal Parracho Sant'Anna</i>	107
Use of the q-Gaussian Function in Radial Basis Function Networks <i>Renato Tinós, Luiz Otávio Murta Júnior</i>	127

Part II: Function Approximation and Classification: Success Stories and Real World Applications

Novel Biomarkers for Prostate Cancer Revealed by (α, β) - k -Feature Sets*Martín Gómez Ravetti, Regina Berretta, Pablo Moscato* 149**A Tutorial on Multi-label Classification Techniques***André C.P.L.F. de Carvalho, Alex A. Freitas* 177**Computational Intelligence in Biomedical Image****Processing***Felix Bollenbeck, Udo Seiffert* 197**A Comparative Study of Three Graph Edit Distance Algorithms***Xinbo Gao, Bing Xiao, Dacheng Tao, Xuelong Li* 223**Classification of Complex Molecules***Francisco Torrens, Gloria Castellano* 243**Intelligent Finite Element Method and Application to Simulation of Behavior of Soils under Cyclic Loading***A.A. Javadi, T.P. Tan, A.S.I. Elkassas* 317**An Empirical Evaluation of the Effectiveness of Different Types of Predictor Attributes in Protein Function Prediction***Fernando Otero, Marc Segond, Alex A. Freitas, Colin G. Johnson, Denis Robilliard, Cyril Fonlupt* 339**Genetic Selection Algorithm and Cloning for Data Mining with GMDH Method***Marcel Jirina, Marcel Jirina Jr.* 359**Author Index** 377

Part I

**Function Approximation and
Classification: Theoretical
Foundations**

Feature Selection for Partial Least Square Based Dimension Reduction

Guo-Zheng Li¹ and Xue-Qiang Zeng²

¹ Department of Control Science and Engineering, Tongji University,
Shanghai, 201804 China

² Computer Center, Information Engineering School, Nanchang University,
Nanchang, 330006 China
drgzli, xqzeng@gmail.com

In this chapter, we will introduce our recent works on feature selection for Partial Least Square based Dimension Reduction (PLSDR). Some previous works of PLSDR, have performed well on bio-medical and chemical data sets, but there are still some problems, like how to determine the number of principle components and how to remove the irrelevant and redundant features for PLSDR. Firstly, we propose a general framework to describe how to perform feature selection for dimension reduction methods, which contains the preprocessing step of irrelevant and redundant feature selection and the postprocessing step of selection of principle components. Secondly, to give an example, we try to handle these problems in the case of PLSDR: 1) we discuss how to determine the top number of features for PLSDR; 2) we propose to remove irrelevant features for PLSDR by using an efficient algorithm of feature probes; 3) we investigate an supervised solution to remove redundant features; 4) we study on whether the top features are important to classification and how to select the most discriminant principal components. The above proposed algorithms are evaluated on several benchmark microarray data sets and show satisfied performance.

1 Introduction

As the rapid increase of the computing power during the past decades, large amount of data are accumulated. For instances, DNA microarray experiments are used to collect information from tissue, therefore cell samples have provided a lot of biology data regarding gene expression differences for tumor diagnosis [1] [2] [3]. These data sets present great challenges in data analysis. Traditional statistical methods partly break down, because of the increase in the number of variables associated with each observation.

It is of interest to reduce the dimension of the original data in many applications prior to modeling of the data. In practical, given a huge feature set of p features, we either select a small subset of interesting features (feature selection) or construct K new components summarizing the original data as well as possible, with $K < p$ (dimension reduction).

Feature selection has been studied extensively in the past few years. The most commonly used procedures of feature selection are based on a score which is calculated for all features individually and features with the best scores are selected. Feature selection procedures output a list of relevant features, whose advantages are its simplicity and interpretability. However, much information contained in the data set is lost when features are selected solely according to their individual capacity to separate the samples, since interactions and correlations between features are omitted.

Dimension reduction is an alternative to feature selection to overcome the problem of curse of dimensionality. Unlike feature selection, dimension reduction projects the whole data into a low dimensional space and constructs the new dimensions (components) by analyzing the statistical relationship hidden in the data set.

Researchers have developed different dimension reduction methods in applications of bioinformatics and computational biology [4, 5, 6], among which partial least squares based dimension reduction (PLSDR) is one of the most effective methods [6, 7]. The partial least squares (PLS) method was first developed by Herman Wold in the 1960s to address problems in econometric path-modelling [8], and was subsequently adopted by Svante Wold (and many others) in the 1980s to problems in chemometric and spectrometric modeling [9]. PLS works very well for data with very small sample size and a large number parameters. Thus, it is natural that in the last few years this method is successfully applied to problems in genomics and proteomics. A detailed chronological introduction of PLS was given in [10], some comprehensive overviews of PLS were given in [11, 12, 13] and [7]. PLS methods are in general characterized by its high computational efficiency. They also offer great flexibility and versatility in terms of the analysis problems. Only in recent years, PLSDR has been found to be an effective technique, especially compared to principal component analysis (PCA) [14, 15].

However, there are still some problems during the process of PLSDR, like how to determine the number of principal components and how to remove the irrelevant and redundant features. To solve these problems, we have done some works: 1) we discussed how to determine the top number of principal components for PLSDR in [16]; 2) we studied on whether the top features are important to classification in [17]; 3) we proposed to remove irrelevant features for PLSDR in [18, 19], 4) we investigated to remove redundant features for PLSDR in [20], and other works on application to text mining [21, 22].

In this chapter, we will summarize our works on feature selection for PLSDR and proposed a novel general framework and further give the technical details. The rest of this chapter is arranged as follows. Section 2 introduces a novel framework of feature selection for PLSDR. Section 3 further describes the detail principle of PLSDR and an orthogonal space for PLSDR [23]. Section 4 and Section 5 introduce works on how to remove irrelevant and redundant features for PLSDR. Section 6 introduces how to select the principal components obtained by PLSDR. A summary is given in Section 7. Appendix A describes the data sets we used for experiments.

Some notions are given as follows. Expression of p features in n observations are collected in an $n \times p$ data matrix $X = (\mathbf{x}_{ij})$, $1 \leq i \leq n$, $1 \leq j \leq p$; of which an entry \mathbf{x}_{ij} is the value of the j th variable of the i th sample.

Here we consider binary classification problem, the labels of the n samples are collected in vector \mathbf{y} . When the i th sample belongs to class one, the element \mathbf{y}_i is 1; otherwise it is -1.

Besides, $\|\bullet\|$ denotes the length of a vector. X^T represents the transpose of X , X^{-1} represents the inverse matrix of X . Note that X and \mathbf{y} used in this chapter are assumed to be centered to zero mean by each column.

2 A General Framework of Feature Selection for PLSDR

Dimension reduction methods like PLSDR are favorite methods in gene analysis, but there are several problems which hurt their projecting performance, 1) irrelevant and redundant feature will mislead the projection process; 2) how to determine the number of extracted principal components also affect the performance.

Feature selection and dimension reduction algorithms have complementary advantages and disadvantages. Dimension reduction algorithms thrive on correlation among features but fail to remove irrelevant features from a set of complex features. Feature selection algorithms fail when all the features are correlated but do well with informative features. It would be an interesting work to combine feature selection and dimension reduction into a generalized model.

Here, we propose to apply feature selection techniques to select relevant features for dimension reduction methods and select relevant principal components for classifiers. Fig. 1 illustrates the main steps of the novel general framework of feature selection for PLSDR, from which we see that dimension reduction consists of three parts, dimension reduction, preprocessing and postprocessing. Dimension reduction is performed by PCA and PLSDR. Preprocessing is irrelevant or redundant feature elimination, because the data set is microarray data, so it is also called gene selection. Postprocessing is really a model selection for the dimension reduction methods. We consider it feature selection since the principal components can be viewed as features. Classifier is performed by k nearest neighbor (k NN) or support vector machines (SVM). In the postprocessing step, classifier is also applied to feature selection, that is also a wrapper evaluation strategy, where classification performance of classifiers is used to evaluate the selected feature subset.

We consider preprocessing i.e. preliminary feature selection has two sides of benefits. Firstly, feature selection may improve the classification accuracy. In general, original microarray data sets have some irrelevant and noise genes, which will influence the performance of dimension reduction. In practical, biologists often expect noises are reduced, at least in some extent by the process of dimension reduction before the analysis of data. But, if some irrelevant and noise genes are reduced beforehand, we expect the performance of dimension reduction will be improved. Meanwhile, some useful information will lose due to the selection of genes. We will try to examine this influence of preliminary feature selection to PLSDR in our experiments.

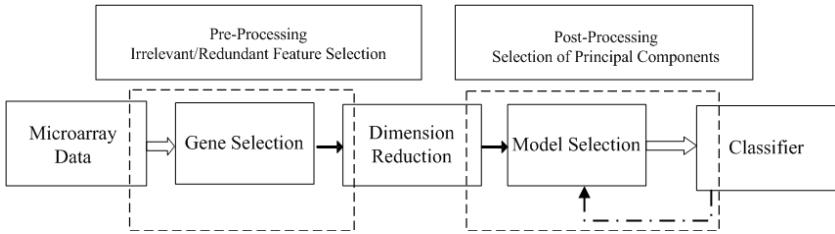


Fig. 1. A framework of feature selection for dimension reduction

Secondly, feature selection reduces the high computational complexity and the huge RAM requirement. The huge dimension of gene data matrices brings high computational time of PLSDR, and even in some extreme cases, the gene matrices are too huge to be loaded into RAM. Preliminary feature selection will definitely alleviates these two problems, since dimension reduction is performed on a gene subset not the original full set. However, we should note that any additional feature selection procedure will bring some extra computation, and Boulesteix [7] objected the preliminary feature selection for PLSDR mainly because of the huge computational complexity of cross-validation, if cross-validation is used. Nevertheless, we consider the effect of feature selection is positive if its extra computation is not too high.

In conclusion, to improve the performance of dimension reduction is the main reason to perform preliminary feature selection before PLSDR. Meanwhile, the major disadvantage of preliminary feature selection is its additional computation. In order to perform irrelevant feature elimination for PLSDR, and at the same time to overcome the shortcoming of additional computation, we propose to use an efficient feature selection method combining with probe features to remove irrelevant features.

Although dimension reduction methods produce independent features, but usually, a large number of features are extracted to represent the original data. As we known, the extracted features also contain noise or irrelevant information. Choosing an appropriate set of features is critical. Some researcher considered that the initial several components of PLSDR contain more information than the others, but it is hard to decide how many tail components are trivial for discrimination. Some authors proposed to fixed the number of components from three to five [24]; some proposed to determine the size of the space by classification performance of cross-validation [25]. However each one has its own weakness. Fixing at an arbitrary dimensional size is not applicable to all data sets, and the cross-validation method is often obstructed by its high computation. An efficient and effective model selection method for PLSDR is demanded. Furthermore, we consider not all the initial components are important for classification, subsets should be selected for classification.

3 Principle of PLSDR

PLS is a class of techniques for modeling relations between blocks of observed variables by means of latent variables. The underlying assumption of PLS is that the observed data is generated by a system or process which is driven by a small number of latent (not directly observed or measured) variables. Therefore, PLS aims at finding uncorrelated linear transformations (latent components) of the original predictor variables which have high covariance with the response variables. Based on these latent components, PLS predicts response variables \mathbf{y} and reconstruct original matrix X at the same time.

Let matrix $T = [\mathbf{t}_1, \dots, \mathbf{t}_K] \in \mathbb{R}^{n \times K}$ represents the n observations of the K components which are usually denoted as latent variables (LV) or scores. The relationship between T and X is defined as:

$$T = XV$$

where $V = [\mathbf{v}_1, \dots, \mathbf{v}_K] \in \mathbb{R}^{p \times K}$ is the matrix of projection weights. PLS determines the projection weights V by maximizing the covariance between the response and latent components.

Based on these latent components, X and \mathbf{y} are decomposed as:

$$\begin{aligned} X &= TP^T + E \\ \mathbf{y} &= TQ^T + \mathbf{f} \end{aligned}$$

where $P = [\mathbf{p}_1, \dots, \mathbf{p}_K] \in \mathbb{R}^{p \times K}$ and $Q = [\mathbf{q}_1, \dots, \mathbf{q}_K] \in \mathbb{R}^{1 \times K}$ are denoted as loadings of X and \mathbf{y} respectively. Generally, P and Q are computed by ordinary least squares (OLS). E and \mathbf{f} are residuals of X and \mathbf{y} respectively.

By the decomposition of X and \mathbf{y} , response values are decided by the latent variables not by X (at least not directly). It is believed that this model would be more reliable than OLS because the latent variables are coincided with the true underlying structure of original data.

The major point of PLS is the construction of components by projecting X on the weights V . The classical criterion of PLS is to sequentially maximizing the covariance between response \mathbf{y} and latent components. There are some variants of PLS approaches to solve this problem [26]. Ignoring the minor differences among these algorithms, we demonstrate the most frequently used PLS approach: PLS1 [11, 26].

PLS1 determines the first latent component $\mathbf{t}_1 = X\mathbf{w}_1$ by maximizing the covariance between \mathbf{y} and \mathbf{t}_1 under the constraint of $\|\mathbf{w}_1\| = 1$. The corresponding objective function is:

$$\mathbf{w}_1 = \arg \max_{\mathbf{w}^T \mathbf{w} = 1} \text{Cov}(X\mathbf{w}, \mathbf{y}) \quad (1)$$

The maximization problem of Equation (1) can be easily solved by the Lagrange multiplier method.

$$\mathbf{w}_1 = X^T \mathbf{y} / \|X^T \mathbf{y}\|$$

To extract other latent components sequentially, we need to model the residual information of X and \mathbf{y} which couldn't be explained by previous latent variables.

So, after the extraction of the score vector \mathbf{t}_1 , PLS1 deflate matrices X and \mathbf{y} by subtracting their rank-one approximations based on \mathbf{t}_1 . The X and \mathbf{y} matrices are deflated as:

$$\begin{aligned} E_1 &= X - \mathbf{t}_1 \mathbf{p}_1^T \\ \mathbf{f}_1 &= \mathbf{y} - \mathbf{t}_1 \mathbf{q}_1^T \end{aligned}$$

where \mathbf{p}_1 and \mathbf{q}_1 are loadings determined by OLS fitting:

$$\begin{aligned} \mathbf{p}_1^T &= (\mathbf{t}_1^T \mathbf{t}_1)^{-1} \mathbf{t}_1^T X \\ \mathbf{q}_1^T &= (\mathbf{t}_1^T \mathbf{t}_1)^{-1} \mathbf{t}_1^T \mathbf{y} \end{aligned}$$

As an iterative process, PLS1 constructs other latent components in turn by using the residuals E_1 and \mathbf{f}_1 as new X and \mathbf{y} .

$$\begin{aligned} \mathbf{w}_k &= E_{k-1}^T \mathbf{f}_{k-1} / \| E_{k-1}^T \mathbf{f}_{k-1} \| \\ \mathbf{t}_k &= E_{k-1} \mathbf{w}_k \\ \mathbf{p}_k^T &= (\mathbf{t}_k^T \mathbf{t}_k)^{-1} \mathbf{t}_k^T E_{k-1} \\ \mathbf{q}_k^T &= (\mathbf{t}_k^T \mathbf{t}_k)^{-1} \mathbf{t}_k^T f_{k-1} \\ E_k &= E_{k-1} - \mathbf{t}_k \mathbf{p}_k^T \\ \mathbf{f}_k &= \mathbf{f}_{k-1} - \mathbf{t}_k \mathbf{q}_k^T \end{aligned}$$

For the convenient of expression, matrices X and \mathbf{y} are often denoted as E_0 and \mathbf{y}_0 respectively. The number of components is a parameter of PLS which can be fixed by user or decided by a cross-validation scheme. In general, the maximal number of latent components is the rank of matrix X which have non-zero covariance with \mathbf{y} .

PLS reduces the complexity of original data analysis by constructing a small number of new predictors, T , which are used to replace the large number of original features. Moreover, obtained by maximizing the covariance between the components and the response variables, the PLS components are generally more predictive than the principal components extracted by other unsupervised methods like PCA [27]. After dimension reduction, classification models and many statistical analysis methods could be used based on these new predictors.

The PLSDR method use the extracted score vectors T as the new representation of original data, where $T = X \times V$. It is obvious that the deflation scheme of PLS guarantees mutual orthogonality of the extracted score vectors T , that is, $T^T T = I$. However, the projection vectors V are nonorthogonal.

As we know, the deflation scheme of PLS guarantees mutual orthogonality of the extracted score vectors T . By the arguments of [28], it can be seen that the weights $W = [\mathbf{w}_1, \dots, \mathbf{w}_K] \in \mathbb{R}^{p \times K}$ are also orthogonal. Furthermore, the relation between V and W was demonstrated [29] as:

$$V = W(P^T W)^{-1}, \quad (2)$$

from which, we evade the iterative construction of latent components T on residual matrix E_k , but relate T to X directly. In general, the loading vectors P and Q are not orthogonal. So deduced from Equation (2), we can see that the projection weights V are not orthogonal.

For the application of dimension reduction, we suppose the orthogonality of projection vectors is more important than that of score vectors. As a result, we advocate to use orthogonal projection vectors W , instead of nonorthogonal ones V . The detail works are given in [23].

3.1 Orthogonal Projecting Subspace

After dimension reduction, many statistical methods may be used for classification based on these new predictors. But the new space has one problem that the projection weights V are nonorthogonal. As the independent assumption (orthogonality) of input variables (latent components projected by V) is important for OLS regression, PLS keep the orthogonality of components T by modifying projection weights from orthogonal (W) to nonorthogonal (V). When it came to the application of dimension reduction, the orthogonality of projection directions is more desired than the orthogonality of projected components.

Additionally, it needed be clarified that the lengths of columns of V are not unit. Due to the deflation scheme of PLS, the significance of components produced iteratively are in the descending order. That is, the tail components are less informative than the initial components. Reflected by V , the lengths of these projection weights are in the descending order too. V instinctively punish the uninformative projection weights by reducing the corresponding vector lengths.

Consequently, when casting classification on the dimensions created by V , the performance of classifiers is hardly influenced by adding tail components to gene expression. This would be a problem when we are interested in these "important components", because in some situations, similar cancers can only be distinguished by certain miner genes. It is hard to say that weighted projection weights are better than united ones to help improve the generalization performance.

Though V is a natural choice of projection weights, we advocate using W to replace V . As for the vector length of W , the length of each projection weight is unit which is guaranteed by the PLS algorithm. It is noted that the latent components projected by W is not the same as original PLS latent components T . The orthogonality of latent components is not preserved as well, while we consider the modification of T is trivial, since we just use PLS as a dimension reduction tool with W and V . The modified algorithm, PLSDR is summarized in Algorithm 1

We conducted experiments on four microarray data sets, Central Nervous System, Colon, Leukemia and Prostate to investigate the difference of two series of projection weights in dimension reduction based on partial least squares from the view of orthogonality, especially to validate the proposed PLSDR algorithm [23]. Experimental results show that our proposed PLSDR performs better than normal PLSDR, and prove that W is better than V to be used in dimension reduction for classification on high dimensional data sets. We also examined the uniformity of

Algorithm 1. The PLSDR algorithm

Input: Data matrix X ; Class information vector \mathbf{y} ; The number of latent variable K
Output: Project directions $W = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_K]$

1. **Begin**
 2. $E_0 = \text{centralized } X;$
 3. $\mathbf{f}_0 = \text{centralized } \mathbf{y};$
 4. **for** $k = 1$ to K **do**
 5. $\mathbf{w}_k = E_{k-1}^T \mathbf{f}_k / \|E_{k-1}^T \mathbf{f}_k\|;$
 6. $\mathbf{t}_k = E_{k-1} \mathbf{w}_k;$
 7. $\mathbf{p}_k = E_{k-1}^T \mathbf{t}_k / \mathbf{t}_k^T \mathbf{t}_k;$
 8. $q_k = \mathbf{f}_{k-1}^T \mathbf{t}_k / \mathbf{t}_k^T \mathbf{t}_k;$
 9. $E_k = E_{k-1} - \mathbf{t}_k \mathbf{p}_k^T;$
 10. $\mathbf{f}_k = \mathbf{f}_{k-1} - \mathbf{t}_k q_k;$
 11. **end for**
 12. $W = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_K]$
 13. **End**
-

vector length of V and found that the unit of direction length is not important for the classification. In the rest of this chapter, the mentioned PLSDR algorithm means Algorithm 1.

Since some researchers use PLS to denote a classification method, in fact PLS is a dimension reduction method in this chapter, therefore we use the term of PLSDR to make a distinction from PLS based classification.

4 Irrelevant Feature Selection for PLSDR

The speciality of analysis of microarray data is the huge amount of genes with few examples, it is believed that there exist many irrelevant and redundant genes among the full gene set [30]. Preserving the most discriminative genes and reducing other irrelevant and redundant genes still remain as an open issue. As in Figure 1 preprocessing is important, we will try to eliminate irrelevant and redundant features for PLSDR. This section shows the works of irrelevant feature selection [18, 19], and the next section focuses on redundant feature selection [20].

4.1 The PLSDR^g Algorithm

PLSDR is famous for its computational efficiency [6], which can handle thousand of genes efficiently. However, researchers often neglect the problem of removing irrelevant features beforehand, which is an interesting and critical issue for its application.

There are four types of features in one data set, I is strongly relevant features, II is weakly relevant but non redundant features, III is weakly relevant and redundant features and IV is irrelevant features. I and II are the essential features in the data sets, and III and IV should be removed [31]. As in the general framework in

Algorithm 2. The PLSDR^g algorithm

Input: Training data set X ; Target variable y **Output:** The PLSDR model N

1. **Begin**
 2. Construct 100 standardized random features, get the mean value δ of their t-statistic scores with y ;
 3. Compute t-statistic scores of genes in X , eliminate those whose t-statistic scores are no greater than δ ;
 4. Train the PLSDR model N on the data subset as output.
 5. **End**
-

Section 2 we propose a simple method to eliminate irrelevant features for PLSDR based on the t-statistic scores [5,6] and the indication of random features. For binary classification, the definition of t-statistic is given as:

$$t = \frac{\bar{x}^0 - \bar{x}^1}{\text{var}^0/N^0 + \text{var}^1/N^1}$$

where N^j , \bar{x}^j and var^j are the size, mean and variance of class j , $j = 0, 1$, respectively.

A t-statistic score is computed for each gene, which stands for the discriminative ability. It is believed that a gene is important for classification when its absolute value of t-statistic is high. As the irrelevant genes have few discriminative power and tiny t-statistic scores, we add some probes of random features into the data set to identify the irrelevant genes by the scores of t-statistic. The genes, whose t-statistic scores are below the mean value of the probes's, are considered as irrelevant ones and will be eliminated from the original gene set. The novel algorithm PLSDR^g which integrates PLSDR with the above irrelevant gene elimination method is given as in Algorithm 2.

The computational complexity of PLSDR^g algorithm is $O(npK)$, which is the same as the ordinary PLSDR. The extra computation of preliminary feature selection is trivial compared with that of matrix computations. The computational complexity of the t-statistic scores of random probes is $O(p)$, and that of the gene elimination procedure is $O(n)$.

4.2 Experimental Settings

Six microarray data sets are used in our study, which are Central nervous system (CNS), Colon, DLBCL, Leukemia, Ovarian and Prostate. Detail description of these data sets refers to Appendix A.

We use the support vector machine (SVM) of both linear and nonlinear version (RBF kernel, whose kernel parameter is setup by default as the inverse of the dimension of input data) with $C = 1$, the k nearest neighbor (kNN) with $k = 1, 5, 10$ respectively, and the decision tree algorithm of C45 [32] as the classifiers, which are trained on the training set to predict the labels of test samples.

We use the stratified 10-fold cross-validation procedure for all experiments, where each data set was split into ten subsets of equal size. Each subset is used as a test set once, and the corresponding left nine subsets are combined together and used as the training set. Within each cross-validation fold, the gene expression data are standardized. The expressions of the training set are transformed to zero mean and unit standard deviation across samples, and the test set are transformed according to the means and standard deviations of the corresponding training set. The cross-validation procedure is repeated 10 times, and the mean values of sensitivity, specificity and BAC (Balanced Accuracy) [33] are recorded to measure the final performance, which are defined in Section 4.2.

We should note that the 10×10 cross-validation measurement is more reliable than the randomized re-sampling test strategy and the leave-one-out cross-validation due to the correlations between the test and training sets [34]. Even in the small-sample data sets, such as microarray data, the 10×10 cross-validation measurement still turn out to be one of the most reliable performance estimation methods [35].

The machine learning software of WEKA [36] is used in our experiments, which includes the implementations of all the learning algorithms. The parameters which haven't mentioned here are remained as default values of WEKA. Our experimental codes which programmed by JAVA language are carried out on a PC workstation with 2.66G CPU and 4GB RAM.

Measures of generalization performance

Sensitivity, specificity and BAC (Balanced accuracy) [33] are recorded to measure the final performance of classifiers, which are defined as:

$$\begin{aligned} \text{Sensitivity} &= \frac{\# \text{ correctly predicted positive examples}}{\# \text{ whole positive examples}}, \\ \text{Specificity} &= \frac{\# \text{ correctly predicted negative examples}}{\# \text{ whole negative examples}}, \\ \text{BAC} &= \frac{\text{Sensitivity} + \text{Specificity}}{2}. \end{aligned}$$

4.3 Results and Discussions

Impact of the variance of gene number

In order to examine the influence of preliminary feature selection to PLSDR, we use t-statistic gene ranking to select top ℓ genes before dimension reduction, where $\ell = 10, 20, 50, 100, 200, 500, 1,000, 1,500$, and $2,000$ respectively. The component number of PLSDR is fixed at 3 to avoid the model selection problem of PLSDR and its influence to our results. The comparative classification results of Sensitivity, Specificity and BAC on the selected gene subsets are shown in Figure 2 to Figure 3 on the DLBCL and Prostate data sets respectively. Six classifiers are used here: the

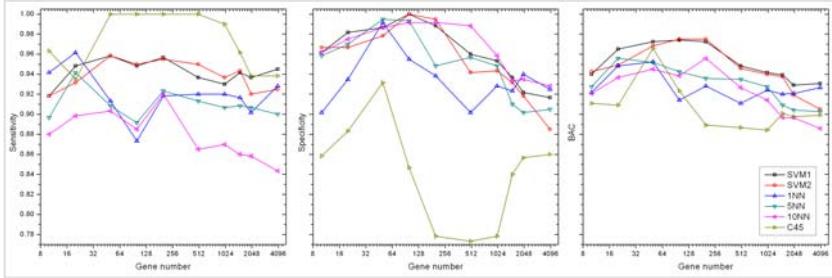


Fig. 2. Comparative results of sensitivity, specificity and BAC with the variance of gene number by using PLSDR with six different classifiers on the DLBCL data set

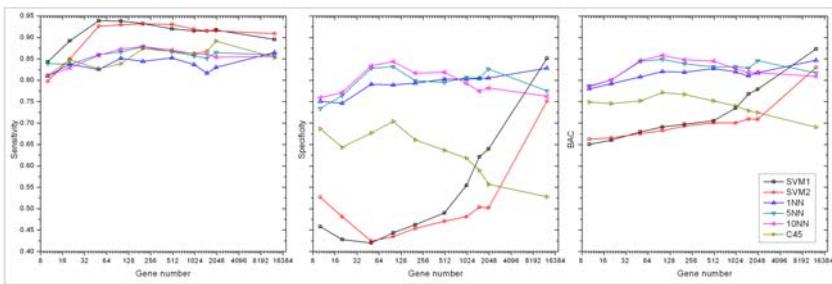


Fig. 3. Comparative results of sensitivity, specificity and BAC with the variance of gene number by using PLSDR with six different classifiers on the Prostate data set

linear support vector machine (SVM1) with $C=1$, the nonlinear SVM with RBF kernel (SVM2) whose kernel parameter is setup by default as the inverse of the dimension of input data, 1 nearest neighbor (1NN), 5 nearest neighbor (5NN), 10 nearest neighbor (10NN), and the decision tree (C45). The PLSDR model on the full gene set is also tried and the results are shown in the figures 2 to Figure 3. Besides the two figures, please refer to [18] for more results on the other four data sets of CNS, Colon, Leukemia and Ovarian.

From the figures, we can see that the effects of filter gene selection to PLSDR are not definitely positive. The impacts are heavily relying on the data sets, the used classifiers and the score measures. Some observations are given in detail as follows.

- (1) Data sets play critical roles to the evaluation of different methods. Different classifiers often show relative similar behavior to the variance of gene number on the same data set, especially for the BAC values. The most obvious example is the BAC curve on the data set of CNS. However, we could not find any method which is insensitive to the variance of data sets. For instance, feature selection before dimension reduction improves the BAC scores of SVM1 on the data set of DLBCL, but dramatically decreases its classification performance on the data set of Prostate.

- (2) Classification models have great influence to the final performances. It's well known that different classifiers have different performance on any given data set. Here, the figures also confirm that SVM is relatively the best classifier and C45 often behaves the worst, which are coincided with common assumptions.
- (3) Evaluation metrics of classification performance are also very important to the evaluation. In figures, the Sensitivity and Specificity scores are often very different when the same classifiers are applied on the same data sets. In some cases, the Sensitivity and Specificity values have changed a lot when the BAC value remains stable. For example, at the beginning stage of the addition of gene on the data set of CNS, the Sensitivity and Specificity scores of SVM2 dramatically change in different direction, one is improved and another is dropped. At the same time, the variance of BAC curve is not obvious.
- (4) In general, the selection of genes can improve the classification performance. If we concentrate on the BAC curves, we can find the top BAC scores are usually not obtained at the full gene set (except for the SVM on the data set of Prostate). However, the optimal gene number variance a lot on different data sets and classifiers. It is difficult to determine the optimal gene number in real applications.

The above results indicate that, without validation, the attempt to dramatically reduce the size of feature set has the danger to hurt the performance of dimension reduction and the final generalization classification. So, we consider that only reducing irrelevant genes from the original set is a wise alternative. It is also not necessary to select a tight gene subset due to the stage of dimension reduction which will project the data into a much smaller subspace.

Efficiency of PLSDR^g

We apply both PLSDR and our proposed algorithm PLSDR^g on the six microarray data sets to examine the difference of efficiency performance. Compared with results of normal PLSDR, the used gene number and CPU running time (millisecond, ms) of PLSDR^g are shown in Table II. The values with \pm std are the statistical mean values with their standard deviations (std), where the mean values and stds are calculated on the ten folds of each cross-validation procedure. Furthermore, the final results are averaged by ten iterations of cross-validation procedure.

The results in Table II show that:

- (1) The proposed PLSDR^g method obviously reduces the dimensionality. In average, about 29 percent genes are reduced from the full set. This fact indicates there exist many irrelevant genes in the microarray data sets, whose t-statistic values with target variable are no great than that of random variables.
- (2) The computational efficiency is slightly improved by PLSDR^g. In average, the CPU running time is reduced from 205.407ms to 204.998ms. PLSDR^g is faster than normal PLSDR on almost all the data sets, except for the data set of Ovarian. A possible reason is that the small reduction of genes from a huge gene set of Ovarian makes the efficiency improvement of PLS modelling not enough to compensate the time loss of the feature selection procedure. But, in most cases, the time saving by feature selection is obviously.

Table 1. Comparative results of the used gene number and CPU time by using PLSDR and PLSDR^g on six data sets

Data Set	Number of Genes		CPU Time (ms)	
	PLSDR	PLSDR ^g	PLSDR	PLSDR ^g
CNS	7,129	3,055.81	62.910±0.604	40.350±2.959
Colon	2,000	1,272.70	15.960±0.587	13.940±0.893
DLBCL	4,026	2,326.97	24.670±0.529	22.160±0.698
Leukemia	7,129	4,988.16	77.400±0.822	69.980±1.931
Ovarian	15,154	11,641.14	705.790±8.390	772.490±33.736
Prostate	12,600	9,923.80	345.710±3.921	311.070±17.573
Average	10,632	7,544.16	205.407±2.475	204.998±9.632

Table 2. Comparative results of sensitivity, specificity and BAC by using SVM1 for PLSDR and PLSDR^g on six data sets

Data Set	Sensitivity		Specificity		BAC	
	PLSDR	PLSDR ^g	PLSDR	PLSDR ^g	PLSDR	PLSDR ^g
CNS	0.280±0.282	0.352±0.298	0.809±0.213	0.794±0.187	0.545±0.158	0.573±0.165
Colon	0.893±0.153	0.890±0.154	0.788±0.272	0.807±0.259	0.840±0.154	0.848±0.154
DLBCL	0.945±0.123	0.928±0.157	0.917±0.174	0.930±0.150	0.931±0.101	0.929±0.106
Leukemia	0.974±0.059	0.976±0.058	0.893±0.199	0.907±0.191	0.934±0.099	0.941±0.096
Ovarian	0.987±0.037	0.974±0.061	0.989±0.025	0.989±0.025	0.988±0.021	0.981±0.033
Prostate	0.896±0.106	0.912±0.096	0.851±0.138	0.843±0.142	0.873±0.087	0.878±0.081
Average	0.829±0.127	0.839±0.137	0.875±0.170	0.878±0.159	0.852±0.104	0.858±0.106

Sum up, the PLSDR^g method is very efficient, which greatly reduces the gene dimension without the loss of computational efficiency. Then, we are interested in the problem of whether the classification performance is improved by PLSDR^g or not, which is examined in the following.

Classification performance of PLSDR^g

After dimension reduction, six classifiers are applied on PLSDR^g and normal PLSDR including SVM1, SVM2, 1NN, 5NN, 10NN and C45 in our experiments. The detailed comparative results of Sensitivity, Specificity and BAC are shown in Table 2 to Table 3 by using SVM1 and SVM2 respectively. Please refer to [18] for more results by using kNNs and C45. The last rows in these tables are the average values across six different data sets. The values with ±std are the statistical mean values with their standard deviations (std), where the mean values and stds are calculated on the ten folds of each cross-validation procedure. Furthermore, the final results are averaged by ten iterations of cross-validation procedure.

Table 3. Comparative results of sensitivity, specificity and BAC by using SVM2 for PLSDR and PLSDR^g on six data sets

Data Set	Sensitivity		Specificity		BAC	
	PLSDR	PLSDR ^g	PLSDR	PLSDR ^g	PLSDR	PLSDR ^g
CNS	0.275±0.297	0.348±0.319	0.916±0.149	0.857±0.189	0.595±0.164	0.603±0.172
Colon	0.910±0.138	0.905±0.139	0.640±0.358	0.698±0.343	0.775±0.197	0.802±0.190
DLBCL	0.925±0.167	0.917±0.168	0.885±0.203	0.915±0.181	0.905±0.136	0.916±0.126
Leukemia	1.000±0.000	1.000±0.000	0.713±0.299	0.737±0.301	0.857±0.150	0.868±0.150
Ovarian	0.951±0.076	0.958±0.069	0.981±0.035	0.982±0.032	0.966±0.040	0.970±0.039
Prostate	0.909±0.103	0.912±0.101	0.752±0.167	0.745±0.165	0.830±0.098	0.828±0.093
Average	0.828±0.130	0.840±0.133	0.778±0.254	0.778±0.266	0.821±0.131	0.831±0.129

The results in Table 2 and Table 3 show that:

- (1) Classification performance of PLSDR^g is better than that of normal PLSDR. There is no method always overwhelms another on any data sets and classifiers, PLSDR^g has shown better performance in general. For instance, the averaged Sensitivity and BAC scores of PLSDR^g are consistently better than that of normal PLSDR by using all classifiers.
- (2) The selected gene subset is more helpful for the improvement of Sensitivity than that of Specificity. As we mentioned above, Sensitivity and Specificity often behave differently to the addition of genes. The effect of PLSDR^g to them is also different. One reason is that all the data sets are somewhat imbalanced, improvement of sensitivity means PLSDR^g is better at handling the imbalanced problem than PLSDR.

4.4 Conclusion

PLSDR is a widely used method in bioinformatics and related fields. Whether a preliminary feature selection should be applied before PLSDR is an interesting issue, which was often neglected in the previous works. In this section, we examined the influence of preliminary feature selection by the t-statistic gene ranking method for PLSDR. We found the effects of feature selection are helpful, but the optimal gene dimension is hard to be determined. Without validation, greatly reduction of the number of genes before dimension reduction has the risk to hurt the final classification performance.

Therefore, eliminating a moderate part of genes from full set before dimension reduction may be a good choice, since dimension reduction finally projects the data into a small subspace. Based on the notion that irrelevant genes are always not useful for modelling, our proposed PLSDR^g seems to be an efficient and effective gene elimination method by the indication of t-statistic scores of random features. The empirical results on six microarray data sets confirmed the efficiency and effect of our new method and proved that PLSDR^g improves prediction accuracy of learning machines for PLSDR.

5 Redundant Feature Selection for PLSDR

In this section, we propose a novel metric of redundancy which effectively eliminates redundant genes before feature extraction. By measuring the discriminative ability of each gene and the pair-wise complementarity, the new method reduce the redundant genes with little contribution of discriminative ability. We also compare our method with commonly used redundant gene reduction methods based on linear correlation. Experiments on several microarray data sets demonstrate the outstanding performance of our method. The work is given in [20] in detail.

5.1 Computational Methods

As we show in Section 2 redundant gene elimination is the critical part in the framework, we propose a novel algorithm based on discriminative ability to improve performance of commonly used linear correlation.

Discriminative ability (predictive ability) is a general notion which can be measured in various ways and be used to select significant features for classification. Many effective metrics had been proposed such as t-statistic, information gain, χ^2 statistic, odds ratio etc. [37, 38]. Filter feature selection methods sort features by the discriminative ability scores, and some top rank features are retained to be essential for classification.

However, t-statistic and most of other discriminative ability measures are based on individual features, which do not consider the redundancy between two features. Because given two features with the same rank scores, they may be redundant to each other when they are completely correlated, otherwise, they may also be complementary to each other when they are nearly independent.

For the task of feature selection, we want to eliminate the redundant features and only retain the interactive ones. But there exist many redundant features in the top rank feature set produced by using the filter methods. The redundant features increase the dimensionality and contribute little for the final classification. In order to eliminate redundant features, metrics need to estimate the redundancy directly.

On the other side, notions of feature redundancy are normally in terms of feature correlation. It is widely accepted that two features are redundant to each other if their values are completely correlated. But in fact, it may not be so straightforward to determine feature redundancy when a feature is correlated with a set of features. The widely used way is to approximate the redundancy of feature set by considering the pair-wise feature redundancy.

For linear cases, the most well known pair-wise redundancy metric is the linear correlation coefficient. Given a pair of features (x, y) , the definition of the linear correlation coefficient $\text{Cor}(x, y)$ is:

$$\text{Cor}(x, y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}}$$

where \bar{x} and \bar{y} are the mean of x and y respectively. The value of $\text{Cor}(x, y)$ lies between -1 and 1. If x and y are completely correlated, $\text{Cor}(x, y)$ takes the value of 1 or -1; if x and y are independent, $\text{Cor}(x, y)$ is zero. It is a symmetrical metric.

The linear correlation coefficient has the advantage of its efficiency and simplicity, but it is not suitable for redundant feature elimination when classification is the final target, since it does not use any label information. For example, two highly correlated features, whose differences are minor in values but happen to causing different critical discriminative ability, may be considered as a pair of redundancy features. Reducing any one of them will decrease classification accuracy. Guyon et al. has also pointed out that high correlation (or anti-correlation) of variables does not mean absence of variable complementarity [37]. The problem of the linear correlation coefficient is that it measures the similarity of the numerical values between two features, but not the similarity of discriminative ability between two features.

The ideal feature set should have both great discriminative ability and little feature redundancy, where redundancy could not be obtained by estimating their properties separately. A more elaborate measure of redundancy is required to estimate the differences of the discriminative ability between two features.

The DISC metric

In order to measure the similarity of discriminative ability of two features, the discriminative ability need be defined more precisely. That is to say, we want to know which example can be rightly classified by the given feature and which can not. Upon the new metric, it is possible to compare the discriminative ability of two features by the corresponding correctly classified examples.

In the field of text classification, Training Accuracy on Single Feature (TASF) has been proved to be an effective metric of discriminative ability [38], which builds a classifier for each feature, and the corresponding training accuracy is used as the discriminative score.

Various classifiers can be used to calculate TASF, in simplification, we consider a linear learner here. Given a feature z , the classification function is given as:

$$\hat{y} = \text{sgn}\left((\bar{z}^1 - \bar{z}^2)\left(z - \frac{n^1\bar{z}^1 + n^2\bar{z}^2}{n^1 + n^2}\right)\right)$$

where \bar{z}^1 and n^1 are the feature mean and the sample size of class one, \bar{z}^2 and n^2 are the feature mean and the sample size of class two. This is a weighted centroid based classifier, which predicts examples as the class label whose weighted distance to its centroid is smaller. The computational complexity of this classifier is $O(n)$.

Putting the whole training set back, we can estimate training accuracy of each classifier by different features, which is used to represent discriminative ability of the corresponding feature. The higher training accuracy, the greater discriminative ability. Since only one feature is used to build the classifier, a part of training examples can be correctly separated in most cases. So the value of TASF ranges from 0 to 1. One feature is considered as an irrelevant one if its TASF value is no greater than 0.5.

Based on TASF, we propose a novel metric of feature redundancy. Given two features of z_1 and z_2 , two classifiers C_1 and C_2 can be constructed. Feeding the

whole training set to the classifiers, both C_1 and C_2 can correctly classify a sample subset. The differences of the correctly classified examples are used to estimate the similarity of discriminative abilities. We record the concrete classification situation as follows:

		C_2	true	false
		C_1		
true	true	a	b	
	false	c	d	

Here $a + b + c + d$ equals to the size of the training set n . The values of $(a + b)/n$ and $(a + c)/n$ are training accuracy of C_1 and C_2 respectively. The score of $a + d$ measures the similarity of the features, and the score of $b + c$ measures the dissimilarity. When $b + c = 0$, the two features z_1 and z_2 have exactly the same discriminative ability.

Our feature elimination problem is becoming whether the contribution of the additional feature to the given feature is significant. The additional feature is considered as redundant if its contribution is tiny. Then, we propose a novel metric of Redundancy based on DIScriminative Contribution (DISC). DISC of z_1 and z_2 , which estimates z_2 's redundancy to z_1 , is defined as follows,

$$\begin{aligned} \text{DISC}(z_1, z_2) &= 1 - \frac{c}{c + d} \\ &= \frac{d}{c + d} \end{aligned} \quad (3)$$

The pair-wise DISC metric is asymmetrical, and the computation complexity is $O(n)$.

It is clear that $c + d$ is the number of examples which could not be discriminated by C_1 , c is that which could be correctly classified by the collaboration of C_1 and C_2 . So the proportion of $c/(c + d)$ is the discriminative contribution of C_2 to C_1 , and the value of $d/(c + d)$ is the DISC metric of redundancy, which varies from 0 to 1. When the DISC score takes 1, C_2 's discriminative ability is covered by C_1 's and then z_2 is completely redundant to z_1 . When the DISC value is 0, all training examples could be correctly classified by the union of C_2 and C_1 and we consider z_2 is complementary to z_1 .

DISC is proposed in a linear way, which shows in two respects, one is the linear classifier, another is the linear way of counting the cross discriminative abilities. The microarray problems meet the assumption, since most microarray data sets are binary classification problems, where each gene has equal position to perform classification.

The REDISC algorithm

Based on the DISC redundancy metric, we propose the REDISC algorithm (Redundancy Elimination based on Discriminative Contribution), which eliminates redundant features by the pair-wise DISC scores. The basic idea of REDISC is that, firstly, REDISC filters out trivial features, which do not have discriminative ability on itself, by the TASF score threshold of 0.5. Then the features are ordered

Algorithm 3. The REDISC algorithm

Input: Feature set $X = [x_1, x_2, \dots, x_p]$; Target variable y ; Threshold δ
Output: Selected feature subset S

```

1. Begin
2. for  $i = 1$  to  $p$  do
3.   Calculate  $TAS F_i$  for  $X_i$ 
4.   if  $TAS F_i \geq 0.5$  then
5.     Append  $X_i$  to  $S'$ 
6.   end if
7. end for
8. Order  $S'$  in descending  $TAS F_i$  value
9. for  $j = 1$  to the size of  $S'$  do
10.   for  $i =$  the size of  $S'$  to  $j + 1$  do
11.     if  $DISC(S'_{j'}, S'_i) < \delta$  then
12.       Remove feature  $i$  from  $S'$ 
13.     end if
14.   end for
15. end for
16.  $S \Leftarrow S'$ 
17. End

```

by their TASF scores. As we usually want to retain the more discriminative one between two redundant features, REDISC tries to preserve the top TASF score ranked features. REDISC uses two nested iterations to eliminate redundant features whose discriminative ability are covered by any higher ranked features. The computational complexity of REDISC is $O(np^2)$. The algorithm is given as Algorithm 3.

In order to compare our method with commonly used redundant feature elimination methods, we present the algorithm of RELIC (Redundancy Elimination based on Linear Correlation) [39], which filters out redundant features by the pair-wise linear correlation. A threshold is needed to control how many features should be eliminated. The computational complexity of RELIC is also $O(np^2)$. The algorithm is given as Algorithm 4.

5.2 Experimental Settings

Two microarray data sets, Colon and Leukemia, are used in this study, whose details refer to Appendix A.

The linear Support Vector Machine (SVM) with $C = 1$ is used as the classifier, which is trained on the training set to predict the label of test samples.

We use the stratified 10-fold cross-validation procedure, where each data set is firstly merged and then split into ten subsets of equal size. Each subset is used as a test set once, and the corresponding left subsets are combined together and used as the training set. Within each cross-validation fold, the gene expression data is standardized. The expressions of the training set are transformed to zero mean and unit standard deviation across samples, and the test set are transformed according to

Algorithm 4. The RELIC algorithm

Input: Feature set $X = [x_1, x_2, \dots, x_p]$; Target variable y ; Threshold δ
Output: Selected feature subset S

1. **Begin**
2. Add each feature in X to S'
3. Order S' in descending $Cor(S'_i, y)$ value
4. **for** $j = 1$ to the size of S' **do**
5. **for** $i =$ the size of S' to $j + 1$ **do**
6. **if** $Cor(S'_j, S'_i) < \delta$ **then**
7. Remove feature i from S'
8. **end if**
9. **end for**
10. **end for**
11. $S \leftarrow S'$
12. **End**

the means and standard deviations of the corresponding training set. We use 10 fold cross validation because the 10×10 cross-validation measurement is more reliable than the randomized re-sampling test strategy and the leave-one-out cross-validation due to the correlations between the test and training sets, some detail discussions can be found at [40].

The cross-validation procedure is repeated 10 times, and the mean values of sensitivity, specificity and BAC [33] are recorded to measure the final performance, which are defined in Section 4.2.

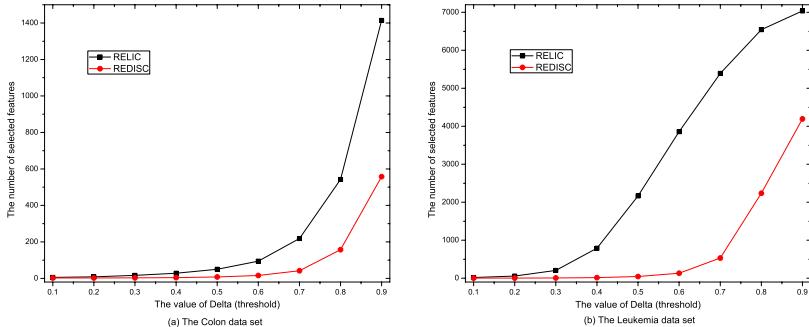


Fig. 4. The number of selected genes by performing REDISC and RELIC with different parameters

5.3 Results and Discussions

According to the framework proposed in this chapter, dimension reduction is performed by combining redundant gene elimination with dimension reduction, then the classifier is used to perform classification on the extracted feature subsets. The

novel proposed algorithm REDISC is compared with the commonly used algorithm RELIC to perform redundant gene elimination on two microarray data sets, i.e. Colon and Leukemia, where the threshold of δ in REDISC and RELIC is varied from 0.1 to 0.9. Dimension reduction is performed by partial least squares (PLS). The classifier is a linear support vector machine (SVM) with $C = 1$.

Statistical results of the number of remained genes after performing REDISC and RELIC are showed in Fig. 4. Detailed results of Sensitivity, Specificity and BACC on Colon and Leukemia are showed in Table 4 and Table 5, where the results are averaged on ten times of run. More details of REDISC and RELIC on PCA can be found in [20].

The results in Fig. 4, Table 4 and Table 5 show that:

- (1) Both REDISC and RELIC dramatically reduce the number of genes from the original data. With the same value of δ , REDISC obtains more compact subsets than RELIC does.
- (2) With $\delta = 0.1$, RELIC always obtains better results than REDISC, but when δ increases, the results of REDISC are better than those of RELIC. On average, REDISC obtains better results than RELIC does.
- (3) When the results of REDISC and RELIC reach their highest point, REDISC uses less features than RELIC.
- (4) The effect of REDISC is positive for both PCA and PLSDR, while RELIC loses in some case, i.e. BACC of PCA on the Leukemia data set.
- (5) REDISC and RELIC with different threshhold values produces different results, no one is optimal for all the data sets.

The experimental results prove our assumption that redundant features hurt performance of dimension reduction and classification, other considerations on the above results are listed as below:

- (1) The results confirm that there exist many redundant genes in the microarray data and it is necessary to perform redundant gene elimination. As we know in the previous section, features are categorized into four types, the weakly relevant but redundant features should be removed before classification. In this chapter, we show they should also be removed for dimension reduction like PLSDR.
- (2) REDISC obtains better results with less features than RELIC, which shows that REDISC has the higher ability to select relevant features and eliminate the redundant features than RELIC. Proper redundant feature elimination help improve performance of dimension reduction and classification. Simply reducing redundant genes by linear correlation is not always positive, because without considering the label information in the data set, linear correlation does not give properly redundancy estimation. REDISC takes label information into account for redundant gene elimination, which may be viewed as a supervised way. Since the final step is classification, so a supervised redundant gene elimination is better than an unsupervised one like RELIC.
- (3) It shows the performances of dimension reduction is improved when redundant genes are properly eliminated. The improvement for PLSDR is much more

Table 4. Statistical results by performing PLSDR after REDISC and RELIC with different parameters on the Colon data set

δ	RELIC				REDISC			
	#genes	Sensitivity	Specificity	BACC	#genes	Sensitivity	Specificity	BACC
0.1	5.62	0.9750	0.3167	0.6458	2	1.0000	0.1850	0.5925
0.2	8.56	0.9600	0.4167	0.6883	2.1	0.9850	0.2083	0.5967
0.3	16.55	0.9350	0.5567	0.7458	2.94	0.9800	0.2983	0.6392
0.4	28.08	0.9150	0.6650	0.7900	4.3	0.9750	0.3750	0.6750
0.5	49.55	0.9100	0.7017	0.8058	8.09	0.9200	0.5133	0.7167
0.6	94.3	0.8975	0.7133	0.8054	15.82	0.8975	0.6533	0.7754
0.7	218.37	0.8950	0.7650	0.8300	41.9	0.9000	0.7800	0.8400
0.8	542.46	0.8825	0.7917	0.8371	157.06	0.8950	0.8150	0.8550
0.9	1413.92	0.8750	0.7967	0.8358	558	0.8900	0.7900	0.8400
Full Set	2000	0.8750	0.7733	0.8242				

Table 5. Statistical results by performing PLSDR after REDISC and RELIC with different parameters on the Leukemia data set

δ	RELIC				REDISC			
	#genes	Sensitivity	Specificity	BACC	#genes	Sensitivity	Specificity	BACC
0.1	18.1	0.9400	0.7567	0.8483	3.31	0.9865	0.6317	0.8091
0.2	55.86	0.9110	0.7850	0.8480	4.29	0.9805	0.6683	0.8244
0.3	205.48	0.9415	0.8150	0.8783	6.88	0.9695	0.7833	0.8764
0.4	790.41	0.9605	0.8133	0.8869	14.83	0.9635	0.8783	0.9209
0.5	2168.33	0.9720	0.9017	0.9368	46.56	0.9710	0.9533	0.9622
0.6	3859.52	0.9795	0.9550	0.9672	131.23	0.9665	0.9633	0.9649
0.7	5394.2	0.9795	0.9500	0.9647	531.33	0.9775	0.9450	0.9612
0.8	6545.28	0.9815	0.9150	0.9483	2239.81	0.9855	0.9383	0.9619
0.9	7035.99	0.9840	0.9067	0.9453	4195.03	0.9885	0.9117	0.9501
Full Set	7129	0.9840	0.9083	0.9462				

dramatic than that of PCA. A possible reason is redundant genes obstruct the performance of supervised methods more obviously, since supervised methods often build more precisely model than unsupervised ones.

5.4 Conclusion

Dimension Reduction is widely used in bioinformatics and related fields to overcome the curse of dimensionality. But the existence of amounts of redundant genes in the microarray data often obscure the application of dimension reduction. Preliminarily redundant gene elimination before dimension reduction for dimension reduction is an interesting issue, which was often neglected.

In this section, a novel metric, DISC, is proposed, which directly estimates the similarity between two features by explicitly building linear classifiers on each

genes. The REDISC algorithm is also proposed. REDISC is compared with a commonly used algorithm RELIC on two real microarray data sets. Experimental results demonstrate the necessity of preliminarily redundant gene elimination before feature extraction for tumor classification and the superiority of REDISC to RELIC, a commonly used method.

6 Selection of Principal Components

As we show in the general framework of in Figure 1, postprocessing is also an important part. Here, we propose and demonstrate the importance of feature selection after dimension reduction in the tumor classification problems. We have performed experiments by using PCA [41] and PLSDR [17] as dimension reduction methods separately. Here, we perform a systematic study on both PCA and PLSDR methods, which will be combined with the feature selection methods (Genetic Algorithm) to get more robust and efficient dimensional space, and then the constructed data from the original data is used with k Nearest Neighbor (k NN) for classification. By applying the systematic study on the analysis of gene microarray data, we try to prove that feature selection selects proper components for PCA and PLSDR dimension reduction and not only the top components are nontrivial for classification [42].

6.1 The GA-FS Algorithm

In the previous works, the number is fixed as 3 or 5 top ones, or obtained by cross validation. These works assume that only the top several components are important. In fact the components are ranked from a statistical view; it may not be the same rank according to their discriminative ability. Therefore, we propose to apply feature selection techniques to select components for classifiers. From Fig. 1 we can see that dimension reduction consists of two parts, dimension reduction and feature selection, here dimension reduction is performed by PCA and PLSDR, feature selection is performed by GA and classifier is performed by k nearest neighbor (k NN). In Fig. 1 classifier is also applied to feature selection, that is also called the wrapper evaluation strategy, classification performance of classifiers is used to evaluate the selected feature subset.

Finding out the optimal feature subset according to classification performance is referred to as feature selection. Given a set of features, the problem is selecting a subset that leads to the least classification error. A number of feature selection methods have been studied in the bioinformatics and machine learning fields [43, 44, 45]. There are two main components in every feature subset selection system: the search strategy used to pick the feature subsets and the evaluation method used to test their goodness based on some criteria. Genetic algorithm as a search strategy is proved to be the best one among different complete and heuristic methods [46]. There are two categories of evaluation strategies: 1) filter and 2) wrapper. The distinction is made depending on whether feature subset evaluation is performed using the learning algorithm employed in the classifier design (i.e., wrapper) or not (i.e., filter). Filter approaches are computationally more efficient than wrapper approaches since they

evaluate the goodness of selected features using criteria that can be tested quickly. This, however, could lead to non-optimal features, especially, when the features dependent on the classifier. As result, classifier performance might be poor. Wrapper methods on the other hand perform evaluation by training the classification error using a validation set. Although this is a slower procedure, the features selected are usually more optimal for the classifier employed. Here we want to improve classification performance, and use the wrapper strategy. Classification performance of k NN is used as the criteria in this chapter.

Genetic Algorithm (GA) is a class of optimization procedures inspired by the biological mechanisms of reproduction. [47]. GA operate iteratively on a population of structures, each one of which represents a candidate solution to the problem at hand, properly encoded as a string of symbols (e.g., binary). Three basic genetic operators guide this search: selection, crossover, and mutation. The genetic search processes it iterative: evaluating, selecting, and recombining strings in the population during each iteration until reaching some termination condition.

The basic idea is that selection probabilistically filters out solutions that perform poorly, choosing high performance solutions to concentrate on or exploit. Crossover and mutation, through string operations, generate new solutions for exploration. Given an initial population of elements, GA uses the feedback from the evaluation process to select fitter solutions, generating new solutions through recombination of parts of selected solutions, eventually converging to a population of high performance solutions.

In our proposed algorithm GA-FS (Genetic Algorithm based Feature Selection), we use a binary chromosome with the same length as the feature vector, which equals 1 if the corresponding feature is selected as the input, and 0 if the feature is discarded. The goal of using GA here is to use fewer features to achieve the same or better performance. Therefore, the fitness evaluation contains two terms: 1) Classification error; 2) The number of selected features. We use the fitness function shown below:

$$\text{fitness} = \text{error} + \gamma * \text{number_of_selected_features}, \quad (4)$$

where error corresponds to the classification error on the validation data set X_v , γ is a trade-off between classification error and the number of selected features. Here between classification error and feature subset size, reducing classification error is our major concern, so γ is set to $1/(2 * 10^4)$.

The GA-FS approach is summarized in Algorithm 5 where the data set is divided into 3 parts, training set X_r , validation set X_v and test set X_s as in the subsection of experimental setting.

k nearest neighbor is a non-parametric classifier [32], where the result of new instance is classified based on majority of k nearest neighbor category, any ties can be broken at random.

6.2 Experimental Settings

Four microarray data sets, CNS, Colon, Leukemia and Lung cancer are used in our study which are briefly described as in Appendix A

Algorithm 5. The GA-FS algorithm

Input: training set X_r , validation set X_v , test set X_s , and the base learner

Output: prediction error on the test set X_s

1. **Begin**
 2. Generate a population of weight vectors
 3. Evolve the population where the fitness of a weight vector \mathbf{w} is measured as in Eq. (4) on X_r and X_v
 4. \mathbf{w}^* = the evolved best weight vector
 5. Test on X_s with features corresponding to 1's in \mathbf{w}^* as the input and those to 0's be removed
 6. **End**
-

To evaluate the performance of the proposed approach, we use the hold out validation procedure. Each data set is used as a whole set, originally split data sets are merged, and then we randomly divide the whole set into the training set and test set X_s (2/3 for training and the rest for test). Furthermore, if a validation data set is needed, we splits the training data set, keeping 2/3 samples for training X_r and the rest for validation X_v . Classification error of k NN is obtained on the test data sets X_s . We repeat the process 50 times.

The parameters of GA is set by default as in the software of MATLAB, and we set different parameters for k NN to test how parameters affect the results.

6.3 Results and Discussions

In order to show the importance of feature selection, we have also performed the following series experiments on the k NN learning machine to reduce the bias caused by learning machines.

- (a) KNN is a baseline method, all the genes without any selection and extraction are input into k NN for classification.
- (b) PCAKNN uses PCA as dimension reduction methods, all the newly extracted components are input into k NN.
- (c) PLSKNN uses PLSDR as dimension reduction methods, all the newly extracted components are input into k NN.
- (d) PPKNN uses PCA+PLSDR as dimension reduction methods, all the newly extracted components are input into k NN.
- (e) GAPCAKNN uses PCA as dimension reduction methods to extract new components from original gene set and GA as feature selection methods to select feature subset from the newly extracted components, the selected subset is input into k NN.
- (f) GAPLSKNN uses PLSDR as dimension reduction methods to extract new components from original gene set and GA as feature selection methods to select feature subset from the newly extracted components, the selected subset is input into k NN.

- (g) GAPPKNN uses PCA+PLSDR as dimension reduction methods to extract new components from original gene set and GA as feature selection methods to select feature subset from the newly extracted components, the selected subset is input into k NN.

Since there are parameters for k NN, we try to reduce its effect to our comparison and use three parameters for k NN, they are $k = 1$ and $k = 4$.

It is noted that different data sets need different optimal parameters for different methods, we do not choose the optimal parameters, because we do not exhibit the top performance of one special method on one single data set, but we want to show the effect of our proposed framework.

Prediction performance

The average error rates and the corresponding standard deviation values are shown in Table 6, from which we can find the similar observations:

- (1) Results of all the classification methods with feature selection and extraction like PLSKNN, GAPLSKNN, PCAKNN, GAPCAKNN, GAPPKNN are better than that of KNN without any other dimension reduction on average and on each cases.
- (2) Results of classification methods with feature selection like GAPLSKNN, GAPCAKNN and GAPPKNN are better than those of the corresponding dimension reduction methods without feature selection like PLSKNN, PCAKNN and PPKNN on average and each cases.
- (3) Results of GAPPKNN are better than those of PCAKNN and GAPCAKNN, even the corresponding results of PLSKNN and GAPLSKNN on average. Only on the Lung data set out of four data sets, GAPLSKNN obtains the best results than other methods do.

Table 6. Statistical classification error rates (and their corresponding standard deviation) by using k NN with different different parameters on four microarray data sets (%)

DATASET	KNN	PCAKNN	GAPCAKNN	PLSKNN	GAPLSKNN	PPKNN	GAPPKNN
$k = 1$ for k NN							
CNS	47.5(3.5)	43.8(8.7)	40.8(10.2)	44.9(9.3)	36.4(10.7)	44.9(10.3)	34.3(8.3)
COLON	32.5(1.4)	28.4(8.4)	27.1(10.3)	24.8(14.3)	21.9(7.5)	30.2(12.2)	18.2(7.2)
LEUKEMIA	16.1(2.2)	14.7(10.8)	11.4(8.7)	15.7(10.4)	12.3(8.6)	15.9(11.9)	8.4(6.4)
LUNG	17.6(2.3)	11.8(7.1)	11.0(4.6)	11.8(5.3)	6.1(4.7)	13.2(5.6)	7.8(4.1)
Average	28.4(2.3)	24.6(8.7)	22.57(8.4)	24.3(9.8)	19.2(7.8)	25.3(10.0)	17.1(6.5)
$k = 4$ for k NN							
CNS	48.6(1.2)	46.5(11.2)	41.5(11.0)	44.9(9.9)	38.4(10.8)	47.8(9.9)	38.5(8.7)
COLON	44.6(2.8)	42.9(12.9)	36.2(9.2)	35.3(14.3)	28.8(8.8)	34.9(13.7)	24.5(8.4)
LEUKEMIA	32.5(1.9)	31.5(14.1)	28.1(11.9)	15.5(9.6)	14.8(14.9)	18.6(11.5)	10.0(7.8)
LUNG	16.2(0.8)	15.8(4.6)	13.5(4.8)	12.6(6.3)	10.1(4.8)	13.4(6.5)	9.4(3.5)
Average	35.4(1.7)	34.1(10.7)	28.8(9.2)	27.0(10.0)	23.0(8.0)	28.67(10.4)	20.6(7.1)

Table 7. Average percentage of features (and their corresponding standard deviation) used by k NN with different parameters on four microarray data sets (%)

DATASET	PCAKNN	GAPCAKNN	PLSKNN	GAPLSKNN	PPKNN	GAPPKNN
$k = 1$ for k NN						
CNS	68.5(6.5)	32.3(8.0)	69.2(8.0)	32.2(6.4)	62.5(7.3)	32.3(9.1)
COLON	78.2(4.4)	29.7(7.8)	58.3(5.2)	32.8(6.4)	61.4(9.0)	34.2(7.2)
LEUKEMIA	68.0(8.8)	28.6(6.2)	47.8(7.1)	31.2(7.6)	54.3(8.3)	33.3(6.8)
LUNG	73.4(6.2)	72.2(7.6)	78.4(7.2)	68.9(5.9)	79.8(8.1)	71.9(6.9)
Average	72.0(6.5)	40.7(7.4)	63.4(6.9)	41.2(6.6)	64.5(8.2)	42.9(7.5)
$k = 4$ for k NN						
CNS	71.2(6.8)	26.6(7.9)	68.2(7.8)	31.6(9.5)	62.4(9.8)	23.1(8.2)
COLON	80.3(7.5)	32.2(6.8)	59.7(5.2)	27.3(8.3)	62.3(8.8)	32.5(7.5)
LEUKEMIA	81.4(6.9)	26.7(5.7)	46.8(8.8)	35.5(7.1)	50.7(7.3)	33.2(6.0)
LUNG	78.2(8.7)	71.2(6.3)	74.7(6.0)	69.3(4.1)	80.2(8.9)	70.0(6.2)
Average	77.7(7.5)	39.1(6.7)	62.3(6.9)	40.9(7.2)	63.9(8.7)	39.7(7.0)

Number of selected features

We also show the number of features selected by each method in Table 7, where the values for PCAKNN means the ratios of the number of top principal components to that of extracted components, those of PLSKNN and PPKNN have the same meaning. The values for GAPCAKNN means the ratios of the number of selected components used in k NN to that of extracted components, and those of GAPLSKNN and GAPPKNN have the same meaning.

From Table 7, we can see that if we use the top components as in PCAKNN, PLSKNN and PPKNN, about 60–80% components are selected into learning machines, while if we use feature selection to select useful components as in GAPCAKNN, GAPLSKNN and GAPPKNN, about 30% components are selected on average. Only on the LUNG data set, the selected by different methods are 70–80% of extracted components.

Distribution of selected features

Fig. 5 shows the comparison of distributions of components selected by GA in two cases of GAPCAKNN and GAPLSKNN, and Fig. 6 shows that of GAPPKNN. Difference between Fig. 5 and Fig. 6 is that in Fig. 5, PCA and PLSDR are used as dimension reduction individually, while in Fig. 6, PCA is combined with PLSDR as dimension reduction methods.

From Fig. 5 and Fig. 6, we can find the similar observations as below:

- (1) When only PLSDR is used for dimension reduction, the top components are more than that of others in the selected components, but the others are also selected, the top, the more.
- (2) When only PCA is used, the top components is less than others in the selected features, and the tail components are more important than others.

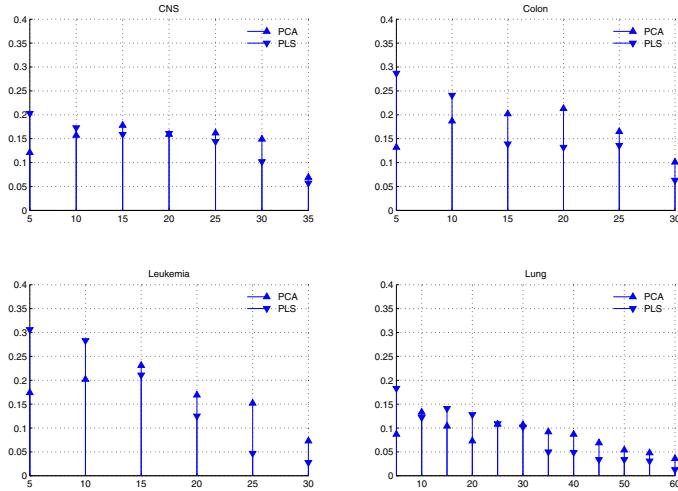


Fig. 5. comparison of distributions of eigenvectors used by GAPCAKNN and GAPLSKNN with $k = 1$ for k NN. X-axis corresponds to the eigenvectors in descending order by their eigenvalues and has been divided into bins of size 5. Y-axis corresponds to the average value of times that eigenvectors within some bin are selected by GA.

- (3) When both PCA and PLSDR are used as dimension reduction methods, they are nearly equal in the selected components, and the top components of PLSDR are a little more than others.

Discussions

The results are interesting, beyond our imagination, but they are reasonable.

From the experimental results, we know not the top components are important. The reason can be found in the subsection of dimension reduction. For PCA, components are extracted by maximizing the variance of a linear combination of the original genes, $\mathbf{u}_q = \arg \max_{\mathbf{u}' \mathbf{u}=1} (\text{Var}(X\mathbf{u}))$, but not maximizing the discriminative power for classifiers like k nearest neighbor (k NN). Therefore, the top component of PCA is not the top one with high discriminative power of classifiers. For PLSDR, components are extracted by maximizing the covariance between the response variable \mathbf{y} and the original genes X , $\mathbf{w}_q = \arg \max_{\mathbf{w}' \mathbf{w}=1} (\text{Cov}(X\mathbf{w}, \mathbf{y}))$. Therefore, the top component of PLSDR is more important than the others for classifiers. Furthermore, the top components of PCA are not the top feature subset with high discriminative power for classifiers, while the top ones of PLSDR are the top feature subset with high discriminative power, but the tail ones have also discriminative power, they are selected too. So, we should not only choose the top components, but employ feature selection methods to select a feature subset from the extracted components for classifiers.

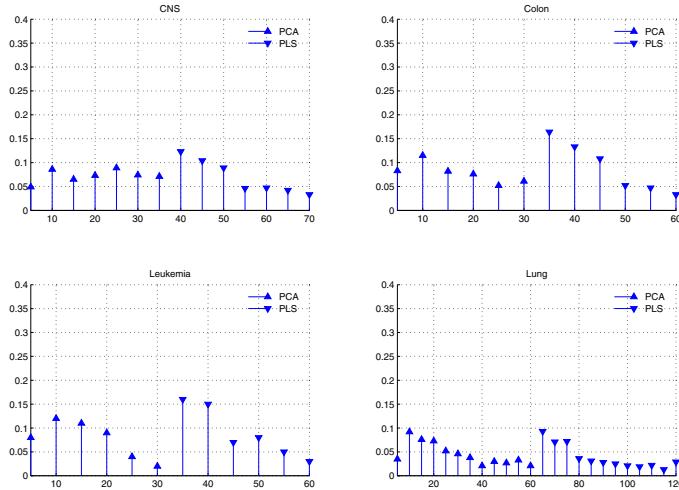


Fig. 6. Comparison of distributions of eigenvectors used by GAPPKNN with $k = 1$ for k NN. X-axis corresponds to the eigenvectors in descending order by their eigenvalues and has been divided into bins of size 5. Y-axis corresponds to the average value of times that eigenvectors within some bin are selected by GA.

Feature selection is performed by genetic algorithm (GA), which shows great power to select feature subsets for classifiers, this can be seen from the experimental results. Here genetic algorithm based feature selection is a so called wrapper model, which uses the classifier to measure the discriminative power of feature subsets from the extracted components. This method has been proved the best one feature selection method [46]. While this wrapper method is time consuming, nowadays, the scale of data sets is increasing rapidly, so efficient feature selection methods need be developed.

Partial least squares is superior to principal component analysis as dimension reduction methods. The reason is simple, PLSDR extracts components by maximizing the covariance between the response variable \mathbf{y} and the original genes X , which considers using the labels \mathbf{y} and can be viewed as a supervised method. While PCA extracts components by maximizing the variance of a linear combination of the original genes, which does not consider using the label \mathbf{y} and can be viewed as an unsupervised method. Here, we try to improve the classification accuracy of k NN, this is a supervised task, so PLSDR a supervised method is superior to PCA, an unsupervised method.

Features selected by different classifiers has minor difference, and results of prediction accuracy are also different. We have also conducted experiments on support vector machine (SVM), which show feature selection has done more effect on k NN than that on SVM. Because k NN is more sensitive on high dimensional data sets than SVM. But, they all benefit from feature selection.

6.4 Conclusion

We investigated a systematic feature reduction framework by combining dimension reduction with feature selection. To evaluate the proposed framework, we used four typical data sets. In each case, we used principal component analysis (PCA) and partial least squares (PLSDR) for dimension reduction, GA as feature selection, k nearest neighbor (k NN) for classification. Our experimental results illustrate that the proposed method improves the performance on the gene expression microarray data in accuracy. Further study of our experiments indicates that not all the top components of PCA and PLSDR are useful for classification, the tail component also contain discriminative information. Therefore, it is necessary to combine feature selection with dimension reduction and replace the traditional dimension reduction step as a new preprocessing step for analyzing high dimensional problems.

7 Summary and Future Direction

PLSDR is an effective and efficient way to reduce the size of original data. Especially, PLSDR is obviously more effective than other dimension reduction methods when the size of feature is much larger than that of observations [6]. To overcome the shortcomings of PLSR, we propose a general feature selection framework, where preprocessing and postprocessing are used to eliminate the irrelevant/redundant features and select the relevant principal components of PLSDR.

For preprocessing, we proposed a PLSDR^g algorithm to eliminate irrelevant features, which is more efficient than PLSDR without feature selection. At the same time PLSDR^g improves the generalization performance of classifiers. This has been proved by the experimental results on four microarray data sets. Furthermore, a supervised redundant feature elimination algorithm, REDISC, is proposed and obtains better performance than the widely used correlation algorithm.

For postprocessing, we proposed a wrapper strategy by combining accuracy of classifiers with genetic algorithm, which show better generalization performance in the experiments.

Since this is the first try to solve the problems during the PLSDR process, the ideas in this chapter are a start, more works need to do in future. The first is to try make REDISC more effective. The second is to fuse PLSDR^g with REDISC to remove irrelevant and redundant features simultaneously. The third is to introduce more feature selection methods into model selection of principal components, e.g. filter and embedded methods are more efficient than wrapper methods.

Extension of the works in this chapter includes application of PLSDR and related algorithms to other fields and make some modifications according the characteristics of the domain problems, e.g. text processing [22, 21]. If the problem is a multi-value classification case or an imbalanced case [48, 49], further investigation is needed. Since PLSDR is popular in the chemometrics field, it is necessary to perform a thorough comparison of PLSDR with other related dimension reduction methods in various scientific fields.

Table 8. Experimental microarray data sets

Data Sets	Number of Samples	Class Ratio	Number of Features
Breast Cancer	97	46/51	24,481
CNS	60	21/39	7,129
Colon	62	22/40	2,000
DLBCL	47	23/24	4,026
Leukemia	72	25/47	7,129
Lung	181	31/150	12,533
Ovarian	253	91/162	15,154
Prostate	136	59/77	12,600

A Description of Benchmark Data Sets

There are totally eight microarray data sets used in this chapter which are listed in Table 8.

A.1 Breast Cancer

Veer *et al.* [50] used DNA microarray analysis on primary breast tumors and applied supervised classification methods to identify significant genes for the disease. The data contains 97 patient samples, 46 of which are from patients who had developed distance metastases within 5 years (labeled as "relapse"), the rest 51 samples are from patients who remained healthy from the disease after their initial diagnosis for an interval of at least 5 years (labeled as "non-relapse"). The number of genes is 24,481 and the missing values of "NaN" are replaced with 100.

A.2 Central Nervous System

Pomeroy *et al.* developed a classification system based on DNA microarray gene expression data derived from 99 patient samples of Embryonal tumors of the central nervous system (CNS) [51]. Only data set C is used in our study. The data set contains 60 patient samples, 21 are survivors and 39 are failures. Survivors are patients who are alive after treatment whiles the failures are those who succumbed to their disease. There are 7,129 genes in the data set.

A.3 Colon

Alon *et al.* used Affymetrix oligonucleotide arrays to monitor expressions of over 6,500 human genes with samples of 40 tumor and 22 normal colon tissues [2]. Using two-way clustering, Alon *et al.* were able to cluster 19 normal and 5 tumor samples into one group and 35 tumor and 3 normal tissues into the other. Expression of the 2,000 genes with highest minimal intensity across the 62 tissues were used in the analysis.

A.4 DLBCL

Alizadeh *et al.* [52] used gene expression data to analyze distinct types of diffuse large B-cell lymphoma (DLBCL). DLBCL is the most common subtype of non-Hodgkin's lymphoma. There are 47 samples, 24 of them are from "germinal centre B-like" group and 23 are "activated B-like" group. Each sample is described by 4,026 genes. The missing values in the data set are replaced by the corresponding averaged column values.

A.5 Leukemia

The acute leukemia data set was published by Golub *et al.* [1]. The original training data set consists of 38 bone marrow samples with 27 ALL and 11 AML (from adult patients). The independent (test) data set consists of 24 bone marrow samples as well as 10 peripheral blood specimens from adults and children (20 ALL and 14 AML). Four AML samples in the independent data set are from adult patients. The gene expression intensities are obtained from Affymetrix high-density oligonucleotide microarrays containing probes for 7,129 genes.

A.6 Lung Cancer

Gordon *et al.* [53] proposed a data set for the purpose of classifying lung Cancer between malignant pleural mesothelioma (MPM) and adenocarcinoma (ADCA) of the lung. The data set includes 181 tissue samples (31 MPM and 150 ADCA). Each sample is described by 12,533 genes.

A.7 Ovarian

Petricoin *et al.* [54] identified proteomic patterns in serum to distinguish ovarian cancer from non-cancer. The proteomic spectral data includes 91 controls (Normal) and 162 ovarian cancers, each sample contains the relative amplitude of the intensity at 15,154 molecular mass/charge (M/Z) identities.

A.8 Prostate

Singh *et al.* used microarray expression analysis to determine whether global biological differences underlie common pathological features of prostate cancer and to identify genes that might anticipate the clinical behavior of Prostate tumors [55]. In Singh's experiments, the training set contains 52 prostate tumor samples and 50 non-tumor (labeled as "Normal") prostate samples with around 12,600 genes. An independent set of test samples is also prepared, which is from a different experiment and has a nearly 10-fold difference in overall microarray intensity from the training data. After removing extra genes, 25 tumor and 9 normal samples were left in the test samples.

Acknowledgements

Thanks go to the anonymous reviewers for their valuable comments. This work was supported by the Natural Science Foundation of China under grant no. 20503015 and 60873129, the STCSM "Innovation Action Plan" Project of China under grant no. 07DZ19726, the Shanghai Rising-Star Program under grant no. 08QA1403200 and Scientific Research Fund of Jiangxi Provincial Education Departments under grant no. 2007-57.

References

1. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S.: Molecular classification of cancer: Class discovery and class prediction by gene expression. *Bioinformatics & Computational Biology* 286, 531–537 (1999)
2. Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. In: *Proceedings of the National Academy of Sciences of the United States of America*, pp. 6745–6750 (1999)
3. Dudoit, S., Fridlyand, J., Speed, T.P.: Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association* 97, 77–87 (2002)
4. Antoniadis, A., Lambert-Lacroix, S., Leblanc, F.: Effective dimension reduction methods for tumor classification using gene expression data. *Bioinformatics* 19, 563–570 (2003)
5. Nguyen, D.V., David, D.M., Rocke, M.: On partial least squares dimension reduction for microarray-based classification: a simulation study. *Computational Statistics & Data Analysis* 46, 407–425 (2004)
6. Dai, J.J., Lieu, L., Rocke, D.: Dimension reduction for classification with gene expression data. *Statistical Applications in Genetics and Molecular Biology* 6, Article 6 (2006)
7. Boulesteix, A.L., Strimmer, K.: Partial least squares: A versatile tool for the analysis of high-dimensional genomic data. *Briefings in Bioinformatics* 8, 32–44 (2007)
8. Wold, H.: Path models with latent variables: the NIPALS approach. In: *Quantitative Sociology: International Perspectives on Mathematical and Statistical Model Building*, pp. 307–357. Academic Press, London (1975)
9. Wold, S., Ruhe, A., Wold, H., Dunn, W.: Collinearity problem in linear regression the partial least squares (pls) approach to generalized inverses. *SIAM Journal of Scientific and Statistical Computations* 5, 735–743 (1984)
10. Martens, H.: Reliable and relevant modeling of real world data: a personal account of the development of pls regression. *Chemometrics and Intelligent Laboratory Systems* 58, 85–95 (2001)
11. Helland, I.S.: On the structure of partial least squares regression. *Communications in statistics. Simulation and computation* 17, 581–607 (1988)
12. Wold, S., Sjostrom, M., Eriksson, L.: Pls-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems* 58, 109–130 (2001)
13. Helland, I.S.: Some theoretical aspects of partial least squares regression. *Chemometrics and Intelligent Laboratory Systems* 58, 97–107 (2001)
14. Nguyen, D.V., Rocke, D.M.: Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics* 18, 39–50 (2002)

15. Nguyen, D.V., Rocke, D.M.: Multi-class cancer classification via partial least squares with gene expression profiles. *Bioinformatics* 18, 1216–1226 (2002)
16. Zeng, X.Q., Li, G.Z., Wu, G.: On the number of partial least squares components in dimension reduction for tumor classification. In: BioDM 2007. LNCS (LNBI), vol. 4819, pp. 206–217. Springer, Heidelberg (2007)
17. Bu, H.L., Li, G.Z., Zeng, X.Q., Yang, M.Q., Yang, J.Y.: Feature selection and partial least squares based dimension reduction for tumor classification. In: Proceedings of IEEE 7th International Symposium on Bioinformatics & Bioengineering (IEEE BIBE 2007), Boston, USA, pp. 1439–1444. IEEE Press, Los Alamitos (2007)
18. Zeng, X.Q., Li, G.Z., Wu, G.F., Yang, J.Y., Yang, M.Q.: Irrelevant gene elimination for partial least squares based dimension reduction by using feature probes. *International Journal of Data Mining & Bioinformatics* (in press) (2008)
19. Li, G.Z., Zeng, X.Q., Yang, J.Y., Yang, M.Q.: Partial least squares based dimension reduction with gene selection for tumor classification. In: Proceedings of IEEE 7th International Symposium on Bioinformatics & Bioengineering (IEEE BIBE 2007), Boston, USA, pp. 967–973 (2007)
20. Zeng, X.Q., Li, G.Z., Yang, J.Y., Yang, M.Q., Wu, G.F.: Dimension reduction with redundant genes elimination for tumor classification. *BMC Bioinformatics* 9(suppl. 6), 8 (2008)
21. Zeng, X.Q., Wang, M.W., Nie, J.Y.: Text classification based on partial least square analysis. In: The 22nd Annual ACM Symposium on Applied Computing, Special Track on Information Access and Retrieval, pp. 834–838 (2007)
22. Zeng, X.Q., Li, G.Z., Wang, M., Wu, G.F.: Local semantic indexing based on partial least squares for text classification. *Journal of Computational Information Systems* 4, 1145–1152 (2008)
23. Zeng, X.Q., Li, G.Z.: Orthogonal projection weights in dimension reduction based on partial least squares. *International Journal of Computational Intelligence of Bioinformatics & System Biology* 1(1), 105–120 (2008)
24. Nguyen, D.V., Rocke, D.M.: Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics* 18, 39–50 (2002)
25. Dai, J.J., Lieu, L., Rocke, D.: Dimension reduction for classification with gene expression microarray data. *Statistical Applications in Genetics and Molecular Biology* 5(1), Article 6 (2006)
26. Wold, S., Sjostrom, M., Eriksson, L.: PLS-regression: A basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems* 58, 109–130 (2001)
27. Barker, M., Rayens, W.: Partial least squares for discrimination. *Journal of Chemometrics* 17, 166–173 (2003)
28. Hoskuldsson, A.: PLS regression methods. *Journal of Chemometrics* 2, 211–228 (1988)
29. Manne, R.: Analysis of two partial-least-squares algorithms for multivariate calibration. *Chemometrics and Intelligent Laboratory Systems* 2, 187–197 (1987)
30. Yu, L., Liu, H.: Redundancy based feature selection for microarray data. In: Proc. 10th ACM SIGKDD Conf. Knowledge Discovery and Data Mining, pp. 22–25 (2004)
31. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research* 5, 1205–1224 (2004)
32. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley Interscience, Hoboken (2000)
33. Levner, I.: Feature selection and nearest centroid classification for protein mass spectrometry. *BMC Bioinformatics* 6, 68 (2005)
34. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation* 10, 1895–1923 (1998)

35. Molinaro, A.M., Simon, R., Pfeiffer, R.M.: Prediction error estimation: A comparison of resampling methods. *Bioinformatics* 21, 3301–3307 (2005)
36. Witten, I., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005)
37. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *Journal of Machine Learning Research* 3, 1157–1182 (2003)
38. Forman, G.: An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research* 3, 1289–1305 (2003)
39. Hall, M.A., Holmes, G.: Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering* 15, 1437–1447 (2003)
40. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation* 10, 1895–1923 (1998)
41. Bu, H.L., Li, G.Z., Zeng, X.Q.: Reducing error of tumor classification by using dimension reduction with feature selection. *Lecture Notes in Operations Research* 7, 232–241 (2007)
42. Li, G.Z., Bu, H.L., Yang, M.Q., Zeng, X.Q., Yang, J.Y.: Selecting subsets of newly extracted features from PCA and PLS in microarray data analysis. *BMC Genomics* 9(S2), S24 (2008)
43. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. *Machine Learning* 46, 389–422 (2002)
44. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *Journal of Machine Learning Research* 3, 1157–1182 (2003)
45. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering* 17(3), 1–12 (2005)
46. Kudo, M., Sklansky, J.: Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition* 33, 25–41 (2000)
47. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Boston (1998)
48. Li, G.Z., Meng, H.H., Ni, J.: Embedded gene selection for imbalanced microarray data analysis. In: *Proceedings of Third IEEE International Multisymposium on Computer and Computational Sciences (IEEE- IMSCCS 2008)*. IEEE Press, Los Alamitos (in press) (2008)
49. Li, G.Z., Meng, H.H., Lu, W.C., Yang, J.Y., Yang, M.Q.: Asymmetric bagging and feature selection for activities prediction of drug molecules. *BMC Bioinformatics* 9(suppl. 6), 7 (2008)
50. Van't Veer, L.V., Dai, H., Vijver, M.V., He, Y., Hart, A., Mao, M., Peterse, H., Kooy, K., Marton, M., Witteveen, A., Schreiber, G., Kerkhoven, R., Roberts, C., Linsley, P., Bernards, R., Friend, S.: Gene expression profiling predicts clinical outcome of breast cancer. *Nature* 415, 530–536 (2002)
51. Pomeroy, S.L., Tamayo, P., Gaasenbeek, M., Sturla, L.M., Angelo, M., McLaughlin, M.E., Kim, J.Y., Goumnerovak, L.C., Blackk, P.M., Lau, C., Allen, J.C., Zagzag, D., Olson, J.M., Curran, T., Wetmo, C.: Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature* 415, 436–442 (2002)
52. Alizadeh, A.A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwald, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X., Powell, J.I., Yang, L., Marti, G.E., Moore, T., Jr, J.H., Lu, L., Lewis, D.B., Tibshirani, R., Sherlock, G., Chan, W.C., Greiner, T.C., Weisenburger, D.D., Armitage, J.O., Warnke, R., Levy, R., Wilson, W., Grever, M.R., Byrd, J.C., Botstein, D., Brown, P.O., Staudt, L.M.: Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature* 403, 503–511 (2000)

53. Gordon, G.J., Jensen, R.V., Hsiao, L.L., Gullans, S.R., Blumenstock, J.E., Ramaswamy, S., Richards, W.G., Sugarbaker, D.J., Bueno, R.: Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Research* 62, 4963–4967 (2002)
54. Petricoin, E.F., Ardekani, A.M., Hitt, B.A., Levine, P.J., Fusaro, V.A., Steinberg, S.M., Mills, G.B., Simone, C., Fishman, D., Kohn, E.C., Liotta, L.: Use of proteomic patterns in serum to identify ovarian cancer. *The Lancet* 359, 572–577 (2002)
55. Singh, D., Febbo, P.G., Ross, K., Jackson, D.G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A.A., D'Amico, A.V., Richie, J., Lander, E.S., Loda, M., Kantoff, P.W., Golub, T.R., Sellers, W.R.: Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell* 1, 203–209 (2002)

Classification by the Use of Decomposition of Correlation Integral

Marcel Jiřina¹ and Marcel Jiřina Jr.²

¹ Institute of Computer Science, Pod vodarenskou vezi 2,
182 07 Prague 8 – Liben, Czech Republic
marcel@cs.cas.cz

² Faculty of Biomedical Engineering, Czech Technical University in Prague,
Nam. Sitna 3105, 272 01, Kladno, Czech Republic
jirina@fbmi.cvut.cz

Summary. The correlation dimension is usually used to study features of fractals and data generating processes. For estimating the value of the correlation dimension in a particular case, a polynomial approximation of correlation integral is often used and then linear regression for logarithms of variables is applied. In this Chapter, we show that the correlation integral can be decomposed into functions each related to a particular point of data space. For these functions, one can use similar polynomial approximations such as the correlation integral. The essential difference is that the value of the exponent, which would correspond to the correlation dimension, differs in accordance to the position of the point in question. Moreover, we show that the multiplicative constant represents the probability density estimation at that point. This finding is used to construct a classifier. Tests with some data sets from the Machine Learning Repository show that this classifier can be very effective.

1 Introduction

A lot of tasks of data mining have to do with associating objects to a limited number of types or classes. A typical task is whether an e-mail is spam or not. This is a classification into two classes. Many other tasks may be recognized as classification into several classes. Usually, objects to be classified are not used directly, but are described by some number of parameters (or features, variables etc.) There are many approaches to classification, simple ones or very sophisticated ones. In this chapter, an approach closely related to the characterization of fractals by the correlation dimension is introduced.

The target of this Chapter is to show that one can construct a classifier for multivariate data that uses fractal nature of data and provides a very low classification error. We show that the correlation integral can be decomposed in functions each related to particular point x of data space. For these functions one can use similar polynomial approximations as is usually used

for correlation integral. The value of exponent q , which corresponds to the correlation dimension, differs in accordance to the position of the point x in question. Moreover, we show that the multiplicative constant C in these cases represents the probability density estimation at point x . This finding is used to construct a classifier. Tests with some data sets from the Machine Learning Repository [5] show that this classifier can have a very low classification error.

2 Decomposition of the Correlation Integral

We work in n -dimensional metric space with L_2 (Euclidean) or L_1 (taxicab or Manhattan) metrics.

2.1 Correlation Integral

The correlation integral, in fact, a distribution function of all binate distances in a set of points in a space with a distance was introduced by Grassberger and Procaccia in 1983 [1]. Camastra and Vinciarelli [6] consider the set $\{X_i, i = 1, 2, \dots, N\}$ of points of the attractor. This set of points may be obtained e.g. from a time series with a fixed time increment. Most pairs (X_i, X_j) with $i \neq j$ are dynamically uncorrelated pairs of essentially random points [1]. However, the points lie on the attractor. Therefore, they will be spatially correlated. This spatial correlation is measured by the correlation integral $C_I(r)$ defined according to:

$$C_I(r) = \lim_{N \rightarrow \infty} \frac{1}{N^2} \times \{\text{number of pairs } (i, j) : \|X_i - X_j\| < r\}.$$

In a more comprehensive form one can write:

$$C_I(r) = \Pr(\|X_i - X_j\| < r).$$

Grassberger and Procaccia [1] have shown that for a small r the $C_I(r)$ grows like a power $C_I(r) \sim r^\nu$ and that the "correlation exponent" ν can be taken as a most useful measure of the local structure of the strange attractor. This measure allows one to distinguish between deterministic chaos and random noise [6]. These authors also mention that the correlation exponent (dimension) ν seems to be more relevant in this respect than the Hausdorff dimension D_h of the attractor. In general, there is $\nu \leq \sigma \leq D_h$, where σ is the information dimension [4], and it can be found that these inequalities are rather tight in most cases, but not all cases. Given an experimental signal and $\nu < n$ (n is the degree of freedom or the dimensionality or the so-called embedding dimension), then we can conclude that the signal originates from deterministic chaos rather than random noise, since random noise will always result in $C_I(r) \sim r^n$.

The correlation integral can be rewritten in form [6]

$$C_I(r) = \lim_{N \rightarrow \infty} \frac{1}{N(N-1)} \sum_{1 \leq i < j \leq N} h(r - \|X_j - X_i\|),$$

where $h(\cdot)$ is Heaviside step function. From it

$$\nu = \lim_{r \rightarrow \infty} \frac{\ln C_I(r)}{\ln r}.$$

The correlation dimension [1], [2] as well as other effective dimensions - Hausdorff, box-counting, information dimension [3], [4] - is used to study features of different fractals and data generating processes. For estimation of the value of the correlation dimension in a particular case, linear regression is often used for logarithms of variables [1]. We write it in the form:

$$\ln(s) = \ln(C) + q \ln(r_s), \quad s = 1, 2, \dots \quad (1)$$

Here, ν is a correlation dimension and C is a multiplicative constant in the relation:

$$s = Cr_s^q, \quad s = 1, 2, \dots \quad (2)$$

Constant C has no particular meaning.

There are other methods for estimating the correlation dimension ν , but the problem is that they are either too specialized for one kind of equation or they use some kind of heuristics that usually optimize the size of radius r to get the proper value of the correlation dimension. One of the most cited is Taken's estimator [7], [8], [9].

2.2 Probability Distribution Mapping Function

Two important notions, the probability distribution mapping function and the distribution density mapping function are introduced here. We use these notions for developing a decomposition of the correlation integral and a new classifier. To understand these terms, we give a brief example that demonstrates them.

Let a query point x be placed without loss of generality in the origin. Let us build balls with their centers at point x and with volumes V_i , $i = 1, 2, \dots$

The individual balls are in one another, the $(i-1)$ -st inside the i -th are like peels of an onion. Then the mean density of the points in the i -th ball is $\rho_i = m_i/V_i$. The volume of the ball of radius r in n -dimensional space is $V(r) = \text{const.}r^n$. Thus, we have constructed a mapping between the mean density ρ_i in the i -th ball ρ_i and its radius r_i . Then $\rho_i = f(r_i)$. Using a tight analogy between the density $\rho(z)$ and the probability density $p(z)$, one can write $p(r_i) = f(r_i)$, and $p(r_i)$ is the mean probability density in the i -th ball

with radius r_i . This way, a complex picture of the probability distribution of the points in the neighborhood of a query point x is simplified to a function of a scalar variable. We call this function the probability distribution mapping function $D(x, r)$, where x is a query point, and r the distance from it. More exact definitions follow:

Definition 1. *The probability distribution mapping function $D(x, r)$ of the neighborhood of the query point x is the function $D(x, r) = \int_{B(x,r)} p(z)dz$,*

where r is the distance from the query point and $B(x, r)$ is a ball with center x and radius r .

Definition 2. *The distribution density mapping function $d(x, r)$ of the neighborhood of the query point x is function $d(x, r) = \frac{\partial}{\partial r} D(x, r)$, where $D(x, r)$ is a probability distribution mapping function of the query point x and radius r .*

Note: It can be seen that for a fixed x , the function $D(x, r)$, $r > 0$ is monotonically growing from zero to one. Functions $D(x, r)$ and $d(x, r)$ for a fixed x are one-dimensional analogs to the probability distribution function and the probability density function, respectively.

One can write the probability distribution mapping function in the form

$$D(x, r) = \lim_{N \rightarrow \infty} \frac{1}{N-1} \sum_{j=1}^{N-1} h(r - r_j), \quad (3)$$

where $h(\cdot)$ is the Heaviside step function. For a finite number of points, we have the empirical probability distribution mapping function

$$D'(x, r) = \frac{1}{N-1} \sum_{j=1}^{N-1} h(r - r_j).$$

2.3 Power Approximation of the Probability Distribution Mapping Function

Let us introduce a simple polynomial function in the form $D(x, r) = Cr^q$. We shall call it a power approximation of the probability distribution mapping function $D(x, r)$. Exponent q is a distribution-mapping exponent.

Definition 3. *The power approximation of the probability distribution mapping function $D(x, r^q)$ is the function r^q such that $\frac{D(x, r^q)}{r^q} \rightarrow C$ for $r \rightarrow 0+$. The exponent q is a distribution-mapping exponent.*

Using this approximation of the probability distribution mapping function $D(x, r)$, we, in fact, linearize this function as a function of the variable $z = r^q$ in the neighborhood of the origin, i.e. in the neighborhood of the

query point. The distribution density mapping function $d(x, r)$, as a function of the variable $z = r^q$, is approximately constant in the vicinity of the query point. This constant includes a true distribution of the probability density of the points as well as the influence of boundary effects.

An important fact is that the distribution-mapping exponent reminds us of the correlation dimension by Grassberger and Procaccia [1]. Although, there are three essential differences: First, the distribution-mapping exponent is a local feature of the data set because it depends on a position of the query point, whereas the correlation dimension is a feature of the whole data space. Second, the distribution mapping exponent is related to the data only and not to a fractal or data generating process by which we can have an unlimited number of data points. Third, the distribution mapping exponent is influenced by boundary effects, which have a larger influence with a larger dimension n and a smaller learning set size [6], [10].

2.4 Decomposition of Correlation Integral to Local Functions

We show, in this section, that the correlation integral is the mean of the distribution mapping function and that the correlation dimension can be approximated by the mean of the distribution mapping exponent as shown in the theorem below:

Theorem 1. *Let there be a learning set of N points (samples). Let the correlation integral, i.e. the probability distribution of binate distances of the points from the learning set, be $C_I(r)$ and let $D(x_i, r)$ be the distribution mapping function corresponding to point x_i . Then, $C_I(r)$ is a mean value of $D(x_i, r)$:*

$$C_I(r) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N D(x_i, r). \quad (4)$$

Proof. Let $h(x)$ be a Heaviside step function and l_{ik} be the distance of k -th neighbor from point x_i . Then the correlation integral is

$$C_I(r) = \lim_{N \rightarrow \infty} \frac{1}{N(N-1)} \sum_{i=1}^N \sum_{j=1}^{N-1} h(r - l_{ij})$$

and also

$$C_I(r) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \left(\frac{1}{N-1} \sum_{j=1}^{N-1} h(r - l_{ij}) \right). \quad (5)$$

Comparing (5) with (3) we get (4) directly.

The correlation dimension ν can be approximated as a mean of the distribution mapping exponents

$$\nu = \frac{1}{N} \sum_{i=1}^N q_i.$$

2.5 Distribution Mapping Exponent Estimation

Let U be a learning set composed of points (patterns, samples) x_{cs} , where $c = \{0, 1\}$ is the class mark and $s = 1, 2, \dots, N_c$ is the index of the point within class c . N_c is the number of points in class c and let $N = N_0 + N_1$ be the learning set size.

Let point $x \notin U$ be given and let points x_{cs} of one class be sorted so that index $s = 1$ corresponds to the nearest neighbor, index $s = 2$ to the second nearest neighbor, etc. In Euclidean metrics, $r_s = \|x - x_{cs}\|$ is the distance of the s -th nearest neighbor of class c from point x .

We look for exponent q so, that r_s^q is proportional to index s , i.e. for polynomial approximation

$$s = Cr_s^q, \quad s = 1, 2, \dots, N_c, c = 0 \text{ or } 1, \quad (6)$$

where C is a suitable constant. Using a logarithm we get

$$\ln(s) = \ln(C) + q \ln(r_s), \quad s = 1, 2, \dots, N_c. \quad (7)$$

On one hand, we exaggerate distances nonlinearly to make small differences in the distance appear much larger for the purposes of density estimation. On the other hand, there is a logarithm of distance in (7), which decreases large influences of small noise perturbations on the final value of q . Note that it is the same problem as in the correlation dimension estimation where equations of the same form as (6) and (7) arise. Grassberger and Procaccia [1] proposed a solution by linear regression. In [2], [9], [11] different modifications and heuristics were later proposed. Many of these approaches and heuristics can be used for distribution mapping exponent estimation, e.g. use a half or a square root of N_c nearest neighbors instead of N_c to eliminate the influence of the limited number of the points of the learning set.

The system of N_c (or $N_c/2$ or $\sqrt{N_c}$ as mentioned above) equation (7) with respect to an unknown q can be solved using standard linear regression for both classes. Thus, for two classes, we get two values of q , q_0 and q_1 and two values of C' , C'_0 and C'_1 .

At this point we can say that q_c is something like a local effective dimensionality of the data space including the true distribution of the points of each class. At the same time, we get the constant C'_c . The values of q_c and C'_c are related to each particular point x and thus they vary from one point x to another.

2.6 Probability Density Estimation

Let $n'_c(r)$ be a number of points of class c up to distance r from the query point x . Let q_c be the distribution mapping exponent for the points of class c and let

$$z_c = r^{q_c}. \quad (8)$$

Also, let $n_c(z_c) = n'_c(r) = n'_c(z_c^{1/q^c})$. Then $P_c(z_c) = n_c(z_c)/N$ is a percentage of all points of class c up to distance $r = z_c^{1/q^c}$ from the query point x , i.e. up to a “distance” measured by z_c from point x .

Due to polynomial approximation (6), $n_c(z_c) = C'_c z_c$. It is a number of points up to distance r , which is related to z_c according to (8). The derivative according to z_c is $dn_c(z_c)/dz_c = C'_c$ and it represents a number of points of class c on a unit¹ of the z_c , i.e., in fact, a density of points with respect to z_c .

By dividing with total number of points N , we get a percentage of points of class c on a unit of z_c . This percentage is equal to $p(c|x, z_c) = C'_c/N$. In the limit case for $r \rightarrow 0$ (and z_c as well) there is $p(c|x, 0) = p(c|x) = C'_c/N = C_c$.

Finally, as there are two classes, there must be $p(0|x) = p(1|x) = 1$ and then $C'_0 + C'_1 = N$. This result includes a priori probabilities N_c/N for both classes. When we need to exclude a priori probabilities we use the formula:

$$p(c|x) = \frac{C'_c/N_c}{C'_0/N_0 + C'_1/N_1}. \quad (9)$$

The generalization of the too many classes case is straightforward. For k classes there is

$$p(c|x) = \frac{C'_c/N_c}{\sum_{i=1}^k C'_i/N_i} \quad c = 1, 2, \dots, k. \quad (10)$$

A more exact development follows:

Definition 4. Let \mathbf{N} be n -dimensional space with metrics ρ . Let there be a subset $\mathbf{Q} \subseteq \mathbf{N}$ and a number $q \in \mathbf{R}_+$, $1 \leq q \leq n$ associated with subset \mathbf{Q} . A q -dimensional ball with center at point $x \in \mathbf{Q}$ and radius r is $B_q = B(q, x, r, \rho) = \{y \in \mathbf{Q}: \rho(x, y) < r\}$. The volume of B_q is $V(q, x, r, \rho) = S(q, \rho) \cdot r^q$, where $S(q, \rho)$ is a function independent of r .

Note: The metrics ρ can be omitted when it is clear what metrics we are dealing with.

Lemma 1. Let $B(q, x, R)$ be a q -dimensional ball with center at point $x \in \mathbf{g}$ and radius R , and let $V(q, x, r)$ be its volume. Let points in \mathbf{Q} in the neighborhood of point x up to distance R be distributed with the constant probability density $p = p_0$. Then, for $r < R$, where r is the distance from point x , the distribution function is given by

$$P(x, r) = \int_{B(q, x, R)} p dr = \int p dV(q, x, r) = p_0 V(q, x, r).$$

¹ We cannot say “unit length” here, as the dimensionality of z_c is $(length)^{qc}$.

The proof is obvious.

Conversely, let in **Q** hold $P(x, r) = p_0 V(q, x, r)$, where p_0 is a constant as long as $r < R$. It is obvious that this can be fulfilled even when the distribution density is not a constant. On the other hand, it is probably a rare case. Then we can formulate an assumption.

Assumption 1

If in **Q** holds $P(x, r) = p_0 V(q, x, r)$ then it holds $p(x) = p_0$.

Illustration

A sheet of white paper represents 2 dimensional subspace embedded in 3 dimensional space. Let point x be in the center of the sheet. White points of paper are uniformly distributed over the sheet with some constant (probability) density and a distribution function (frequentistically the number of white points) is proportional to the circular area around point x . Thus, the distribution function grows quadratically with distance r from point x , and only linearly with the size of the circular area. And the size of circular area is nothing other than the volume of the two-dimensional ball embedded in 3 dimensional space.

Theorem 2

Let, in a metric space, each point belongs to one of two classes $c = \{0, 1\}$. Let, for each point x and each class c , a distribution mapping function $D(x, c, r)$ exist where r is the distance from point x . Let Assumption 1 hold and the power approximation of the distribution mapping function be $C_c r^{q_c}$, where q_c is the distribution mapping exponent for point x and class c . Then it holds $p(c|x) = C_c S(q) = p_0$.

Proof

Let $z_c = r^{q_c}$ be a new variable. We can rewrite $D(x, c, r)$ as a function of variable z_c in the form $D(x, c, z_c)$. The $D(x, c, z_c)$ is, in fact, a distribution function of the points of class c with respect to variable z_c . When using a power approximation, we can write $D(x, c, z_c) = C_c r^q = C_c z_c$. This distribution function corresponds to uniform distribution in a subspace of dimension q_c . We express r^q with the help of the volume of the ball in q_c dimensional space with center x and radius r : $D(x, c, z_c) = C_c V(q_c, x, r)/S(q_c) = P(x, r)/S(q_c)$. From Assumption 1, it follows $d(x, c, z_c) = C_c = p(x, r)/S(q_c)$ and then $p(x, r) = C_c S(q_c) = p_0$.

Note: We see that beyond the unit ball volume $S(q_c)$, the proportionality constant C_c governs the probability density in the neighborhood of point x including this point. Also note that due to the ratios in formulas (9) and (10) the volume $S(q_c)$ of the unit ball in a q_c dimensional space in the probability estimation is eliminated.

2.7 Classifier Construction

In this section, we show how to construct a classifier that incorporates the idea above. Using formulas (9) or (10) we have a relatively simple method for estimating the probabilities $p(c|x)$. First, we sort the points of class c according to their distances from the query point x . Then, we solve the linear regression equation

$$q_c \ln(r_s) = \ln(C_c) + \ln(s), \quad s = 1, 2, \dots, K \quad (11)$$

for the first K points especially with respect to the unknown C_c . Number K may be a half or a square root or so of the total number N_c of the points of class c . This is made for all k classes, $c = 1, 2, \dots, k$. Finally, we use formula (9) for $k = 2$ or formula (10) for more than two classes. Formulas (9) or (10) give a real number. For two class classification, a discriminant threshold (cut) θ must be chosen, and then if $p(1|x) > \theta$, then x belongs to class 1 or else to class 0. The default value of θ is 0.5.

2.8 Error Analysis

There are two sources of errors. The first one depends on choosing the proper constant K , i.e. the number of nearest points to point x which is also the number of regression equations (11) used for computation of C_c . This is a problem very similar to the problem of the correlation dimension estimation. For correlation dimension estimation, many approaches including a lot of heuristic ones exist, see e.g. [2], [9], [11]; we do not discuss it in detail here.

The other kind of error is an error of estimation by linear regression. The Gauss–Markov theorem [12] states that in a linear model in which the errors have an expectation of zero and are uncorrelated and have equal variance, the best linear unbiased estimators of the coefficients are the least-squares estimators. At the same time, it holds that the regression coefficients, as random variables, have normal distribution [12], [14] each with a mean equal to the true value and with variance given by the well-known formulae [13] [14]. When the data is of the same quality, the variance converges to zero proportionally to $1/K$ for the number of samples K going to infinity.

In our case Gauss–Markov assumptions are well fulfilled, especially the assumption of homoscedasticity, i.e., all errors have the same variance. It is given by fact that each class usually represents a particular “source” of data with a particular statistic. Regression equations are constructed for each class separately here, i.e. all samples should have the same or very similar statistical characteristics including variance.

Variable $\ln(C_c)$ is found by linear regression and has normal distribution. Then variable C_c has lognormal distribution. From it, it follows that if $\mu_{\ln C_c}$ is the mean (also mode and median) of $\ln(C_c)$ and $\sigma_{\ln C_c}^2$ its variance then variable $C_c = \exp(\ln(C_c))$ has the median $M_e = \exp(\mu_{\ln C_c})$. The mean of C_c is $\exp(\ln(C_c) + \sigma_{\ln C_c}^2 / 2)$, i.e. it is slightly larger than the median. On the

other hand, the mode is slightly smaller as it holds that $M_o = \exp(\ln(C_c) - \sigma_{\ln C_c}^2)$. Considering these three measures of position, we use the median for C_c estimation, using formula $C_c = \exp(\ln(C_c))$. $\ln(C_c)$ is found by the linear regression above. For variance of the lognormal distribution, it holds:

$$\sigma_{C_c}^2 = (\exp(\sigma_{\ln C_c}^2) - 1) \cdot \exp(2\mu_{\ln C_c} + \sigma_{\ln C_c}^2).$$

From the fact that variance of regression coefficients converge to zero proportionally to $1/K$ for the number of samples K going to infinity, the $\sigma_{C_c}^2$ converges to zero proportionally to $1/K$ as well. Simply, for a small $\sigma_{\ln C_c}^2$ there is $\exp(\sigma_{\ln C_c}^2) \approx 1 + \sigma_{\ln C_c}^2$ and $\exp(2\mu_{\ln C_c} + \sigma_{\ln C_c}^2) = (\exp(\mu_{\ln C_c}))^2(1 + \sigma_{\ln C_c}^2) \approx C_c^2$. Then $\sigma_{C_c}^2 \approx \sigma_{\ln C_c}^2 C_c^2$ and because $\sigma_{\ln C_c}^2 \sim 1/K$ and C_c^2 is a constant here then $\sigma_{C_c}^2 \sim 1/K$.

We can conclude that variable C_c converges to its true value as fast as the standard linear regression (11) used for estimation of its logarithm $\ln(C_c)$.

Error estimation

When using linear regression for (11), it is easy to state individual residuals ρ_i and thus to know the true sum of the squared residuals $\rho = \sum_{i=1}^K \rho_i^2$. The standard deviation on a parameter estimate is $\hat{\sigma}_j = \sqrt{\frac{\rho}{K-1}} [(X^t X)^{-1}]_{jj}$, $j=1, 2$ and the $100(1-\alpha)\%$ confidence interval is $\hat{\beta}_j \pm t_{\frac{\alpha}{2}, K-2} \hat{\sigma}_j$. Variables ρ and $[(X^t X)^{-1}]_{jj}$ are known during computation of $\ln(C_c)$ and thus one can get the confidence interval for $\ln(C_c)$ which is symmetric. Due to exponential transformation, C_c has an asymmetric confidence interval.

This confidence interval computation can be easily included into the construction of the classifier.

3 Experimental Results

The method described above has one free parameter to be set up, the number of nearest neighbors used for linear regression. We tested different possibilities, e.g. the square root of the total number of samples N_c of the learning set, one third, and one half of the number of samples of the learning set, and a simplest robust modification of the linear regression. We found that the use of a half of the total number N_c of samples of the learning set often to be quite practical.

Another strategy uses a robust procedure in linear regression. The approach starts with half of the points of the learning set nearest to the query point in the same way as the previous one. In this step, the largest residuum is found and the corresponding sample of the learning set is excluded. This procedure is repeated until the largest residuum is small enough or $1/4$ of the total number N_c of the samples of the learning set remain. Then, the result of the linear regression is accepted.

The experiments described below follow the procedures described by Paredes and Vidal [15] as truly thorough tests. The tests consist of three kinds of experiments. The first one is a test with a synthetic data set [15] for which Bayes limit is known and one can estimate how close a particular approach allows one to get close to this limit. The second uses real-life data from the UCI Machine Learning repository [16]. The third consist of a more detailed comparison of the results for three selected data sets from [16].

In the experiments, we compare results obtained by the method described here with the results of some standard methods and up-to date Learning Weighted metrics method by Paredes and Vidal [15]. In each set of the tasks, we give a short description of the problem, the source of data, test procedure, results, and a short discussion.

3.1 Synthetic Data

Synthetic data [15] is two dimensional and consists of three two dimensional normal distributions with identical a-priori probabilities. If μ denotes the vector of the means and C_m is the covariance matrix, there is

Class A: $\mu = (2, 0.5)^t$, $C_m = (1, 0; 0, 1)$ (identity matrix)

Class B: $\mu = (0, 2)^t$, $C_m = (1, 0.5; 0.5, 1)$

Class C: $\mu = (0, -1)^t$, $C_m = (1, -0.5; -0.5, 1)$.

In this experiment, we used a simple strategy of using half of the total number of samples of the learning set nearest to the query point. Fig. 1 shows the results obtained by different methods for different learning sets

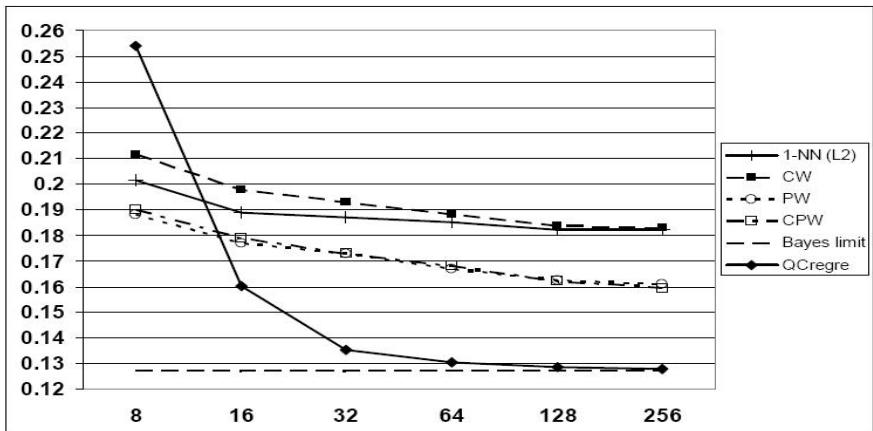


Fig. 1. Comparison of classification errors of the synthetic data for different approaches. In the legend, 1-NN (L2) means the 1-NN method with Euclidean metrics, CW, PW, and CPW are three variants of the method by Paredes and Vidal; points are estimated from the reference [15]. “Bayes” means the Bayes limit. QCregre means the method presented here.

Table 1. Classification error rates for different datasets and different NN-based approaches by [18] and LWM1. Empty cells denote data not available.

Dataset	L2	CDM	CW	PW	CW	QcregreL2
Australian	34.37	18.19	17.37	16.95	16.83	22.72
balance	25.26	35.15	17.98	13.44	17.6	41.32
cancer	4.75	8.76	3.69	3.32	3.53	4.08
diabetes	32.25	32.47	30.23	27.39	27.33	33.54
DNA	23.4	15	4.72	6.49	4.21	46.63
German	33.85	32.15	27.99	28.32	27.29	42.49
glass	27.23	32.9	28.52	26.28	27.48	49.46
heart	42.18	22.55	22.34	18.94	19.82	22.67
ionosphere	19.03					12.87
iris	6.91					5.00
led17	20.5					21.84
letter	4.35	6.3	3.15	4.6	4.2	44.20
liver	37.7	39.32	40.22	36.22	36.95	43.33
monkey1	2.01					10.91
phoneme	18.01					23.03
Satimage	10.6	14.7	11.7	8.8	9.05	28.95
segmen	11.81					15.87
sonar	31.4					40.01
vehicle	35.52	32.11	29.38	29.31	28.09	45.96
vote	8.79	6.97	6.61	5.51	5.26	8.79
vowel	1.52	1.67	1.36	1.68	1.24	16.81
waveform21	24.1	0	0	0	0	52.97
waveform40	31.66	0	0	0	0	58.12
wine	24.14	2.6	1.44	1.35	1.24	8.68

sizes from 8 to 256 samples and testing set of 5000 samples all from the same distributions and mutually independent. Each point was obtained by averaging over 100 different runs.

In our method “QCregre”, we used a simple strategy of using half of the total number of samples of the learning set nearest to the query point in this experiment. For other methods, i.e. 1-NN method with L2 metrics and variants of the LWM method by Paredes and Vidal [15], the values were estimated from the literature cited.

In Fig. 1, it is seen that the use of the class probability estimation with the method presented here in this synthetic experiment outperforms all other methods shown in Fig. 1 and for a large number of samples, it quickly approaches the Bayes limit.

3.2 Data from the Machine Learning Repository

Data sets prepared just for running with a classifier were prepared by Paredes and Vidal and are available on the net [17]. We used all data sets of

this corpus. Each task consists of 50 pairs of training and testing sets corresponding to 50-fold cross validation. For DNA data [16], Letter data (Letter recognition [16]), and Satimage (Statlog Landsat Satellite [16]) the single partition into training and testing set according to the specification in [16] was used. We also added the popular Iris data set [16] with ten-fold cross validation.

The results obtained by the QCregre approach presented here, in comparison with data published in [15], are summarized in Table 1. Each row of the table corresponds to one task from [16]. For tasks where the data is not available from [15], only the results for 1-NN method with L2 metrics were amended.

In the QCregre method, we used a rather complex strategy of robust modification of linear regression as described above. The interesting point is the experiment with the simplest strategy of using half of the samples nearest to the query point. For some tasks we obtained very good results. In Table 2, the results are shown together with the results for other methods published in [16] for tasks “Heart”, “Ionosphere”, and “Iris”. Here, we shortly characterize these data sets as follows:

The task *Heart* indicates the absence or presence of heart disease for a patient.

For the task “Ionosphere”, the targets were free electrons in the ionosphere. “Good” radar returns are those showing evidence of some type of structure in the ionosphere. “Bad” returns are those that do not; their signals pass through the ionosphere.

The task *Iris* is to determine whether an iris flower is of class Versicolor or Virginica. The third class, Setoza is deleted, as it is linearly separable from the other two. 100 samples, four parameters and ten-fold cross validation were used, as in [18].

We do not describe these tasks in detail here as all of them can be found in descriptions of individual tasks of the Repository and also the same approach to testing and evaluation was used. Especially, splitting the data set into two disjoint subsets, the learning set and the testing set and the use of cross validation were the same as in [16] or – for the Iris database as in [18].

We also checked some standard methods for comparison as follows:

- 1-NN – standard nearest neighbor method [19]
- Sqrt-NN – the k -NN method with k equal to the square root of the number of samples of the learning set [10]
- Bay1 – the naïve Bayes method using ten bins histograms [20]
- LWM1 – the learning weighted metrics by Paredes and Vidal [15].

For k -NN, Bayes, LWM and our method the discriminant thresholds θ_g were tuned accordingly. All procedures are deterministic (even Bayes algorithm) and then no repeated runs were needed.

Table 2. Classification errors for three different tasks shown for the different methods presented in the Machine Learning Repository. The note [fri] means the results according to the report by Friedman [18]. The results computed by authors are shown in bold.

Heart		Ionosphere		Iris	
Algorithm	Test	Algorithm	Error	Algorithm	Test
QCregre1	0.178	QCregre1	0.02013	scythe[fri]	0.03
LWM1	0.189	Bay1	0.02013	QCregre1	0.04878
Bayes	0.374	LWM1	0.0265	sqrt-NN	0.04879
Discrim	0.393	IB3 (Aha & Kibler, IJCAI-1989)	0.033	mach:ln [fri]	0.05
LogDisc	0.396	backprop an average of over	0.04	mach-bth [fri]	0.05
Alloc80	0.407	sqrt-NN	0.0537	CART	0.06
QuaDisc	0.422	Ross Quinlan's C4 algorithm	0.06	mach [fri]	0.06
Castle	0.441	nearest neighbor	0.079	mach:ds [fri]	0.06
Cal5	0.444	"non-linear" perceptron	0.08	1-NN	0.0609
Cart	0.452	"linear" perceptron	0.093	LWM1	0.0686
Cascade	0.467			Bay1	0.0854
KNN	0.478			CART	0.11
Smart	0.478			k-NN	0.8
Dipol92	0.507				
Itrule	0.515				
BayTree	0.526				
Default	0.56				
BackProp	0.574				
LVQ	0.6				
IndCart	0.63				
Kohonen	0.693				
Ac2	0.744				
Cn2	0.767				
Radial	0.781				
C4.5	0.781				
NewId	0.844				

4 Conclusion and Discussion

In the first part of this section we show that the approach presented can be useful in some cases. Some notes on the computational complexity and relation of the distribution mapping exponent to the correlation dimension follow:

The main goal of this chapter is to show that the correlation integral can be decomposed into local functions – the probability distribution mapping functions (PDMF). Each PDMF corresponds to a particular point of data space and characterizes the probability distribution in some neighborhood of a given point. In fact, the correlation integral is a distribution function of the bivariate distances of the data set, and PFMF is a distribution function of the distances of the points of the data set from a particular point, the query point x . We have also shown that – similarly as the correlation integral – the PDMF can be approximated by a polynomial function. This polynomial approximation is governed by two constants, the distribution mapping exponent, which can be considered as the local analog to the correlation dimension, and a multiplicative constant. It is proven here that this multiplicative constant is very closely related to the probability density at the given point. The estimation of this constant is used to construct a classifier.

This classifier is slightly related to the nearest neighbor methods. It uses information about distances of the neighbors of different classes from the query point and neglects information about the direction where the particular neighbor lies.

Nearest neighbor methods do not differentiate individual distances of nearest points. E.g. in the k -NN method the number of points of one and the other class among k nearest neighbors is essential, but not the individual distances of points. The method proposed here takes the individual distances into account even if these distances are a little bit hidden in the regression equations. The method outperforms 1-NN, k -NN as well as LWM (learning weighted metrics) by Paredes and Vidal [15] in many cases and can be found as the best one for some tasks.

By use of the notion of distance, i.e. a simple transformation $E_n \rightarrow E_1$, the problems with the curse of dimensionality are easily eliminated at the loss of information on the true distribution of the points in the neighborhood of the query point. The curse of dimensionality [21], [22] means that the computational complexity grows exponentially with dimensionality n , while the complexity only grows linearly here.

The method has no tuning parameters except for those related to linear regression. There is no true learning phase. In the "learning phase" only the standardization constants are computed and thus this phase is several orders of magnitude faster than the learning phase of the neural networks or other various methods.

In the regression equations there are multiplicative constants C_c . We have shown that these constants are proportional to the probabilities $p(c|x)$ that point x is of class c . Thus, C_c allows one to differentiate between the densities of the classes at point x and the distribution mapping exponent q has no use in this task. One can deduce that neither the correlation dimension nor the distribution mapping exponent govern the probability that point x is of a class c . Their role in the probability density estimation and classification is indirect via polynomial transformation only.

There is an interesting relationship between the correlation dimension and the distribution mapping exponent q_c . The former is a global feature of the fractal or data generating process; the latter is a local feature of the data set and is closely related to the particular query point. On the other hand, if linear regression were used, the computational procedure is almost the same in both cases. Moreover, it can be found that values of the distribution mapping exponent usually lie in a narrow interval $<-10, +10>$ percentage around the mean value.

The question arises what is the relation of the distribution mapping exponent statistics to the overall accuracy of the classification.

Acknowledgements

This work was supported by the Ministry of Education of the Czech Republic under the project Center of Applied Cybernetics No. 1M0567, and No. MSM6840770012 Transdisciplinary Research in the Field of Biomedical Engineering II.

References

1. Grassberger, P., Procaccia, I.: Measuring the strangeness of strange attractors. *Physica* 9D, 189–208 (1983)
2. Osborne, A.R., Provenzale, A.: Finite correlation dimension for stochastic systems with power-law spectra. *Physica D* 35, 357–381 (1989)
3. Lev, N.: Hausdorff dimension. Student Seminar, Tel-Aviv University (2006), www.math.tau.ac.il/~levnir/files/hausdorff.pdf
4. Weisstein, E.W.: Information Dimension. From MathWorld—A Wolfram Web Resource (2007), <http://mathworld.wolfram.com/InformationDimension.html>
5. Merz, C.J., Murphy, P.M., Aha, D.W.: UCI Repository of Machine Learning Databases. Dept. of Information and Computer Science, Univ. of California, Irvine (1997), <http://www.ics.uci.edu/~mlearn/MLSummary.html>
6. Camstra, P., Vinciarelli, A.: Intrinsic Dimension Estimation of Data: An Approach based on Grassberger-Procaccia's Algorithm. *Neural Processing Letters* 14(1), 27–34 (2001)
7. Takens, F.: On the Numerical Determination of the Dimension of the Attractor. In: *Dynamical Systems and Bifurcations*. Lecture Notes in Mathematics, vol. 1125, pp. 99–106. Springer, Berlin (1985)
8. Camstra, F.: Data dimensionality estimation methods: a survey. *Pattern Recognition* 6, 2945–2954 (2003)
9. Guerrero, A., Smith, L.A.: Towards coherent estimation of correlation dimension. *Physics letters A* 318, 373–379 (2003)
10. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification, 2nd edn. John Wiley and Sons, Inc., New York (2000)
11. Dvorak, I., Klaschka, J.: Modification of the Grassberger-Procaccia algorithm for estimating the correlation exponent of chaotic systems with high embedding dimension. *Physics Letters A* 145(5), 225–231 (1990)

12. Wikipedia - Gauss-Markov theorem,
http://en.wikipedia.org/wiki/Gauss-Markov_theorem
13. Wikipedia – Linear regression, http://en.wikipedia.org/wiki/Linear_regression
14. Leamer, E.E.: Specification searches. Ad hoc inference with non-experimental data. John Wiley and Sons, New York (1978)
15. Paredes, R., Vidal, E.: Learning Weighted Metrics to Minimize Nearest Neighbor Classification Error. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(7), 1100–1110 (2006)
16. Merz, C.J., Murphy, P.M., Aha, D.W.: UCI Repository of Machine Learning Databases. Dept. of Information and Computer Science, Univ. of California, Irvine (1997), <http://www.ics.uci.edu/~mlearn/MLrepository.html>
17. Paredes, R.: <http://www.dsic.upv.es/~rparedes/research/CPW/index.html>
18. Friedmann, J.H.: Flexible Metric Nearest Neighbor Classification. Technical Report, Dept. of Statistics, Stanford University (1994)
19. Cover, T.M., Hart, P.E.: Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory IT-13(1), 21–27 (1967)
20. Gama, J.: Iterative Bayes. Theoretical Computer Science 292, 417–430 (2003)
21. Bellman, R.E.: Adaptive Control Processes. Princeton University Press, Princeton (1961)
22. Pestov, V.: On the geometry of similarity search: Dimensionality course and concentration of measure. Information Processing Letters 73(1-2), 47–51 (2000)

Investigating Neighborhood Graphs for Inducing Density Based Clusters

Viviani Akemi Kasahara¹ and Maria do Carmo Nicoletti²

¹ Computer Science Department - University Federal de S. Carlos - SP - Brazil
`vivi_kasahara@comp.ufscar.br`

² Computer Science Department - University Federal de S. Carlos - SP - Brazil
`carmo@dc.ufscar.br`

Summary. Graph based clustering algorithms seem to be capable of detecting clusters of various shapes. The basic algorithm is based on the concept of minimum spanning tree (MST). A possible extension of the MST approach is based on the concept of neighborhood region defined between each pair of points, used for constructing the neighborhood graph. This chapter investigates the impact of seven different ways of defining a neighborhood region between two points, when identifying clusters in a neighborhood graph, particularly focusing on density based clusters. On the one hand results show that the neighborhood definitions that do not employ parameters are not suitable for inducing density based clusters. On the other hand they also show that although essential for successfully separating density based clusters, the parameters employed by some of the definitions need to have their values tuned by the user. As will be discussed, two of the four neighborhood based graphs that use parameters, namely the Elliptic Gabriel graph and the β -Skeleton based graph, however, are not suitable for inducing density based clusters. A parameterized version of Sphere-of-influence based graphs more suitable for inducing density based clusters is proposed.

1 Introduction

Cluster analysis is the organization of a collection of patterns (usually represented as a vector of measurements, or a point in a multidimensional space) into clusters based on similarity [1]. Clustering can be useful in a variety of knowledge domains such as data analysis, computer vision, image segmentation, document retrieval, marketing, geographic information systems and classification. As commented in [1], in many problems there is no much information about the data, and the decision-maker must make as few assumptions about the data as possible. It is under these restrictions that clustering is particularly appropriate for exploring the interrelationships among the data points searching for their plausible structure. Research work

in the area of cluster analysis focuses on both, understanding the nature of clusters and creating algorithms for inducing clusters from data. Many different clustering algorithms based on a broad variety of mathematical and statistical formalisms can be found in a variety of textbooks [2] [3] [4] [5] [6] [7] [8]. Depending on their main characteristics and their own way of approaching a clustering problem, clustering algorithms can be grouped into many different categories. In the taxonomy proposed in [8] one of the categories, identified as agglomerative, groups algorithms that produce a sequence of clustering, each of them with a smaller number of groups than the previous one. Algorithms categorized as agglomerative can still be further categorized into two sub-categories: those based on the matrix theory and those based on the graph theory.

Graph based algorithms seem to be capable of detecting clusters of various shapes, at least for the case in which they are well separated. The basic graph based clustering algorithm is based on the concept of minimum spanning tree (MST) [9] [10] and was inspired by the way human perception seems to work [11]. Initially a complete graph is constructed having the given points (to be clustered) as nodes and then, a minimum spanning tree is extracted from the complete graph (e.g. using the algorithms proposed by Kruskal [9] or Prim [10]). Based on a user-defined criterion of inconsistency, the algorithm identifies the inconsistent edges of the MST and, by removing them, induces connected components, each identified as a cluster.

A possible extension of the MST approach is based on the concept of neighborhood region defined between each pair of points, used for constructing what is called a neighborhood graph. The task of the clustering algorithm is to identify the connected components of the neighborhood graph (i.e., the clusters).

As mentioned in [12], proximity drawings arise in many knowledge areas, such as pattern recognition, geographic variation analysis, computational geometry, computational morphology and computer vision, as descriptors of the shape or skeleton of a set of points. This chapter empirically investigates the role played by seven different definitions of neighborhood region in inducing the neighborhood graphs, aiming at identifying their influence and role in detecting density based clusters.

Although a few graph based clustering algorithms can be found in the literature [13] [14] [16], they are not as popular as many other cluster algorithms and they have not been used in many applications; this is particularly the case of graph based algorithms that use the concept of neighborhood region. This fact was the main motivation for the investigation into the many different ways of defining neighborhood regions described in this chapter, in an attempt to analyze their adequacy for inducing clusters based on the density of points.

In recent years however, the use of intelligent computational techniques and, among them, clustering techniques, to problems in biosciences has increased and so have increased the use of graph based clustering algorithms in

bioscience related problems. Graph based algorithms particularly have been successfully used for analyzing DNA microarray data [13] [14] [15] [16]. The clustering algorithm based on graph connectivity known as the Highly Connected Subgraphs (HCS) proposed in [13] was motivated by gene expression in molecular biology. The algorithm known as CLICK (Cluster Identification via Connectivity Kernels) [14] is based on the HCS and is applicable to gene expression analysis as well as to other biological applications. CLICK does not make assumptions on the number or structure of the clusters and can be characterized as an algorithm that recursively partitions a weighted graph into components employing the concept of minimum cut.

The remainder of this chapter is organized as follows. Section 2 presents the main idea of the basic MST algorithm as well as some examples, as a motivation for its generalization using the notion of neighborhood region. Section 3 introduces the notation, the basic concepts, and the graph based algorithm for inducing clusters used in the experiments; it also presents and discusses seven different concepts of neighborhood region found in the literature. Section 4 describes the clustering results obtained using the seven different approaches in different data configuration and analyses their impact on the identification of density based clusters. Finally in Sect. 5 the main conclusions of this work are summarized and the next steps of this research work are highlighted.

2 A MST Clustering Algorithm Based on the Identification of Inconsistent Edges

Algorithm 1 describes the necessary steps for inducing clusters from a set of points V and it is loosely based on the algorithm described in [8]. The basic idea is very simple and consists in inducing the minimum spanning tree (MST) having V as nodes (and $|V| - 1$ edges) and then remove from the MST the edges that are considered unusually large compared with their neighboring edges. These edges are identified as inconsistent and presumably they connect points from different clusters. The final result is a forest where each tree represents a cluster.

Consider a set of points V modeled as the vertices of a complete weighted graph $G = (V, E)$, where the weight of an edge $e_i \in E$, $e_i = (p_i, p_j)$, $p_i, p_j \in V$, is the Euclidean distance $d(p_i, p_j)$. For identifying and removing the inconsistent edges two user-defined parameters, k and q , are needed. The parameter k is used for identifying, for each edge $e \in E$, all the edges $e_i \in E$ that lie (at the most) k steps away from e . The mean m_e and the standard deviation σ_e of the weights of these edges are calculated. If $\text{weight}(e)$ lies more than q standard deviations (σ_e) away from m_e , the edge e is considered inconsistent as expressed by Eq. 1 where w_e means $\text{weight}(e)$.

$$w_e - m_e / \sigma_e > q \quad (1)$$

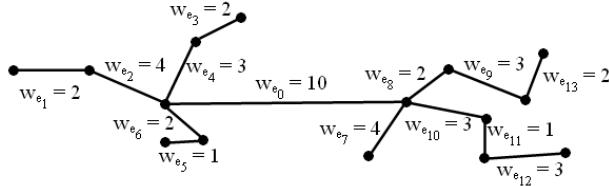


Fig. 1. MST based algorithm ($k = 2$, $q = 3$) - inconsistent edge to be removed: e_0

```

procedure clustering_MST(V,E,k,q)
{ALGORITHM 1
input: set of points V =  $p_1, p_2, \dots, p_N$ , |V| = N.
output: a forest defined by (V,  $E_{new}$ )}
begin
  create_complete_graph(V,E)
  MST  $\leftarrow$  MST_Prim(V,E) {or MST_Kruskal(V,E)}
  inconsistent_edges  $\leftarrow$   $\emptyset$ 
  for i  $\leftarrow$  1 to N do
    begin
      CE  $\leftarrow$   $\emptyset$ 
      for j  $\leftarrow$  i to N do
        if steps( $e_i, e_j$ )  $\leq k$  do CE  $\leftarrow$  CE  $\cup$  { $e_j$ }
         $m_{e_i} \leftarrow$  mean_value(CE)
         $\sigma_{e_i} \leftarrow$  standard_deviation(CE,  $m_{e_i}$ )
        if  $(w_{e_i} - m_{e_i})/\sigma_{e_i} > q$  then
          inconsistent_edges  $\leftarrow$  inconsistent_edges  $\cup$  { $e_i$ }
    end
     $E_{new} \leftarrow E -$  inconsistent_edges
end

```

Figure 1 illustrates the process. For $k = 2$ and $q = 3$, the edges that are 2 steps away from e_0 are e_i , $i = 1, \dots, 11$, giving $m_{e_0} = 2.45$ and $\sigma_{e_0} = 1.03$. So, e_0 is at 7.33 standard deviations away from m_{e_0} and consequently, e_0 is inconsistent since $7.33 > q$.

The many different ways of characterizing inconsistent edges (other than the mean and standard deviation) give rise to several possible variations of Algorithm 1. Other measurements can be used with a slight modification in the structure of Algorithm 1. Algorithm 1 is particularly successful when the clusters are well separated; its efficiency, however, is related to its sensitivity to the position of a point as well as to the limitations of the tree structure. Reference [17] presents a clustering algorithm that uses a structure called scale-free [18] minimum spanning tree in an attempt to overcome some of the problems associated with the MST algorithm and reference [19] proposes two clustering algorithms based on minimum and maximum spanning trees. Figure 2 shows five clusters induced by Algorithm 1, as expected.

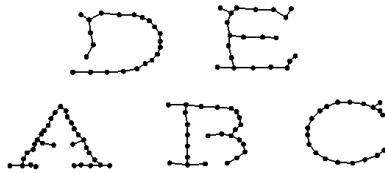


Fig. 2. Graph induced by Algorithm 1 for $k = 2$ and $q = 2$

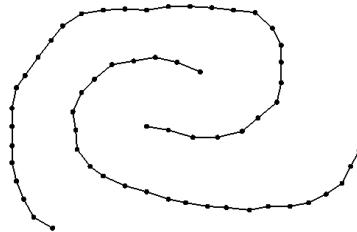


Fig. 3. Graph induced by Algorithm 1 for $k = 2$ and $q = 2$

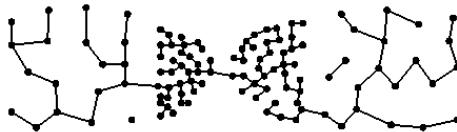


Fig. 4. Graph induced by Algorithm 1 for $k = 5$ and $q = 2$

Points in Fig. 3 have similar characteristics to those in Fig. 2 and this fact confirms the successful performance of the MST algorithm with the removal of inconsistent edges.

However, as can be seen in Fig. 4, Algorithm 1 did not succeed in inducing the density based clusters that can be visually identifiable in the set of 114 points shown in the figure. The data can be characterized as two regions of high density and two regions of low density or then as one region of high density and two regions of low density points. A similar set of points will be considered later in this chapter (Sect. 4).

3 Inducing Neighborhood Graphs - Basic Concepts

As mentioned in Sect. 2, Algorithm 1 can be extended by exploring the concept of neighborhood region (region of influence). In the new approach the concept of minimum spanning tree is no longer used. The algorithm that uses the concept of neighborhood region for determining if there is (or there is not) an edge between any two points induces a unique neighborhood graph. Its main step consists in finding all pair of points whose neighborhood region

satisfies a given property P. SubSection 3.1 presents the general definition of neighborhood graph and details the general pseudocode (Algorithm 2) for inducing neighborhood graphs. In SubSect. 3.2 the seven different ways of defining the neighborhood region between two points are reviewed and detailed.

3.1 The Basic Algorithm for Inducing Neighborhood Graphs

Let V be a set of points. The neighborhood graph (NG) of V is a graph whose vertices are the points of V and an edge between two nodes indicates a neighborhood relation between them. Generally neighborhood relations are dependent upon the concept of neighborhood region [20] [21] [22]. Informally described, given a set of points V , the construction of a NG having V as the set of vertices is carried out by adding edges between two vertices p_i and p_j when no other vertices of V are found inside the neighborhood region defined by the unordered pair (p_i, p_j) .

Generally the definition of a neighborhood region is carried out by establishing a mathematical equation that should be satisfied by a pair of vertices. In many cases the neighborhood region determined by a pair of vertices represents a geometric figure.

Definition 1. (*Neighborhood graph in a bi-dimensional space*) Let V be a set of points in R^2 . Each unordered pair of points¹ $(p_i, p_j) \in V \times V$, $p_i \neq p_j$ is associated with a neighborhood region $U_{p_i, p_j} \subseteq R^2$. Let P be a property defined on $U = \{U_{p_i, p_j} \mid (p_i, p_j) \in V \times V\}$. A neighborhood graph $G_{U, P}(V, E)$ defined by property P is a graph with the set of vertices V whose set of edges E is such that $(p_i, p_j) \in E$ if and only if the neighborhood region U_{p_i, p_j} has property P .

Depending on the definition of the neighborhood region as well as on the property P , different neighborhood graphs can be constructed. The algorithm described in the following pseudocode is general; its main tasks are the determination of the neighborhood region induced by any two vertices of G and, in sequence, verifying if the region satisfies a given property P .

In the following pseudocode (loosely based on the version described in [8]) the procedure `neighborhood_graph` is used for inducing the set of edges of an initially null graph, based on the neighborhood region induced by any two vertices as well as on a given property P and procedure `find_clusters` is used for determining the connected components in the previously constructed graph.

In procedure `neighborhood_graph(V, E)`, depending on the way the function `neighborhood_region(p_i, p_j)` is defined, i.e., the mathematical formalism it is based upon as well as the adopted P property, an edge connecting p_i to p_j may (or may not) be included in the graph - this obviously has a strong

¹ In case of Sphere-of-influence based graph the neighborhood region is defined by only one point.

```

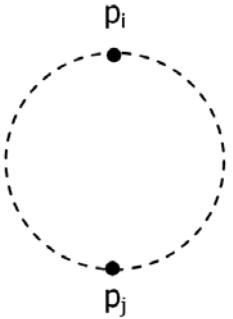
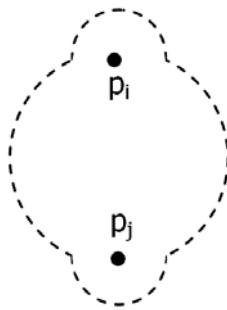
procedure neighborhood_graph(V,E)
{ALGORITHM 2
input: set of points V = { $p_1, p_2, \dots, p_N$ }. |V| = N.
output: the set of edges E of an initially
null graph ( $E = \emptyset$ ) G=(V,E)}
begin
    k  $\leftarrow$  1 {index controlling the edges to be included}
    E  $\leftarrow$   $\emptyset$ 
    for i  $\leftarrow$  1 to N do
        for j  $\leftarrow$  i + 1 to N do
            begin
                 $U_{p_i,p_j} \leftarrow$  neighborhood_region( $p_i, p_j$ )
                {property P of neighborhood graph definition}
                if  $U_{p_i,p_j} \cap (V - \{p_i, p_j\}) = \emptyset$  then
                    begin
                         $e_k \leftarrow (p_i, p_j)$ 
                        E  $\leftarrow$  E  $\cup$  { $e_k$ }
                        k  $\leftarrow$  k + 1
                    end
            end
        end
    end.
procedure find_clusters(V,E,C)
begin
    neighborhood_graph(V,E)
    identify_connected_components(V,E,C)
end.
```

influence on the number of connected components the final graph will have i.e., on the number of clusters the algorithm will induce. In the pseudocode the property P is given by condition $U_{p_i,p_j} \cap (V - \{p_i, p_j\}) = \emptyset$, but could also be given by the condition $U_{p_i} \cap U_{p_j} \neq \emptyset$ when using the Sphere-of-influence concept for inducing graphs, discussed in Subsect. 3.2. The procedure $\text{find_clusters}(V, E, C)$ can be implemented using any algorithm that finds connected components in graphs (see [23] for instance).

3.2 The Many Different Ways of Defining the Neighborhood Region

This section focuses on seven different ways of constructing neighborhood regions, identified as: Gabriel [20] and Modified Gabriel [22], Relative Neighborhood and Modified Neighborhood [22], Elliptic Gabriel [24], the β -Skeleton [25] parameterized family and the Sphere-of-influence [26] [27].

For analyzing the influence of each of the seven definitions of neighborhood region in inducing density based clusters, the property P as defined in the pseudocode described in Subsect. 3.1 will be considered.

**Fig. 5.** Gabriel Neighborhood region**Fig. 6.** Modified Gabriel Neighborhood region

In what follows, $\delta(p_i, p_j)$ means the distance between two points p_i and p_j and $B(p_i, r)$ means an open circle centered in p_i with radius r , i.e., $B(p_i, r) = \{p_j \in V \mid \delta(p_i, p_j) < r\}$.

The well-known Gabriel graph $G=(V,E)$ (see Fig. 5) can be characterized as a neighborhood graph where the neighborhood region U_{p_i,p_j} induced by two points $p_i, p_j \in V$ is the circle with diameter defined by the line segment (p_i, p_j) , as in Eq. 2.

$$U_{p_i,p_j} = B((p_i + p_j)/2, \delta(p_i, p_j)/2) \quad (2)$$

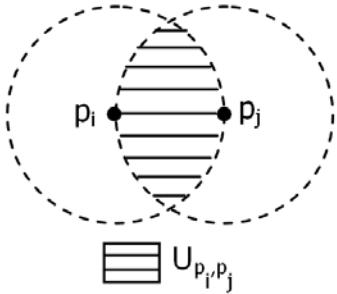
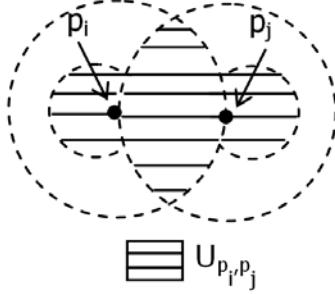
With the introduction of a user defined parameter σ , named the relative edge consistency, the basic circle defined by Eq. 2 can be modified as described by Eq. 3 [22] and the corresponding neighborhood graph is known as Modified Gabriel Neighborhood region (see Fig. 6).

$$U_{p_i,p_j}(\sigma) = B((p_i + p_j)/2, \delta(p_i, p_j)/2) \cup \{x \mid \sigma \times \min\{\delta(p_i, x), \delta(p_j, x)\} < \delta(p_i, p_j)\} \quad (3)$$

As proposed in [22], the Relative Neighborhood graph is defined having a lune as the neighborhood region induced by two points $p_i, p_j \in V$, formally described by Eq. 4 (see Fig. 7). Similarly to the Modified Gabriel Neighborhood region, the Relative Neighborhood region has a modified version which introduces the parameter σ , as described by Eq. 5, responsible for enlarging the neighborhood region as shown in Fig. 8.

$$U_{p_i,p_j} = B(p_i, \delta(p_i, p_j)) \cap B(p_j, \delta(p_i, p_j)) \quad (4)$$

$$U_{p_i,p_j}(\sigma) = B((p_i + p_j)/2, \delta(p_i, p_j)/2) \cup \{x \mid \sigma \times \min\{\delta(p_i, x), \delta(p_j, x)\} < \delta(p_i, p_j)\} \quad (5)$$

**Fig. 7.** Relative Neighborhood region**Fig. 8.** Modified Relative Neighborhood region

The Elliptic Gabriel neighborhood graphs (EGG) as proposed in [24] are a parameterized family of graphs, based on a parameterized elliptic neighborhood region definition. The parameter α gives the elongation shape of the ellipses along the y-axis as its value increases. Depending on the value of α , three situations may arise.

- If $\alpha = 1$ the resulting geometric figure is the circle,
- If $\alpha < 1$ the resulting geometric figure is an ellipsis with its x-axis larger than its y-axis and
- If $\alpha > 1$ the resulting geometric figure is an ellipsis with its y-axis larger than its x-axis.

Without losing generality, consider two points, p_i and $p_j \in V$ given by their coordinates $p_i = (x_{p_i}, y_{p_i})$ and $p_j = (x_{p_j}, y_{p_j})$ and that $x_{p_j} > x_{p_i}$ and $y_{p_j} > y_{p_i}$ hold. Let the center (x_c, y_c) of the line segment that joins the two points be: $x_c = (x_{p_j} - x_{p_i})/2 + x_{p_i}$ and $y_c = (y_{p_j} - y_{p_i})/2 + y_{p_i}$. The three possible formulas defining three α -based elliptical neighborhood regions are described by Eqs. 6, 7 and 8

$$\alpha = 1 \quad U_{p_i, p_j}(\alpha) = \{ \langle x, y \rangle \mid (x - x_c)^2 + (y - y_c)^2 < (\delta(p_i, p_j)/2)^2 \} \quad (6)$$

$$\alpha < 1 \quad U_{p_i, p_j}(\alpha) = \{ \langle x, y \rangle \mid (x - x_c)^2 + ((y - y_c)/\alpha)^2 < (\delta(p_i, p_j)/2)^2 \} \quad (7)$$

$$\alpha > 1 \quad U_{p_i, p_j}(\alpha) = \{ \langle x, y \rangle \mid ((x - x_c)/\alpha)^2 + (y - y_c)^2 < (\delta(p_i, p_j)/2)^2 \} \quad (8)$$

Figure 9 shows a general diagram for the parameterized family of elliptic neighborhood region highlighting the influence of the parameter α in shaping the neighborhood region. As suggested by the authors, the EGG neighborhood region can be used for curvature analysis, simplification, smoothing and surface reconstruction.

The β -Skeleton neighborhood family of graphs defined in [25] can be approached as a general case which encompasses some of the previously defined graphs. Three cases arise, depending on the value of parameter β :

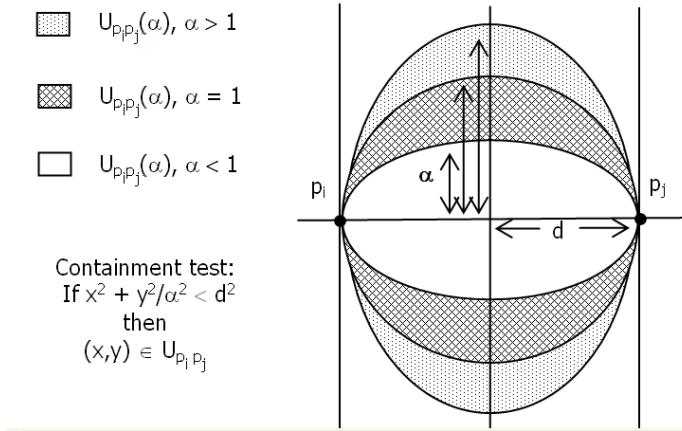


Fig. 9. Parameterized family of Elliptic Neighborhood region

- If $\beta = 1$ the resulting geometric figure is a circle,
- If $0 < \beta < 1$ the resulting geometric figure is the intersection of two circles and
- If $1 < \beta < \infty$ the resulting geometric figure is the union of two circles.

Accordingly, the β -Skeleton neighborhood regions are defined by Eqs. 9, 10 and 11 respectively.

$$\beta = 1 \quad U_{p_i, p_j}(\beta) = B((1 - \beta/2)p_i + \beta/2p_j, \beta/2 \times \delta(p_i, p_j)) \cap B((1 - \beta/2)p_j + \beta/2p_i, \beta/2 \times \delta(p_i, p_j)) \quad (9)$$

$$0 < \beta < 1 \quad U_{p_i, p_j}(\beta) = B((1 - \beta/2)p_i + \beta/2p_j, \delta(p_i, p_j))/(2\beta) \cap B((1 - \beta/2)p_j + \beta/2p_i, \delta(p_i, p_j))/(2\beta) \quad (10)$$

$$1 < \beta < \infty \quad U_{p_i, p_j}(\beta) = B((1 - \beta/2)p_i + \beta/2p_j, \beta/2 \times \delta(p_i, p_j)) \cap B((1 - \beta/2)p_j + \beta/2p_i, \beta/2 \times \delta(p_i, p_j)) \quad (11)$$

Figure 10 shows the general diagram of the parameterized β -Skeleton neighborhood family exhibiting the influence of the value of the parameter β in the shape of the neighborhood region.

As mentioned in [27] β -Skeleton based graphs can be applied in many knowledge domains. Reference [28] uses β -Skeleton based graphs for boundary curve reconstruction from point samples and [29] describes the use of β -Skeleton for reducing the size of the training set for Support Vector Machine (SVM).

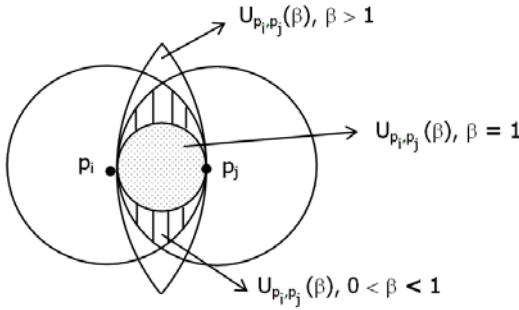


Fig. 10. Parameterized family of β -Skeleton Neighborhood region



Fig. 11. A Sphere-of-influence based graph

An interesting definition of neighborhood region that can be useful for density based clustering is known as Sphere-of-influence region, proposed in [26]. Consider V the set of points of a graph $G=(V, E)$ and $p_i \in V$ be any point. Let $p_j \in V$ be the closest point to p_i and let $r = \delta(p_i, p_j)$. The Sphere-of-influence region U_{p_i} is the circle that has p_i as its center and radius r , formally defined as Eq. 12.

$$U_{p_i} = B(p_i, \delta(p_i, p_j)) \quad s.t. \quad \delta(p_i, p_j) = \min\{(p_i, p_k), p_k \in V, k \neq i\} \quad (12)$$

Let U_{p_i} be the Sphere-of-influence region defined by point p_i and U_{p_j} be the Sphere-of-influence region defined by point p_j . The property P used for inducing neighborhood graphs can be rewritten as: $(p_i, p_j) \in E$ if and only if $U_{p_i} \cap U_{p_j} \neq \emptyset$. Figure 11 shows a general diagram of a Sphere-of-influence based graph.

Although this chapter focuses on the Sphere-of-influence graph (using Eq. 12), this algorithm has several extensions, as can be seen in [30].

4 Experiments and Results

The goal of the experiments described in this section was to determine how well each of the seven neighborhood regions, used in conjunction with the

pseudocodes previously defined would induce density based clusters. The influence of the neighborhood regions previously described was analyzed considering four different situations:

1. sets of well-separated points each having a different density: low, medium and high,
2. sets of points, with different densities, one nested inside the other and
3. regions of points with similar sub-region density patterns, placed as mirror-images of each other in relation to an axis.
4. more elaborated sets of points obtained with random generation of Gaussian samples in a 2-dimensional space.

The Modified Gabriel graph and the Modified Relative Neighborhood graph work well in cases 1, 2 and 3 due to the use of the parameter σ (whose value is empirically determined), that expands the neighborhood region defined by two points p_i and p_j , adding to the region two extra sets of points, one around p_i and the other around p_j .

Consider a graph that has a high density area (H) of points close to a low density area (L) of points. When a neighborhood region U_{p_i, p_j} is defined by two points p_i and p_j such that $p_i \in H$ and $p_j \in L$, there is a great chance that another point(s) of H is in U_{p_i, p_j} (due to the fact that H is a high density area). Considering that $(p_i, p_j) \in E$ if and only if $U_{p_i, p_j} \cap E = \emptyset$ still holds, no edge between p_i and p_j will be induced. This situation will happen along with all the points of H and L that share a mutual frontier.

For comparison purposes, Fig. 12 shows the Neighborhood Gabriel graph induced using points that belong to three different density areas. Figure 13 shows the Modified Neighborhood Gabriel graph, for $\sigma = 2$, which is a disconnected graph with three connected components i.e., three clusters.

It seems that for graphs that have distinctive density regions sharing a common frontier, the Modified Neighborhood Gabriel graph (or the Modified Relative Neighborhood graph) can be a good choice as a clustering algorithm, provided a convenient value for σ be chosen. Similar situation happens when sets of points with high density are nested within a low density region (or vice-versa), as shown in Figs. 14 and 15 and in Figs. 16 and 17, respectively.

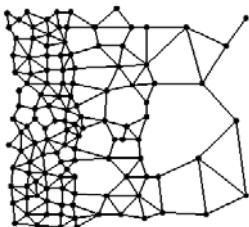


Fig. 12. Neighborhood Gabriel graph

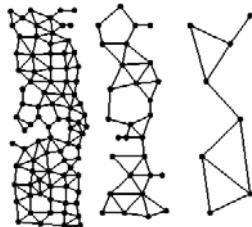


Fig. 13. Modified Neighborhood Gabriel graph ($\sigma = 2$)

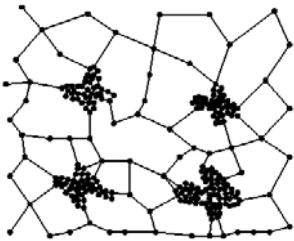


Fig. 14. Relative Neighborhood graph

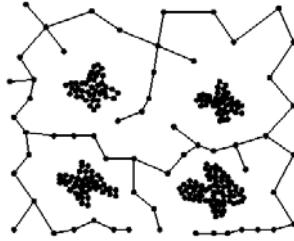


Fig. 15. Modified Relative Neighborhood graph ($\sigma = 2$)

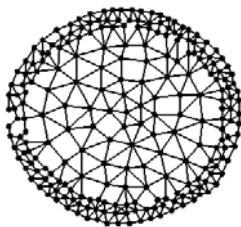


Fig. 16. Neighborhood Gabriel graph

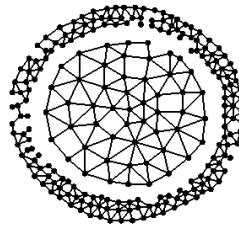


Fig. 17. Modified Neighborhood Gabriel graph ($\sigma = 2$)

In Fig. 14 the Relative Neighborhood graph is shown for comparison purposes. Figure 15 shows the Modified Relative Neighborhood graph obtained using parameter $\sigma = 2$, evidencing five clusters (four dense and one sparse) as its connected components.

In Figs. 16 and 17 the low density region is nested inside the high density region. Figure 16 shows the Neighborhood Gabriel graph for comparison purposes. As expected, the Modified Gabriel graph using parameter $\sigma = 2$, separated both regions into two connected components as shows Fig. 17 evidencing two clusters.

In case 3, where the sets of points with different densities are density based symmetrically opposed to each other in relation to an axis, the clusters are well separated by the Modified Gabriel graph or Modified Relative Neighborhood graph.

Figure 18 shows the Neighborhood Gabriel graph for comparison purposes – it is a connected graph with only one connected component. Figure 19 shows the Modified Neighborhood Gabriel graph obtained using parameter $\sigma = 2$; the graph has four clusters, since the high density areas are close to each other. However if the two highly dense areas were even closer, the final induced graph would have only one central high density cluster instead of two.

Let $\text{boundary}(R)$ be the set of points that belong to the boundary of area R . Consider a graph that has a high density area (H_1) of points close to another high density area (H_2) of points and that $H_1 \cap H_2 = \emptyset$. If p_i and



Fig. 18. Neighborhood Gabriel graph

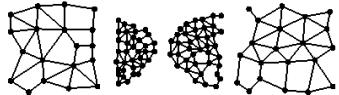


Fig. 19. Modified Neighborhood Gabriel graph ($\sigma = 2$)



Fig. 20. Relative Neighborhood graph

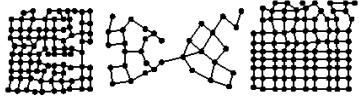


Fig. 21. Modified Relative Neighborhood graph ($\sigma = 2$)

p_j are points such that $p_i \in \text{boundary}(H_1)$ and $p_j \in \text{boundary}(H_2)$, there is a great chance that another point(s) of H_1 (or H_2) will be in U_{p_i, p_j} and consequently, $(p_i, p_j) \notin E$.

It should be noted, however, that this depends on the distance between the points belonging to $\text{boundary}(H_1)$ and $\text{boundary}(H_2)$. If there are points in both boundaries that are close enough, eventually they will define an edge. If the low density areas are close to each other the previous situation does not occur. Consider, for instance, the set of points in Figs. 20 and 21. Using the algorithm described in Sect. 3, only three clusters are constructed. Since the areas have low density, the possibility of finding points that are inside the neighborhood region defined by two points is low.

Figure 20 shows the Relative Neighborhood graph for comparison purposes. Figure 21 shows the Modified Relative Neighborhood graph obtained using parameter $\sigma = 2$, evidencing three clusters as its connected components.

A few neighborhood regions (used in conjunction with the basic algorithm for inducing neighborhood graphs) do not work well when identifying clusters with different densities. Using the Elliptic Gabriel graph or the β -Skeleton based graph the clusters can be separated by modifying the value of a parameter (parameter α for the Elliptic Gabriel graph and β for β -Skeleton based graph).

The shape of the enlarged region (as a consequence of modifying the corresponding parameter) is not suitable for separating clusters using density as a criterion. When considering the Modified Gabriel graph and the Modified Relative Neighborhood graph, however, the parameter σ allows a little enlargement of the neighborhood region, around the points that define the region. In the case of the Elliptic Gabriel graph or β -Skeleton based graph the corresponding parameter provokes an enlargement of the whole neighborhood region, which will prevent the split of regions with different densities. This can be observed in Fig. 22, using the Elliptic Gabriel graph with $\alpha = 0.5$

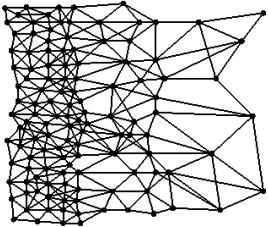


Fig. 22. Elliptic Gabriel graph ($\alpha = 0.5$)

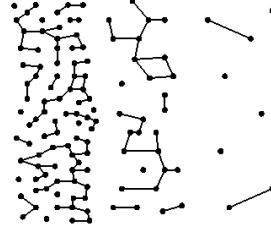


Fig. 23. Elliptic Gabriel graph ($\alpha = 3$)

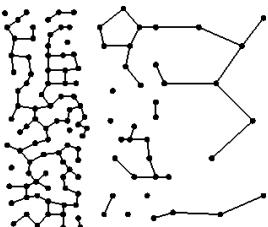


Fig. 24. β -Skeleton based graph ($\beta = 0.5$)

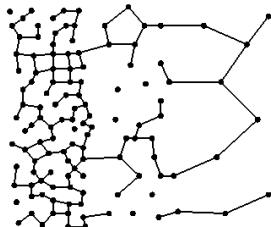


Fig. 25. β -Skeleton based graph ($\beta = 3$)

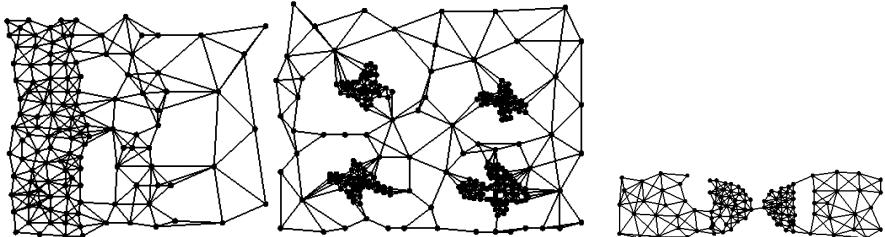


Fig. 26. Sphere-of-influence based graph **Fig. 27.** Sphere-of-influence based graph **Fig. 28.** Sphere-of-influence based graph

and in Fig. 23, using the Elliptic Gabriel graph with $\alpha = 3.0$. This can also be observed in Fig. 24, using the β -Skeleton with $\beta = 0.5$ and in Fig. 25 with $\beta = 3.0$. Note that the set of points shown in Figs. 22 and 23 and in Figs. 24 and 25 is the same as the one shown in Figs. 12 and 13.

Taking into consideration the three first cases listed at the beginning of this section, the Sphere-of-influence region does not work well for sets of points conforming to cases 1, 2 and 3 as shows Figs. 26 and 27 and 28. The resulting graph has only one connected component and consequently, the algorithm has not induced the potential density based clusters embedded in the graph.

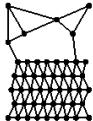


Fig. 29. Sphere-of-influence based graph

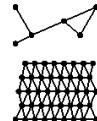


Fig. 30. Modified Neighborhood Gabriel graph ($\sigma = 2$)

The Sphere-of-influence region is more sensitive than the Modified Gabriel region (and the Modified Relative Neighborhood region) when focusing on the distance between the areas with different densities, as can be observed in Figs. 29 and 30. As both density based regions in the figure are relatively close to each other, the Sphere-of-influence region based criteria is unable to detect the two regions.

So, when using the Sphere-of-influence region, the LD (low density) area needs to be at a convenient distance from the HD (high density) region. The ‘convenient distance’ is determinated by the points that belong to the shared frontier of both regions. We suggest that this distance should be such that the spheres defined by the points of the LD region and HD region that share a common frontier, should not intersect each other. This can be accomplished by establishing a minimum distance they should be allowed to be from each other, as described next.

Let P_{LD} and P_{HD} be the sets of points that share a common frontier between the LD and HD regions, respectively. Let $U_{Max_P_{LD}} = \max\{U_{p_i} \mid p_i \in P_{LD}\}$ and $U_{Max_P_{HD}} = \max\{U_{p_i} \mid p_i \in P_{HD}\}$. Accordingly to this convention, the distance between both regions should be greater or equal than $\text{radius}(U_{Max_P_{LD}}) + \text{radius}(U_{Max_P_{HD}})$, where $\text{radius}(U_{p_i})$ gives the radius of the spherical region U_{p_i} .

The only difference between Figs. 29, 30 and 31 is the distance that separates the LD from the HD regions. Figure 31 shows the Sphere-of-influence based graph with the two expected connected components as a result of the strategy based on distance previously discussed.

Let P_{LD} and P_{HD} be the sets of points that share a common frontier between the LD and HD regions, respectively. Let $U_{P_{LD}} = \{U_{p_i} \mid p_i \in P_{LD}\}$ and $U_{P_{HD}} = \{U_{p_i} \mid p_i \in P_{HD}\}$. Using the previous definition of Neighborhood Sphere region, if the closest point to a point that belongs to P_{LD} is also in P_{LD} then the tendency is that the sphere

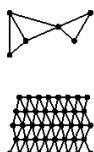


Fig. 31. Sphere-of-influence based graph

regions determined by points in P_{LD} are larger than those determined by points in P_{HD} . When this occurs the points that define the sphere regions in $U_{P_{LD}}$ and $U_{P_{HD}}$ will define inter-region edges, since each element in $U_{P_{LD}}$ will be large enough to intersect an element from $U_{P_{HD}}$. To prevent this, we propose the use of a parameter θ aiming at refining the P property.

Let V the set of points and consider the null graph $G = (V, E)$. Consider also $p_1 \in V$ and $p_2 \in V$ and the sphere regions $U_{p_1} = B(p_1, r_1)$ and $U_{p_2} = B(p_2, r_2)$. The new property P' can be defined as: $(p_1, p_2) \in E$ if and only if $(U_{p_1} \cap U_{p_2} \neq \emptyset) \wedge (\min\{r_1, r_2\}/\max\{r_1, r_2\}) > \theta$. The parameter θ can be understood as a minimum allowed limit value on the rate between the smaller and the greater radius of the two spheres. For example, if $\theta = 0.4$ then the size of smaller radius must be, at least, 40% of the value of the greater radius.

By using the property P' the induced θ -Sphere-of-influence based graph reflects the density based clusters as shows Figs. 32 and 33 and 34 (for the three sets of points shown in Figs. 26 and 27 and 28 respectively). Figure 32 shows the Sphere-of-influence based graph using $\theta = 0.71$, evidencing seven clusters as its connected components. Figure 33 shows the Sphere-of-influence based graph using $\theta = 0.6$, evidencing five successful clusters as its connected components and Fig. 34 shows the Sphere-of-influence based graph using $\theta = 0.65$, evidencing three clusters as its connected components.

Comparing the Sphere-of-influence based graphs shown in Figs. 26, 27 and 28 and Figs. 32, 33 and 34, it is noticeable that those in Figs. 32, 33 and 34 are improved versions of those in Figs. 26, 27 and 28 as far as the number of cluster is concerned. The improvement is due to the use of the θ parameter and was particularly evident for points in the low density regions. As can be seen in Fig. 32 the θ -Sphere-of-influence based graph with θ still does not precisely discriminate between high and medium density regions; in Fig. 34 the θ -Sphere-of-influence based graph induces only one central cluster instead of two, as expected. In spite of these problems that can be seen in Figs. 32 and 34, the θ -Sphere-of-influence based graph shows a little improvement in separating clusters that have different density regions over its counterpart that does not use the parameter θ .

Figure 35 shows an example using the modified θ -Sphere-of-influence in a set of points conforming to a more complex shape generated according to Eq. 13 and 14 and parameter settings shown in Table 1. The induced cluster for $\theta = 0.5$ can be seen in Fig. 35. It can be noted in the figure (marked with ellipsis) that points belonging to highly dense regions are well-separated from clusters induced in low density region. This will be observed also in the next two examples (Fig. 36 and Fig. 37).

$$x_{i_1} = 10 \times \cos(\theta) + B_1 \quad (13)$$

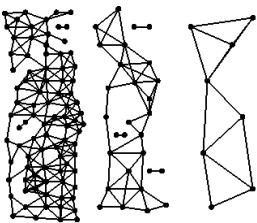


Fig. 32. θ -Sphere-of-influence based graph ($\theta = 0.71$)

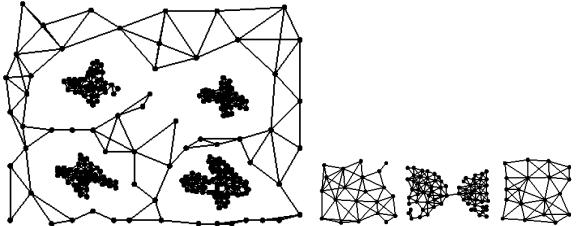


Fig. 33. θ -Sphere-of-influence based graph ($\theta = 0.6$)

Fig. 34. θ -Sphere-of-influence based graph ($\theta = 0.65$)

Table 1. Parameters used for creating data points shown in Fig. 35

θ	B_1	B_2	Total
Cluster 1 mean = 0 Standard deviation= 55	mean = 1 variance = 2	mean = 5 variance = 2	500
Cluster 2 mean = 180 Standard deviation= 55	mean = 1 variance = 2	mean = 5 variance = 2	500

$$x_{i2} = 10 \times \sin(\theta) + B_2 \quad (14)$$

The fourth situation for analyzing the influence of the neighborhood regions uses complex examples based on the random generation of Gaussian samples in a 2-dimensional space. For the two next examples the statistical concepts of mean and covariance were used as well as two extra parameters for the algorithm.

In order to deal with the two examples, Algorithm 2 was used with two extra parameters, L_1 and L_2 (defining an interval) to remove edges connecting points belonging to regions of different densities. In the first example (Fig. 36) it was used the Neighborhood Gabriel graph and in the second example (Fig. 37) the Relative Neighborhood graph. In both cases the removed edges were those whose Euclidian distance was within the interval limits. The value of L_1 should be greater than the Euclidian length of an edge in the high density region and L_2 should be lesser than the length of an edge in the low density region.

Figure 36 shows the 600 bi-dimensional points generated according to the normal distributions specified in Table 2, using the Neighborhood Gabriel graph and the interval [0.36, 0.8]. Although the algorithm induces single and small clusters in the low density region, the results show that clusters in the high density region and the clusters in the low density region are very distinguishably separated.



Fig. 35. θ -Sphere-of-influence based graph ($\theta = 0.5$)

Table 2. Parameters used for creating data points shown in Fig. 36

Mean Vector	Covariance Matrix $_{2 \times 2}$	Total
Cluster 1 [1.0138 0.9545]	[2.9465 -0.1299; -0.1299 3.0983]	300
Cluster 2 [6.0754 4.0872]	[2.9723 -0.0390; -0.0390 1.9843]	300

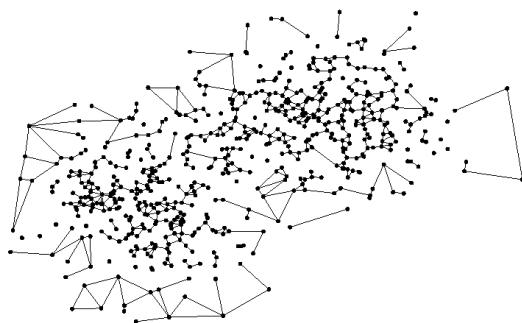
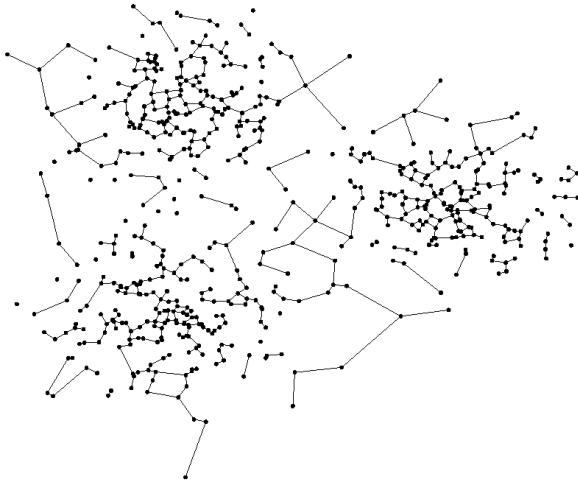


Fig. 36. Gabriel graph

Figure 37 also represents a set of 600 data points, with a slightly different configuration, according to parameter values given in Table 3. For this example was used the Relative Neighborhood region and the interval [0.48, 0.8]. As shows Fig. 37, although there are many (small) induced clusters, the three highly dense clusters have been successfully induced and can be identified from the others.

Table 3. Parameters used for creating data points shown in Fig. 37

	Mean Vector	Covariance Matrix $_{2 \times 2}$	Total
Cluster 1	[0.0914 -0.0962]	[3.1904 0.0839; 0.0839 2.7216]	200
Cluster 2	[7.9666 4.0396]	[3.4970 -0.1357; -0.1357 2.2209]	200
Cluster 3	[0.3858 8.1702]	[2.7714 0.3711; 0.3711 2.7874]	200

**Fig. 37.** Relative Neighborhood graph

5 Conclusion

For all sets of points in Sect. 3 the Modified Neighborhood Gabriel graph and the Modified Relative Neighborhood graph separate different density areas in different clusters as a consequence of parameter σ used to enlarge the neighborhood region defined by any two points. A suitable value for σ can be empirically determined and for the sets of points considered in this chapter the value was approximately 2, enough to cover a convenient area around each point allowing the split between areas with different densities. A larger value for σ induces a greater number of clusters and a smaller value for σ does not separate regions with different densities. The parameters α and β in the Elliptic Gabriel graph and β -Skeleton based graph respectively, do not have a convenient value that can be used in order to promote density based clusters. Changing the values of α (or β) will cause a formation of only one cluster or then, various clusters i.e., does not help inducing density based clusters.

With the introduction of the parameter θ , the θ -Sphere-of-influence based graph version has an improvement over its counterpart when inducing density based clusters. The parameter θ , however, does not modify the shape

of the neighborhood region like the parameters associated to other neighborhood regions; it helps, nevertheless, to avoid the definition of an edge between points that belong to different density regions. We intend next to explore a wider variety of density based configurations to confirm the results described in this chapter and also, to investigate the influence of the many neighborhood region definitions when inducing shape based clusters.

Acknowledgements. To CNPq and Fapesp for the financial help provided to the first and the second author respectively and to Leonie C. Pearson for proofreading the first draft of this paper.

References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. *ACM Computing Surveys* 31(3), 264–323 (1999)
2. Tan, P.N., Steinbach, M., Kumar, V.: *Introduction to data mining*. Addison-Wesley, Reading (2005)
3. Xu, R., Wunsch, D.: *Clustering*. IEEE Press Series on Computational Intelligence. Wiley/ IEEE Press, New York (2008)
4. Kaufman, L., Rousseeuw, P.: *Finding groups in data: an introduction to cluster analysis*. Wiley Series in Probability and Statistics. Wiley Interscience, New York (2005)
5. Jain, A.K., Dubes, R.C.: *Algorithms for clustering data*. Prentice-Hall, Englewood Cliffs (1988)
6. Arabie, P., Hubert, L.J., Soete, G.D. (eds.): *Clustering and Classification*. World Scientific, River Edge (1998)
7. Duda, R.O., Hart, P.E., Store, D.G.: *Pattern classification*. John Wiley, New York (2001)
8. Theodoridis, S., Koutroumbas, K.: *Pattern recognition*. Academic Press, San Diego (1998)
9. Kruskal, J.B.: On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. American Math. Soc.* 7, 48–50 (1965)
10. Prim, R.C.: Shortest Connection networks and some generalizations. *Bell System Technical Journal* 36, 1389–1401 (1957)
11. Zahn, C.T.: Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers* 20(1), 68–86 (1971)
12. Liotta, G., Lubiw, A., Meijer, H., Whitesides, S.H.: The rectangle of influence drawability problem, Technical Report CS-96-22, Department of Computer Science, Brown University, USA (June 1996)
13. Hartuv, E., Shamir, R.: A clustering algorithm based on graph connectivity. *Information Processing Letters* 76(4–6), 175–181 (1999)
14. Sharan, R., Shamir, R.: CLICK: a clustering algorithm with applications to gene expression analysis. In: Proc. of 8th International Conference on Intelligent Systems for Molecular Biology (ISMB), pp. 307–316. AAAI Press, Menlo Park (2000)
15. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences* 95, 14863–14868 (1998)

16. Ben-Dor, A., Yakhini, Z.: Clustering gene expression patterns. *Journal of Computational Biology* 6(3-4), 281–297 (1999)
17. Päivinen, N.: Clustering with minimum spanning tree of scale-free structure. *Pattern Recognition* 26, 921–930 (2005)
18. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* 286, 509–512 (1999)
19. Asano, T., Bhattacharia, B., Keil, M., Yao, F.: Clustering algorithms based on minimum and maximum spanning trees. In: *Proceedings of the Fourth Annual Symposium on Computational Geometry*, Urbana-Champaign, Illinois, pp. 252–257 (1998)
20. Gabriel, K., Sokal, R.: A new statistical approach to geographic variation analysis. *Systematic Zoology* 18(3), 259–278 (1969)
21. Toussaint, G.T.: The relative neighborhood graph of a finite planar set. *Pattern Recognition* 12, 261–268 (1980)
22. Urquhart, R.: Graph theoretical clustering based on limited neighborhood sets. *Pattern Recognition* 15(3), 173–187 (1982)
23. Hopcroft, J., Tarjan, R.: Efficient algorithms for graph manipulation. *CACM* 16(6), 372–378 (1973)
24. Park, J.C., Shin, H., Choi, B.K.: Elliptic Gabriel graph for finding neighbors in a point set and its application to normal vector estimation. *Computer-Aided Design* 38, 619–626 (2006)
25. Kirkpatrick, D.G., Radke, J.D.: A framework for computational morphology. In: Toussaint, G.T. (ed.) *Computational Geometry*, pp. 217–248. North-Holland, Amsterdam (1985)
26. Toussaint, G.T.: A graph-theoretical primal sketch. In: Toussaint, G.T. (ed.) *Computational Morphology*, pp. 229–260. North-Holland, Amsterdam (1988)
27. Jaromczyk, J.W., Toussaint, G.T.: Relative neighborhood graphs and their relatives. *Proc. of the IEEE* 80(9), 1502–1517 (1992)
28. Amenta, N., Bern, M., Eppstein, D.: The crust and the β -Skeleton: combinatorial curve reconstruction. *Graph Models Image Process* 60(2), 125–135 (1998)
29. Zhang, W., King, I.: Locating support vectors via β -Skeleton technique. In: *Proceedings of the International Conference on Neural Information Processing (ICONIP)*, pp. 1423–1427 (2002)
30. Klein, J., Zachmann, G.: Point cloud surfaces using geometric proximity graphs. *Computers & Graphics* 28(6), 839–850 (2004)

Some Issues on Extensions of Information and Dynamic Information Systems

Krzysztof Pancerz^{1,2}

¹ University of Information Technology and Management

Sucharskiego Str. 2, 35-225 Rzeszów, Poland

kpancerz@wsiz.rzeszow.pl

² Zamość University of Management and Administration

Akademicka Str. 4, 22-400 Zamość, Poland

Summary. The aim of the chapter is to give the basic principles and issues of extensions of information and dynamic information systems. Information systems are understood in the Pawlak's sense as the systems of knowledge representation. The extensions of information and dynamic information systems can be used in prediction problems. Assuming that objects of an information system represent states of a system of processes we can determine possibility of appearing new states in the system of processes in the future. Analogously, assuming that a dynamic information system includes, among others, information about transition between states of a system of processes, we can determine possibility of appearing new transitions between states in the future.

Keywords: information system, dynamic information system, extensions, prediction, possibility distribution.

1 Introduction

An information system can represent a finite set of states of a given system. Each attribute represents an individual component of a system (called a process). A finite set of internal (local) states is associated with each process. Each object is interpreted as a global state of the system (a record of local states of individual processes). A dynamic information system additionally includes information on transitions between global states (i.e., information on behavior of a system in time). Besides all global states appearing in the original information system, an extension of it can include new global states, which have not been observed yet, but which are consistent to a certain degree with the knowledge included in the original system. Analogously, an extension of a dynamic information system can include new transitions between global states, not observed yet. The knowledge can be represented in the form of rules (production, association, etc.). One can consider only the

so called consistent extensions of information systems, when all new global states are totally consistent with the knowledge included in the original information systems. However, in general case, we can consider partially consistent extensions, when some new global states are consistent with the knowledge possessed only to a certain degree. The important problem is to determine consistency factors of new global states or new transitions taking into consideration different ways of knowledge representation. The chapter shows theoretical foundations, methods and algorithms, and some applications in prediction problems for extensions of information and dynamic information systems.

The rest of the chapter is organized as follows. In Section 2, a brief review of the literature concerning extensions of information and dynamic information systems and a motivation for this chapter are presented. In Sections 3 and 4, basic definitions, notions and notations concerning information systems and dynamic information systems, respectively, are recalled. Sections 5 and 6 present basic definitions, algorithms and applications for extensions of information systems and dynamic information systems, respectively. Finally, Section 7 consists of some conclusions.

2 Literature Review and Motivation

A notion of an extension of an information system has been introduced in [9]. The authors considered consistent and maximal consistent extensions of information systems. For generating a maximal consistent extension of a given information system, the authors proposed to use the concurrent model in the form of a place-transition net constructed for this system. An application of maximal consistent extensions in the design of concurrent systems described by information systems has been considered, among others, in [10], [11], [13]. Generating a consistent extension of a given information system using Boolean formulas determined on the basis of information vectors representing objects in a data table has been considered in [14]. A new method for determining consistent extensions of information systems has been given in [8]. The authors presented necessary and sufficient conditions for the existence of such extensions. In [15], a notion of a partially consistent extension of an information system has been introduced. The notion of a dynamic information system was introduced by Z. Suraj in [12]. In [12], a method of computing maximal consistent extensions of dynamic information systems has been proposed. The method requires computing all transition rules from the transition system. In [16], the authors proposed an improved method of computing maximal consistent extensions of dynamic information systems. A necessary condition for the existence of a non-trivial maximal consistent extension of a given dynamic information system was given. Moreover, a method of computing the maximal consistent extension without the necessity of computing all transition rules was presented. The majority of methods for determining consistent and partially consistent extensions of information

systems and dynamic information systems presented earlier (with the exception of [8]) required computing all minimal rules (or only a part of them) true and realizable in information and dynamic information systems, respectively. Therefore, working out efficient methods became an important research problem. One of such methods has been presented in [2] and [18].

The main objective of this chapter is to systematize and gather some issues concerning extensions of information and dynamic information systems presented in the literature until now. Moreover, we adjust an efficient method of computing consistent and partially consistent extensions of information systems to computing consistent and partially consistent extensions of dynamic information systems. We also try to look on partially consistent extensions of information and dynamic information systems from the point of view of the possibility theory [19].

3 Information Systems

The notion of an information system is one of fundamental notions of the rough set theory. Information systems (called also, in literature, knowledge representation systems [6], information tables, attribute-value systems) are a tool for data representation. Such data may come from measurements, observations, specifications of some phenomena, signals, systems, etc. The data representation by means of information systems can be compared with the data representation in relational databases.

3.1 Basic Definitions

In this subsection, we recall two definitions concerning information systems. The first one is a definition of an information system whereas the second one is a definition of a special kind of an information system which is called a decision system.

Definition 1 (Information System). *An information system is an ordered pair $S = (U, A)$, where:*

- *U is a nonempty, finite set of objects which is also called universum,*
- *A is a nonempty, finite set of attributes.*

Each attribute $a \in A$ is a total function $a : U \rightarrow V_a$, where V_a is a set of values of the attribute a .

In some information systems, we can distinguish two classes of attributes called condition attributes and decision attributes. Such information systems are called decision systems.

Definition 2 (Decision System). *A decision system is an information system $S = (U, A)$, where $A = C \cup D$ and $C \cap D = \emptyset$ (empty set). C is a set of condition attributes (in short, conditions) whereas D is a set of decision attributes (in short, decisions).*

Each information (decision) system $S = (U, A)$ can be presented in the form of a data table. Columns are labeled with attributes from A whereas rows are labeled with objects from U . Cells of the table include values of appropriate attributes. Each row of the table delivers information about one object in the system S .

3.2 Decision Logic Language

A formal language $L(S)$ of decision logic can be associated with a given information (decision) system $S = (U, A)$ (see [6]). The alphabet of the language $L(S)$ consists of:

- A - a set of constants corresponding to attributes of the information system S ,
- $V = \bigcup_{a \in A} V_a$ - a set of constants corresponding to values of attributes in the information system S ,
- $\{\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow\}$ - a set of propositional connectives, called negation, disjunction, conjunction, implication and equivalence, respectively.

It is easy to see, that the alphabet of $L(S)$ does not consist of any variables. Formulae (expressions) in the language $L(S)$ are built up from symbols of attributes, symbols of values of attributes, propositional connectives, and some auxiliary symbols like parentheses. A set of all formulae of the language $L(S)$ is denoted by $F(S)$.

Definition 3 (Atomic Formula). Let $S = (U, A)$ be an information system and $L(S)$ a decision logic language of S . An expression (a, v) , where $a \in A$ and $v \in V_a$, is called an atomic formula (or elementary formula) of $L(S)$.

A set $F(S)$ of all formulae of $L(S)$ is the least set satisfying the following conditions:

- (a, v) is a formula of $L(S)$,
- if ϕ and ψ are formulae of $L(S)$, then so are $\neg\phi$, $\phi \vee \psi$, $\phi \wedge \psi$, $\phi \Rightarrow \psi$, and $\phi \Leftrightarrow \psi$.

Remark 1. Let $S = (U, A)$ be an information (decision) system, $B \subseteq A$, and $L(S)$ a decision logic language of S . By $\phi|_B$ we denote a formula of $L(S)$ consisting of only atomic formulae in the form (a, v) , where $a \in B$, $v \in V_a$, and V_a is a set of values of the attribute a . Such a formula is called a formula of $L(S)$ restricted to the set B of attributes in the system S .

An essential thing is satisfiability of a given formula ϕ of $L(S)$ by an object $u \in U$ in the information system $S = (U, A)$, what is written as $u \models_S \phi$. Satisfiability of a formula may be defined inductively with respect to complexity of a formula:

- $u \models_S (a, v)$ if and only if $a(u) = v$,
- $u \models_S (\neg\phi)$ if and only if non $u \models_S \phi$,
- $u \models_S (\phi \vee \psi)$ if and only if $u \models_S \phi$ or $u \models_S \psi$,
- $u \models_S (\phi \wedge \psi)$ if and only if $u \models_S \phi$ and $u \models_S \psi$,
- $u \models_S (\phi \Rightarrow \psi)$ if and only if $u \models_S (\neg\phi \vee \psi)$,
- $u \models_S (\phi \Leftrightarrow \psi)$ if and only if $u \models_S (\phi \Rightarrow \psi)$ and $u \models_S (\psi \Rightarrow \phi)$.

Definition 4 (Set of objects satisfying a formula). Let $S = (U, A)$ be an information system and $L(S)$ a decision logic language of S . A set of objects of S satisfying a formula ϕ of $L(S)$, denoted by $|\phi|_S$, is defined as:

$$|\phi|_S = \{u \in U : u \models_S \phi\}. \quad (1)$$

The set $|\phi|_S$ may be also defined inductively with respect to complexity of a formula:

- $|(a, v)|_S = \{u \in U : a(u) = v\}$,
- $|\neg\phi|_S = U - |\phi|_S$,
- $|\phi \vee \psi|_S = |\phi|_S \cup |\psi|_S$,
- $|\phi \wedge \psi|_S = |\phi|_S \cap |\psi|_S$,
- $|\phi \Rightarrow \psi|_S = (U - |\phi|_S) \cup |\psi|_S$,
- $|\phi \Leftrightarrow \psi|_S = (|\phi|_S \cap |\psi|_S) \cup ((U - |\phi|_S) \cap (U - |\psi|_S))$.

3.3 Rules in Information Systems

Dependencies among values of attributes in an information system may be expressed by means of the so-called rules. The notion of a rule is one of essential notions used in this paper. Rules are designed as a tool for the purpose of the knowledge representation.

Definition 5 (Rule). Let $S = (U, A)$ be an information system and $L(S)$ a decision logic language of S . A rule in the information system S is a formula ρ of $L(S)$ in the form $\phi \Rightarrow \psi$. The formula ϕ is called a predecessor of ρ whereas the formula ψ is called a successor of ρ .

The predecessor of ρ will be denoted by $Pred(\rho)$ whereas the successor of ρ will be denoted by $Succ(\rho)$.

Remark 2. Let $S = (U, A)$ be an information system. In the paper, for the system S we consider rules in the form:

$$(a_{i_1}, v_{i_1}) \wedge (a_{i_2}, v_{i_2}) \wedge \dots \wedge (a_{i_r}, v_{i_r}) \Rightarrow (a_d, v_d), \quad (2)$$

where $a_d \in A$ and $v_d \in V_{a_d}$, while $a_{i_j} \in B \subseteq A - \{a_d\}$ and $v_{i_j} \in V_{a_{i_j}}$ for $j = 1, 2, \dots, r$.

Therefore, predecessors of rules are conjunctions of atomic formulae. A successor of a rule is an atomic formula.

Remark 3. Let $S = (U, C \cup D)$ be a decision system, where C is a set of condition attributes whereas D is a set of decision attributes. In the paper, for the system S we consider rules in the form:

$$(a_{i_1}, v_{i_1}) \wedge (a_{i_2}, v_{i_2}) \wedge \dots \wedge (a_{i_r}, v_{i_r}) \Rightarrow (a_d, v_d), \quad (3)$$

where $a_{i_j} \in B \subseteq C$ and $v_{i_j} \in V_{a_{i_j}}$ dla $j = 1, 2, \dots, r$, while $a_d \in D$ and $v_d \in V_{a_d}$.

Rules considered by us satisfy three requirements, namely each rule is:

- true,
- minimal,
- realizable

in a given information system.

Definition 6 (True Rule). Let $S = (U, A)$ be an information (decision) system and ρ a rule in the form $\phi \Rightarrow \psi$ in S . The rule ρ is true in S , what is denoted as $\phi \xrightarrow{S} \psi$, if:

$$\forall_{u \in U} u \models_S (\phi \Rightarrow \psi). \quad (4)$$

Otherwise, the rule ρ is not true in S .

Remark 4. For simplicity, a true rule in S will be also written as $\phi \Rightarrow \psi$ instead of $\phi \xrightarrow{S} \psi$.

Definition 7 (Minimal Rule). Let $S = (U, A)$ be an information (decision) system and ρ a rule in the form:

$$(a_{i_1}, v_{i_1}) \wedge (a_{i_2}, v_{i_2}) \wedge \dots \wedge (a_{i_r}, v_{i_r}) \Rightarrow (a_d, v_d) \quad (5)$$

true in S . The rule ρ is a minimal rule if removing any atomic formula (a_{i_j}, v_{i_j}) , where $j = 1, 2, \dots, r$, from the predecessor of a rule makes this rule not true in S .

Definition 8 (Realizable Rule). Let $S = (U, A)$ be an information (decision) system and ρ a rule in the form:

$$(a_{i_1}, v_{i_1}) \wedge (a_{i_2}, v_{i_2}) \wedge \dots \wedge (a_{i_r}, v_{i_r}) \Rightarrow (a_d, v_d) \quad (6)$$

true in S . The rule ρ is a realizable rule if:

$$|(a_{i_1}, v_{i_1}) \wedge (a_{i_2}, v_{i_2}) \wedge \dots \wedge (a_{i_r}, v_{i_r}) \wedge (a_d, v_d)|_S \neq \emptyset. \quad (7)$$

Any object belonging to

$$|(a_{i_1}, v_{i_1}) \wedge (a_{i_2}, v_{i_2}) \wedge \dots \wedge (a_{i_r}, v_{i_r}) \wedge (a_d, v_d)|_S \quad (8)$$

is referred to as an object supporting a rule.

A set of all minimal rules true and realizable in the information system S is denoted by $Rul(S)$.

Some factors can be associated with each rule in the information (decision) system. These factors can be used, for example, to determine the importance of rules in a given system.

Definition 9 (Support factor of a rule). Let $S = (U, A)$ be an information (decision) system and ρ a rule in the form $\phi \Rightarrow \psi$ in S . A number

$$supps(\phi \Rightarrow \psi) = card(|\phi \wedge \psi|_S) \quad (9)$$

is called a support factor (in short, support) of the rule ρ in S .

Definition 10 (Strength factor of a rule). Let $S = (U, A)$ be an information (decision) system and ρ a rule in the form $\phi \Rightarrow \psi$ in S . A number

$$strs(\phi \Rightarrow \psi) = \frac{supps(\phi \Rightarrow \psi)}{card(U)} \quad (10)$$

is called a strength factor (in short, strength) of the rule ρ in S .

Definition 11 (Certainty factor of a rule). Let $S = (U, A)$ be an information (decision) system and ρ a rule in the form $\phi \Rightarrow \psi$ in S . A number

$$cers(\phi \Rightarrow \psi) = \frac{supps(\phi \Rightarrow \psi)}{card(|\phi|_S)} \quad (11)$$

is called a certainty factor (in short, certainty) of a rule ρ in S .

It is worth noting that for any rule $\phi \Rightarrow \psi$ true in a given information (decision) system S , its certainty factor equals to 1, i.e., $cers(\phi \Rightarrow \psi) = 1$.

In some situations, it is legitimate to consider, for a given information system S , only some rules from the set $Rul(S)$, for example, only strong rules, i.e., rules, for which a strength factor is not smaller than a threshold value. This means that we are interested in rules in a given information system which are supported by a proper number of objects.

Remark 5. Let S be an information system. By $StrRul^{(\tau)}(S)$ we denote a set of all strong rules minimal, true and realizable in S , for which a strength factor is not smaller than τ , where $0 \leq \tau \leq 1$, i.e.:

$$StrRul^{(\tau)}(S) = \{\rho \in Rul(S) : strs(\rho) \geq \tau\}. \quad (12)$$

For any set $Rul^*(S)$ of rules true in the information system S , we can calculate some numerical factors such as those defined below.

Definition 12 (Support of a set of rules). Let $S = (U, A)$ be an information system and $Rul^*(S)$ any set of rules true in S . A number $supp(Rul^*(S))$ is called support of the set $Rul^*(S)$ of rules in S and it is defined as:

$$supp(Rul^*(S)) = card \left(\bigcup_{(\phi \Rightarrow \psi) \in Rul^*(S)} |\phi \wedge \psi|_S \right). \quad (13)$$

Support of a set $Rul^*(S)$ of rules is equal to the number of objects in the information system S satisfying simultaneously the predecessor and successor of at least one rule from the set $Rul^*(S)$. Obviously, $0 \leq supp(Rul^*(S)) \leq card(U)$.

Definition 13 (Strength of a set of rules). Let $S = (U, A)$ be an information system and $Rul^*(S)$ any set of rules true in S . A number $str(Rul^*(S))$ is called a strength of the set $Rul^*(S)$ of rules in S and it is defined as:

$$str(Rul^*(S)) = \frac{supp(Rul^*(S))}{card(U)}. \quad (14)$$

Strength of the set $Rul^*(S)$ of rules is a relative measure of support with respect to the number of all objects in the information system S . For strength, we have $0 \leq str(Rul^*(S)) \leq 1$.

3.4 Application: Description of Systems of Processes

An application of information systems as a tool of a description of systems of processes was proposed in [7] and then it was developed by different authors (see [9], [10], [11], [13]).

Let $S = (U, A)$ be an information system. Elements of the set U may be interpreted as global states of a given system \mathbb{S} of processes whereas attributes (elements of the set A) are interpreted as processes in \mathbb{S} . A set V_a of local states is associated with each process a from A . A description of the system \mathbb{S} may be presented in the form of a data table. Columns are labeled with names of processes of \mathbb{S} . Each row of a table (labeled with an object from U) includes a record of local states of processes. Each record may be interpreted as a global state of \mathbb{S} .

Remark 6. In the further part of this paper, we will use additionally (interchangeably) the following terms for an information system S :

- attributes of S will be also called processes in S ,
- objects of S will be also called global states in S ,
- values of attributes in S will be also called local states of processes in S .

4 Dynamic Information Systems

A description of systems of processes by means of information systems does not take into consideration their behavior. In [12], dynamic information systems were proposed for describing systems of processes. Behavior of systems of processes can be easily described using dynamic information systems.

Let $DS = (U, A, E, T)$, where $S = (U, A)$ and $TS = (U, E, T)$, be a dynamic information system. The set $A = \{a_1, a_2, \dots, a_m\}$ can be treated as nonempty, finite set of processes. We associate a finite set V_a of local states with each process $a \in A$. Behavior of a system \mathbb{S} of processes is presented by means of two linked subtables marked with S and TS . The first subtable S represents global states of a given system \mathbb{S} of processes, observed by us and it is called an underlying system of DS . The second subtable TS represents a transition relation T between global states of \mathbb{S} and it is called a transition system of DS . Each row of the first subtable includes a record of local states of processes from A . Each record is labeled with an element from the set U of global states of \mathbb{S} . The second subtable represents a transition relation. Columns of the second subtable are labeled with events from the set E . Rows, analogously, with global states from U . Elements of the second subtable make up successor states of a given global state (labeling a proper row). The first global state of an underlying system can be an initial state of a transition system.

4.1 Basic Definitions

In this subsection , we give formal definitions concerning dynamic information systems and describe representation of dynamic information systems by means of data tables.

Definition 14 (Transition System). *A transition system is an ordered triple $TS = (S, E, T)$, where:*

- S is a nonempty set of states,
- E is a nonempty set of events,
- $T \subseteq S \times E \times S$ is a transition relation.

A transition system is finite if the sets S and E are finite. Selecting an initial state $s_0 \in S$ we obtain an initialized transition system.

Definition 15 (Initialized Transition System). *Initialized transition system is an ordered quadruple $TS^{in} = (S, E, T, s_0)$, where:*

- $TS = (S, E, T)$ is a transition system,
- $s_0 \in S$ is an initial state of TS .

Let $TS = (S, E, T)$ be a transition system. Each ordered triple $(s, e, s') \in T$ is called a transition in TS , occurring due to an event e . The state s is called a previous state of the transition (s, e, s') whereas the state s' is called a next state of the transition (s, e, s') .

Definition 16 (Dynamic information system). A dynamic information system is an ordered quadruple $DS = (U, A, E, T)$, where:

- $S = (U, A)$ is an information system called an underlying system of DS ,
- $TS = (U, E, T)$ is a transition system of DS .

We can define, analogously, an initialized dynamic information system.

Definition 17 (Initialized Dynamic Information System). Initialized dynamic information system is an ordered quintuple $DS^{in} = (U, A, E, T, u_0)$, where:

- $S = (U, A)$ is an information system called an underlying system of DS^{in} ,
- $TS^{in} = (U, E, T, u_0)$ is an initialized transition system of DS^{in} , where $u_0 \in U$.

4.2 Representation of Dynamic Information Systems

In our considerations, dynamic information systems are presented by means of two data tables representing information systems in the Pawlak's sense. The first data table represents an underlying system of a given dynamic information system DS . The second data table, called further, a decision transition system, represents transitions given by a transition relation in DS .

Let $DS = (U, A, E, T)$ be a dynamic information system, where $S = (U, A)$ is its underlying system. We construct a decision table representing a decision system $S_T = (U_T, A \cup A')$, where $A = \{a_1, \dots, a_m\}$ is a set of condition attributes whereas $A' = \{a'_1, \dots, a'_m\}$ is a set of decision attributes. Each attribute $a' \in A'$ corresponds to exactly one attribute $a \in A$. Such a table includes selected pairs of global states from the underlying system S . Each row of a decision table corresponds to one transition between global states $u, u' \in U$ determined by a transition relation T , i.e., transition, for which there exists an event $e \in E$ such that $(u, e, u') \in T$. Condition attributes a_1, \dots, a_m determine previous states of transitions fixed by the relation T whereas decision attributes a'_1, \dots, a'_m determine next states fixed by the relation T .

Remark 7. Objects in decision transition systems will be shortly called transitions.

4.3 Rules in Dynamic Information Systems

In a dynamic information system, we can consider two kinds of rules:

- rules of an underlying system called shortly underlying rules,
- rules of a decision transition system called shortly transition rules.

Definition 18 (Underlying rule). Let $DS = (U, A, E, T)$ be a dynamic information system, $S = (U, A)$ its underlying system, and $L(S)$ a decision logic language of S . An underlying rule in DS is a formula ρ of $L(S)$ in the form $\phi \Rightarrow \psi$, where ϕ and ψ are formulae of $L(S)$.

Underlying rules represent dependencies among values of attributes in the underlying system S of a dynamic information system DS .

Definition 19 (Transition rule). Let $DS = (U, A, E, T)$ be a dynamic information system, $S_T = (U_T, A \cup A')$ its decision transition system, and $L(S_T)$ a decision logic language of S_T . A transition rule in DS is a formula ρ of $L(S_T)$ in the form $\phi|_A \Rightarrow \psi|_{A'}$, where $\phi|_A$ and $\psi|_{A'}$ are formulae of $L(S_T)$ restricted to sets of attributes A and A' , respectively.

Transition rules represent dependencies between values of condition and decision attributes in a decision transition system.

Remark 8. Let $DS = (U, A, E, T)$ be a dynamic information system, $S = (U, A)$ its underlying system, and $S_T = (U_T, A \cup A')$ its decision transition system. We will consider underlying rules in the form

$$(a_{i_1}, v_{i_1}) \wedge (a_{i_2}, v_{i_2}) \wedge \dots \wedge (a_{i_r}, v_{i_r}) \Rightarrow (a_d, v_d), \quad (15)$$

where $a_d \in A$ and $v_d \in V_{a_d}$, while $a_{i_j} \in B \subseteq A - \{a_d\}$ and $v_{i_j} \in V_{a_{i_j}}$ for $j = 1, 2, \dots, r$. We will consider transition rules in the form

$$(a_{i_1}, v_{i_1}) \wedge (a_{i_2}, v_{i_2}) \wedge \dots \wedge (a_{i_r}, v_{i_r}) \Rightarrow (a_d, v_d), \quad (16)$$

where $a_{i_j} \in B \subseteq A$ and $v_{i_j} \in V_{a_{i_j}}$ for $j = 1, 2, \dots, r$, while $a_d \in A'$ and $v_d \in V_{a_d}$.

Thus in our considerations, predecessors of underlying and transition rules are conjunctions of adequate atomic formulae whereas their successors are atomic formulae. Moreover, we consider only underlying and transition rules which are true, minimal and realizable in proper systems. Definitions of such rules are analogous to definitions for information (decision) systems (see Subsection 3.3). Each underlying and transition rule can be characterized by three factors: support, strength, and certainty. The ways of their calculations are similar to calculations for information systems given in Subsection 3.3.

A set of all minimal and realizable underlying rules true in a dynamic information system DS will be denoted by $UndRul(DS)$. A set of all minimal and realizable transition rules true in a dynamic information system DS will be denoted by $TranRul(DS)$.

5 Extensions of Information Systems

Extensions of information systems have been considered, among others, in [8], [14]. Any extension S^* of a given information system S is created by adding to the system S new objects whose signatures contain only values of attributes which appeared in S . Especially, an important role among extensions of a given information system S is played by the so-called *consistent extension* of S . Such an extension is obtained if all the new objects added

to the system S satisfy each minimal rule true in S . Among all possible consistent extensions of a given information system S there exists the so-called *maximal consistent extension* (with respect to the number of objects). If an information system S describes a system \mathbb{S} of processes, then its maximal consistent extension contains all the global states consistent with all the minimal rules representing dependencies among local states of processes of \mathbb{S} . These rules are extracted from the information system S . In some cases, a maximal consistent extension can contain new global states of \mathbb{S} , which have not been earlier observed in \mathbb{S} , but these new states are consistent with rules extracted from S . Then, we have a non-trivial maximal consistent extension of S . The new global states can be treated as the new knowledge of \mathbb{S} .

An approach to consistent extensions of information systems can be generalized using the so-called partially consistent extensions. In a case of a partially consistent extension S^* of a given information system S , we admit a situation that new objects added to S satisfy only a part of all minimal rules true in S . Then, an essential thing is to determine a consistency factor of a new object added to S with the knowledge $\mathcal{K}(S)$ included in the original information system S . Here, we propose some method of computing consistency factors. This method is based on rough set theory. Computing a consistency factor for a given object is based on determining importance (relevance) of rules extracted from the system S which are not satisfied by the new object. We assume that if the importance of these rules is greater the consistency factor of a new object with the knowledge $\mathcal{K}(S)$ is smaller. The importance of a set of rules not satisfied by the new object is determined by means of a strength factor of this set of rules in S . It is worth noting that different approaches to determining a consistency factor are also possible.

5.1 Basic Definitions

In this section, we present some formal definitions concerning extensions of information systems.

Definition 20 (Extension of an information system). Let $S = (U, A)$ be an information system. An information system $S^* = (U^*, A^*)$ is an extension of S , where::

- $U \subseteq U^*$,
- $\text{card}(A) = \text{card}(A^*)$,
- for each $a \in A$, there exists $a^* \in A^*$ such that a function $a^* : U^* \rightarrow V_a$ is an extension of a function $a : U \rightarrow V_a$ to U^* .

It is easy to see that a data table representing an information system S is a part of a data table representing an extension S^* of S , i.e., each object which appears in S , appears also in S^* .

Remark 9. A set A^* of attributes in an extension $S^* = (U^*, A^*)$ of an information system $S = (U, A)$ will be denoted by A like in the original system

S . So, we write $S^* = (U^*, A)$ instead of $S^* = (U^*, A^*)$. The same applies to attributes of A^* , i.e., $a_1^*, a_2^*, \dots, a_m^* \in A^*$. So, we write a_1, a_2, \dots, a_m instead of $a_1^*, a_2^*, \dots, a_m^*$, where $a_1, a_2, \dots, a_m \in A$.

A number of extensions of an information system $S = (U, A)$ is equal to $2^{n-k} - 1$ (see [14]), where $k = \text{card}(U)$, $A = \{a_1, a_2, \dots, a_m\}$, $n = \text{card}(V_{a_1} \times V_{a_2} \times \dots \times V_{a_m})$, V_{a_i} is a value set of a_i for $i = 1, 2, \dots, m$.

Remark 10. A nontrivial extension of an information system $S = (U, A)$ is an extension $S^* = (U^*, A)$ such that $U \subset U^*$.

A set of all extensions of a given information system S will be denoted by $\text{Ext}(S)$. This set can be partially ordered by relation \leq defined as follows (see [8]):

$$\forall_{S_1^* = (U_1^*, A), S_2^* = (U_2^*, A) \in \text{Ext}(S)} S_1^* \leq S_2^* \text{ if and only if } U_1^* \subseteq U_2^*. \quad (17)$$

Maximal elements in $\text{Ext}(S)$ partially ordered by relation \leq defined earlier are called maximal extensions of an information system S .

Theorem 1. *For each information system, there exists exactly one maximal extension.*

Proof. For a given information system $S = (U, A)$, where $A = \{a_1, a_2, \dots, a_m\}$, its maximal extension includes all objects determined by the Cartesian product of value sets of attributes, i.e., $V_{a_1} \times V_{a_2} \times \dots \times V_{a_m}$.

The maximal extension of an information system S is also called a Cartesian extension of S and it is denoted by S^{\max} . In that case, a Cartesian extension is defined as follows.

Definition 21 (Cartesian extension of an information system). *Let $S = (U, A)$ be an information system, $A = \{a_1, a_2, \dots, a_m\}$ and $\{V_{a_i}\}_{i=1,2,\dots,m}$ a family of value sets of attributes from A . An information system $S^{\max} = (U^{\max}, A)$ is a Cartesian extension of S if:*

$$U^* = \{u : a_1(u) \in V_{a_1}, a_2(u) \in V_{a_2}, \dots, a_m(u) \in V_{a_m}\}. \quad (18)$$

Let $S = (U, A)$ be an information system, $S^* = (U^*, A)$ its extension, and $u \in U^*$. By $Rul_u^\sim(S)$ we denote a set of all minimal rules true and realizable in S which are not satisfied by u , i.e.,

$$Rul_u^\sim(S) = \{(\phi \Rightarrow \psi) \in Rul(S) : \text{not } u \models (\phi \Rightarrow \psi)\}. \quad (19)$$

For any object u from the extension S^* of a given information system S , we define a coefficient called a consistency factor. This coefficient expresses a degree of consistency of u with the knowledge included in the original system S .

Remark 11. We assume that the knowledge included (hidden) in an information system S is expressed by the set $Rul(S)$ of all minimal rules true and realizable in S .

Definition 22 (Consistency factor of an object). Let $S = (U, A)$ be an information system, $S^* = (U^*, A)$ its extension, and $u \in U^*$. A consistency factor of u with the knowledge included in S is a number defined as:

$$\xi_S(u) = 1 - str(Rul_u(S)). \quad (20)$$

A consistency factor satisfies inequalities $0 \leq \xi_S(u) \leq 1$ for each $u \in U^*$. Obviously, if $u \in U$, then $\xi_S(u) = 1$, because $Rul_u^*(S) = \emptyset$.

Definition 23 (Consistent extension of an information system). Let $S = (U, A)$ be an information system and $S^* = (U^*, A)$ its extension. S^* is called a consistent extension of S if and only if $\xi_S(u) = 1$ for each $u \in U^*$.

Definition 24 (Partially consistent extension of an information system). Let $S = (U, A)$ be an information system and $S^* = (U^*, A)$ its extension. S^* is called a partially consistent extension of S if and only if there exists at least one object $u \in U^*$ such that $\xi_S(u) < 1$.

5.2 Computing Consistency Factors of Objects

In this subsection, we give an efficient algorithm for computing a consistency factor of any object u^* from the extension of a given information system S with the knowledge included in S and expressed by all minimal rules true and realizable in S (the set $Rul(S)$). An approach proposed here does not involve computing any rules from an original information system. The algorithm presented here allows us to determine a set of objects from an original information system S supporting minimal rules from $Rul(S)$, but not satisfied by the object u^* . A consistency factor is computed as a complement to 1 of the strength of the set of rules not satisfied. This algorithm takes advantage of the theorem proposed in [1]. Here, the theorem and its proof taken from [1] are expressed by means of the formalism used in the chapter.

Theorem 2. Let $S = (U, A)$ be an information system, $S^* = (U^*, A)$ its extension, $Rul(S)$ a set of all minimal rules true and realizable in S and $u^* \in U^*$. For each $u \in U$ let $M_u = \{a \in A : a(u^*) = a(u)\}$, and for each $a \in A - M_u$:

$$P_u^a = \{a(u') : u' \in U \text{ and } \forall_{a' \in M_u} a'(u') = a'(u)\}. \quad (21)$$

The object u^* satisfies all rules from $Rul(S)$ if and only if for any $u \in U$ we have $card(P_u^a) \geq 2$ for each $a \in A - M_u$.

Proof. Let us assume that for a new object u^* there exists an object $u \in U$ such that $M_u \neq \emptyset$ and $\text{card}(P_u^a) = 1$ for some $a \in A - M_u$. Then the rule r defined by:

$$\left(\bigwedge_{a' \in M_u} (a', a'(u)) \right) \Rightarrow (a, a(u)) \quad (22)$$

is true in S , because $\text{card}(P_u^a) = 1$. We also have that $a(u) \neq a(u^*)$ because $a \notin M_u$. Hence, the rule r (and also each minimal rule true in S obtained from r by the removal of some atomic formulae from the predecessor of r) is not satisfied by u^* .

Let us assume that u^* does not belong to any consistent extension of S . It means that there exists a minimal rule r in the form $\phi \Rightarrow (a, v)$, where ϕ is a conjunction of atomic formulae of $L(S)$, which is not satisfied by u^* , but it is true and realizable in S . Hence, $u^* \in |\phi|_{S^*}$ and $a(u^*) \neq v$. The rule r is realizable in S . Hence, for some $u \in U$ we have $u \in |\phi|_S$ and $a(u) = v$. M_u consists of all attributes from atomic formulae contained in ϕ . M_u can also contain some other attributes. Hence

$$\left| \bigwedge_{a' \in M_u} (a', a'(u)) \right|_S \subseteq |\phi|_S, \quad (23)$$

and a rule

$$\left(\bigwedge_{a' \in M_u} (a', a'(u)) \right) \Rightarrow (a, v) \quad (24)$$

is true in S . Let us now consider the set $P_u^a = \{a(u') : u' \in U \text{ and } \forall_{a' \in M_u} a'(u') = a'(u)\}$. Since

$$\left| \bigwedge_{a' \in M_u} (a', a'(u)) \right|_S \subseteq |\phi|_S \quad (25)$$

and

$$\text{card}(\{a(u') : u' \in |\phi|_S\}) = 1 \quad (26)$$

we also have that $\text{card}(P_u^a) = 1$.

We immediately obtain the following corollary.

Corollary 1. *The object u^* satisfies all rules from $Rul(S)$ which are supported by an object $u \in U$ if and only if $\text{card}(P_u^a) \geq 2$ for each $a \in A - M_u$.*

Example 1. Now, we give a simple example enabling readers to understand the approach proposed in this section. Let us consider an information system $S = (U, A)$ shown in Table IIa. Formally, for S we have: the set of objects $U = \{u_1, u_2, \dots, u_{11}\}$, the set of attributes $A = \{a_1, a_2, a_3\}$, the sets of attribute values $V_{a_1} = V_{a_2} = V_{a_3} = \{-1, 0, 1\}$. Let us assume that we have obtained new objects (shown in Table IIb) and we add them to our system. We are going to compute consistency of these objects with the knowledge included in the original system S using approach presented in this section. This consistency will be expressed for the new objects by consistency factors

Algorithm 1. Algorithm for an efficient computing a consistency factor of an object belonging to the extension of an information system

Input : An information system $S = (U, A)$, an object u^* belonging to the extension of S .

Output: A consistency factor $\xi_S(u^*)$ of the object u^* with the knowledge included in S .

```

 $\tilde{U} \leftarrow \emptyset;$ 
 $S_{orig} = (U_{orig}, A) \leftarrow S = (U, A);$ 
for each  $u \in U$  do
  for each  $a \in A$  do
    if  $a(u) \neq a(u^*)$  then
      |  $a(u) \leftarrow *$ ;
    end
  end
end

Remove each object  $u \in U$  such that  $\forall_{a \in C} a(u) = *$ ;

for each  $u \in U$  do
   $M_u \leftarrow \{a \in C : a(u) \neq *\};$ 
  for each  $d \in A - M_u$  do
     $P_u^d \leftarrow \{d(u') : u' \in U \text{ and } \forall_{a' \in M_u} a'(u') = a'(u)\}$ , where  $d(u')$  is
    determined on the basis of  $S_{orig}$ ;
    if  $card(P_u^d) = 1$  then
      |  $\tilde{U} \leftarrow \tilde{U} \cup \{u\}$ ;
      | break;
    end
  end
end

 $\xi_S(u^*) \leftarrow 1 - \frac{card(\tilde{U})}{card(U)};$ 

```

computed according to Algorithm 1. Information systems S' with irrelevant values of attributes for each new object are shown in Table 2

The computed sets P_u^a are shown in Table 3. We can see that for each set P_u^a in the case of the new object u_{12} we have $card(P_u^a) \geq 2$. According to Theorem 2 we obtain that the object u_{12} satisfies all minimal rules from $Rul(S)$. Hence, we have that the object u_{12} is consistent to the degree 1 (or consistent in short) with the knowledge included in the original system S . In the case of the new object u_{13} , for some sets P_u^a we have $card(P_u^a) = 1$. So, this object does not satisfy all rules from $Rul(S)$. The set $Rul_{u_{13}}(S)$ of rules not satisfied by u_{13} is supported by objects u_2, u_{10} and u_{11} . For the set $Rul_{u_{13}}(S)$ we have $str(Rul_{u_{13}}(S)) = 0.2727$. Therefore, $\xi_S(u_{13}) = 0.7273$. According to our approach we can say that the object u_{13} is consistent to the degree 0.7273 with the knowledge included in the original information system S . It is easy to see that the information system $S^* = (U^*, A)$, where $U^* = \{u_1, u_2, \dots, u_{13}\}$ is a partially consistent extension of the information system S , because for u_{13} we have $\xi_S(u_{13}) < 1$.

Table 1. a) an original information system S , b) new objects added to S

U/A	a_1	a_2	a_3
u_1	-1	1	0
u_2	0	-1	1
u_3	0	1	-1
u_4	0	0	0
u_5	0	1	0
u_6	0	1	0
u_7	-1	0	-1
u_8	-1	-1	0
u_9	-1	0	1
u_{10}	1	0	-1
u_{11}	1	0	0

U/A	a_1	a_2	a_3
u_{12}	-1	0	0
u_{13}	1	-1	1

Table 2. Information systems S' : (a) for the object u_{12} , (b) for the object u_{13}

U'/A	a_1	a_2	a_3
u'_1	-1	*	0
u'_2	*	*	*
u'_3	*	*	*
u'_4	*	0	0
u'_5	*	*	0
u'_6	*	*	0
u'_7	-1	0	*
u'_8	-1	*	0
u'_9	-1	0	*
u'_{10}	*	0	*
u'_{11}	*	0	0

U'/A	a_1	a_2	a_3
u'_1	*	*	*
u'_2	*	-1	1
u'_3	*	*	*
u'_4	*	*	*
u'_5	*	*	*
u'_6	*	*	*
u'_7	*	*	*
u'_8	*	-1	*
u'_9	*	*	1
u'_{10}	1	*	*
u'_{11}	1	*	*

Table 3. Sets P_u^a : (a) for the system corresponding to the new object u_{12} , (b) for the system corresponding to the new object u_{13}

U'/A	a_1	a_2	a_3
u'_1	*	$\{-1, 1\}$	*
u'_4	$\{0, 1\}$	*	*
u'_5	$\{-1, 0, 1\}$	$\{-1, 0, 1\}$	*
u'_6	$\{-1, 0, 1\}$	$\{-1, 0, 1\}$	*
u'_7	*	*	$\{-1, 1\}$
u'_8	*	$\{-1, 1\}$	*
u'_9	*	*	$\{-1, 1\}$
u'_{10}	$\{-1, 0, 1\}$	*	$\{-1, 0, 1\}$
u'_{11}	$\{0, 1\}$	*	*

U'/A	a_1	a_2	a_3
u'_2	$\{0\}$	*	*
u'_8	$\{-1, 0\}$	*	$\{0, 1\}$
u'_9	$\{-1, 0\}$	$\{-1, 0\}$	*
u'_{10}	*	$\{0\}$	$\{-1, 0\}$
u'_{11}	*	$\{0\}$	$\{-1, 0\}$

Table 4. An information system S describing weather processes

U/A	T	H	P	W
d_1	25	40	h	y
d_2	30	40	l	n
d_3	15	50	l	n
d_4	30	80	l	y
d_5	25	80	l	y

Table 5. Exemplary minimal rules true and realizable in the information system S

$(P, h) \Rightarrow (T, 25)$	$(W, y) \wedge (H, 40) \Rightarrow (T, 25)$
$(W, n) \wedge (T, 30) \Rightarrow (H, 40)$	$(T, 30) \Rightarrow (P, l)$

5.3 Application: Possibility Distribution of States

If we take a Cartesian extension $S^{max} = (U^{max}, A)$ of a given information system $S = (U, A)$, then we can calculate a consistency factor $\xi_S(u)$ for each object $u \in U^{max}$. Since $\xi_S(u) \in [0, 1]$, we obtain a fuzzy set Ξ defined over a set U^{max} of all objects from the Cartesian extension of S . The set U^{max} is the universe of discourse. The fuzzy set Ξ has the form:

$$\Xi = \frac{\xi_S(u_1)}{u_1} + \frac{\xi_S(u_2)}{u_2} + \dots + \frac{\xi_S(u_n)}{u_n}, \quad (27)$$

where $u_1, u_2, \dots, u_n \in U^{max}$, and $\xi(u_i)$ is a consistency factor of u_i with the knowledge included in S for each $i = 1, 2, \dots, n$.

If we have an information system $S = (U, A)$ describing a system of processes \mathbb{S} , it means we have collected all global states observed in \mathbb{S} until now. Let us assume that all possible local states of processes have been observed for each process of \mathbb{S} , but, in many cases, we have observed only a part of all possible combinations of local states (a part of all possible global states) in \mathbb{S} . In this case, if we take a new global state u_{new} which has not been observed in \mathbb{S} yet, then the following question Q arises: "Is it possible that the global state u_{new} will appear in the system \mathbb{S} in the future?". In this question, "possible" means "plausible". We wish to answer the question Q on the basis of the possessed information collected until now in the information system S . If we determine a Cartesian extension $S^{max} = (U^{max}, A)$ of S then for a state variable u ranging on U^{max} we can determine a possibility distribution $\pi_u(u_i)$ such that $\pi_u(u_i) = \xi_S(u_i)$, where $u_i \in U^{max}$. The state variable u denotes a global state appearing in the system \mathbb{S} in the future. The fuzzy set Ξ , determined earlier, represents the set of possible values of "u=a global state from U^{max} ". The quantity $\pi_u(u_i)$ represents the degree of possibility of the assignment $u = u_i$. The function π_u represents a flexible restrictions of the values of u with the following conventions:

Table 6. A Cartesian extension S^{max} of S with a possibility distribution of states

U^{max}/A	State status	T	H	P	W	$\pi_u(u_i)$
u_1	original	25	40	h	y	1.0
u_2	original	30	40	l	n	1.0
u_3	original	15	50	l	n	1.0
u_4	original	30	80	l	y	1.0
u_5	original	25	80	l	y	1.0
u_6	new	15	80	l	y	0.8
u_7	new	15	50	h	y	0.6
u_8	new	15	40	h	y	0.6
u_9	new	25	50	h	y	0.6
u_{10}	new	15	40	l	n	0.6
u_{11}	new	30	50	l	n	0.6
u_{12}	new	15	40	h	n	0.4
u_{13}	new	15	50	l	y	0.4
u_{14}	new	25	40	l	n	0.4
u_{15}	new	25	50	l	n	0.4
u_{16}	new	25	50	l	y	0.4
u_{17}	new	25	80	h	y	0.4
u_{18}	new	25	80	l	n	0.4
u_{19}	new	30	40	h	y	0.4
u_{20}	new	15	80	l	n	0.4
u_{21}	new	30	50	l	y	0.4
u_{22}	new	30	80	l	n	0.4
u_{23}	new	15	50	h	n	0.4
u_{24}	new	15	80	h	y	0.2
u_{25}	new	30	40	h	n	0.2
u_{26}	new	25	40	h	n	0.2
u_{27}	new	30	40	l	y	0.2
u_{28}	new	30	50	h	n	0.2
u_{29}	new	30	50	h	y	0.2
u_{30}	new	25	40	l	y	0.2
u_{31}	new	30	80	h	y	0.2
u_{32}	new	25	50	h	n	0.2
u_{33}	new	30	80	h	n	0.0
u_{34}	new	15	40	l	y	0.0
u_{35}	new	25	80	h	n	0.0
u_{36}	new	15	80	h	n	0.0

- $\pi_u(u_i) = 0$ means that $u = u_i$ is rejected as impossible,
- $\pi_u(u_i) = 1$ means that $u = u_i$ is totally possible (totally plausible).

For each global state (object) $u_i \in U$, we have $\xi_S(u_i) = 1$, thus $\pi_u(u_i) = 1$. This means that "a global state appearing in the future is u_i " is totally possible. It seems to be obvious because in the set U we have collected global states which have been observed until now in the system \mathbb{S} . Hence,

their appearing in the future is totally possible. For the remaining global states, i.e., from $U^{max} - U$ we do not have such certainty (such global states have not been observed yet). Nevertheless, on the basis of the possessed information, we can evaluate the possibility of appearing such states in the system \mathbb{S} in the future. To this end, we can determine consistency factors of new objects from the Cartesian extension of the original information system S .

Example 2. Let us consider an information system describing weather processes: temperature (marked with T), humidity (marked with H), pressure (marked with P), and wind speed (marked with W). Global states observed in our system are collected in Table 4 representing an information system $S = (U, A)$, for which:

- a set of objects (global states) $U = \{u_1, u_2, \dots, u_5\}$,
- a set of attributes (processes) $A = \{T, H, P, W\}$,
- sets of attribute values (local states of processes): $V_T = \{15, 25, 30\}$ [C], $V_H = \{40, 50, 80\}$ [%], $V_P = \{h, l\}$, where h denotes "high" and l denotes "low", $V_W = \{y, n\}$, where y denotes "yes" and "n" denotes "no".

In Table 5, exemplary minimal rules true and realizable in the information system S describing weather processes are shown.

A Cartesian extension $S^{max} = (U^{max}, A)$ of the information system S consists of 36 objects. For each object $u \in U^{max}$ we can calculate a consistency factor $\xi_S(u)$ of u with the knowledge included in S and expressed by all minimal rules true and realizable in S . Consistency factors determine a possibility distribution $\pi_u(u_i)$ such that $\pi_u(u_i) = \xi_S(u_i)$, where $u_i \in U^{max}$. Results are collected in Table 6.

6 Extensions of Dynamic Information Systems

Analogously to extensions of information systems, we can talk about extensions of dynamic information systems. Such extensions have been considered in [12]. An extension DS^* of a given dynamic information system DS is created by adding to its underlying system S new global states (objects) whose signatures contain only values of attributes which appeared in S and moreover by adding to the transition system TS new transitions between global states from the extension of S . Here, a consistent extension of a dynamic information system is important. Such an extension is obtained when each new global state added to S satisfies each minimal underlying rule true and realizable in S and each new transition added to TS satisfies each minimal transition rule true and realizable in a decision transition system created for TS . If a dynamic information system DS describes a system \mathbb{S} of processes, then the maximal consistent extension DS^* of DS represents all the global states of \mathbb{S} which are consistent with all the rules representing dependencies among local states of processes, extracted from the underlying system S and

all the transitions between global states of \mathbb{S} which are consistent with all the transition rules extracted from a decision transition system representing a transition relation in DS . In some cases, the maximal consistent extension can contain new global states of \mathbb{S} and new transitions between global states which have not been observed yet, but which are consistent with rules extracted from DS .

6.1 Basic Definitions

In this subsection, we give definitions concerning extensions of dynamic information systems. Analogously to extensions of information systems we can define consistent and partially consistent extensions of dynamic information systems and adequate consistency factors for global states and transition between them.

Definition 25 (Extension of a transition system). Let $TS = (S, E, T)$ be a transition system. A transition system $TS^* = (S^*, E^*, T^*)$ is an extension of TS if and only if the following requirements are satisfied:

- $S \subseteq S^*$,
- $E \subseteq E^*$,
- $T^*|_{U \times E \times U} = T$.

Definition 26 (Extension of a dynamic information system). Let $DS = (U, A, E, T)$ be a dynamic information system, $S = (U, A)$ its underlying system, and $TS = (U, E, T)$ its transition system. A dynamic information system $DS^* = (U^*, A^*, E^*, T^*)$ is an extension of DS if and only if the following requirements are satisfied:

- an underlying system $S^* = (U^*, A^*)$ is an extension of S ,
- a transition system $TS^* = (U^*, E^*, T^*)$ is an extension of TS .

Remark 12. A set A^* of attributes in an extension $DS^* = (U^*, A^*, E^*, T^*)$ of a dynamic information system $DS = (U, A, E, T)$ will be denoted by A like in the original system DS . So, we write $DS^* = (U^*, A, E^*, T^*)$ instead of $DS^* = (U^*, A^*, E^*, T^*)$. The same concerns attributes of A^* , i.e., $a_1^*, a_2^*, \dots, a_m^* \in A^*$. So, we write a_1, a_2, \dots, a_m instead of $a_1^*, a_2^*, \dots, a_m^*$, where $a_1, a_2, \dots, a_m \in A$.

Let $DS = (U, A, E, T)$ be a dynamic information system with a decision transition system $S_T = (U_T, A \cup A')$, $DS^* = (U^*, A, E^*, T^*)$ an extension of DS with a decision transition system $S_T^* = (U_T^*, A \cup A')$, and $t \in U_T^*$. By $TranRul_t^\sim(DS)$ we denote a set of all minimal transition rules true and realizable in S_T which are not satisfied by the transition t , i.e.

$$TranRul_t^\sim(DS) = \{(\phi \Rightarrow \psi) \in TranRul(DS) : \text{not } t \models (\phi \Rightarrow \psi)\}. \quad (28)$$

Definition 27 (Consistency factor of a transition). Let $DS = (U, A, E, T)$ be a dynamic information system with a decision transition system $S_T = (U_T, A \cup A')$, $DS^* = (U^*, A, E^*, T^*)$ an extension of DS with a decision transition system $S_T^* = (U_T^*, A \cup A')$, and $t \in U_T^*$. A consistency factor of t with the knowledge included in S_T is a number defined as:

$$\xi_{S_T}(t) = 1 - str(TranRul_t^*(DS)). \quad (29)$$

A consistency factor satisfies inequalities $0 \leq \xi_{S_T}(t) \leq 1$ for each $t \in U_T^*$. Obviously, if $t \in U_T$, then $\xi_{S_T}(t) = 1$, because $TranRul_t^*(DS) = \emptyset$.

Definition 28 (Consistent extension of a dynamic information system). Let $DS = (U, A, E, T)$ be a dynamic information system represented by an underlying system $S = (U, A)$ and a decision transition system $S_T = (U_T, A \cup A')$. Let $DS^* = (U^*, A, E^*, T^*)$ be an extension of DS represented by an underlying system $S^* = (U^*, A)$, and a decision transition system $S_T^* = (U_T^*, A \cup A')$. DS^* is a consistent extension of DS if and only if the following requirements are satisfied:

- $\xi_S(u) = 1$ for each $u \in U^*$,
- U_T^* includes only objects representing transitions such that $(u, e, u') \in T^*$, $\xi_S(u) = 1$, $\xi_S(u') = 1$, and $e \in E^*$,
- $\xi_{S_T}(t) = 1$ for $t \in U_T^*$.

6.2 Computing Consistency Factors of Transitions between States

In this subsection, we are interested in computing a consistency factor of a transition from any extension of a transition system TS of a given dynamic information system DS with the knowledge included in DS . We give an efficient algorithm for computing consistency factors. This algorithm has a polynomial time complexity. An approach proposed here does not involve computing any transition rules from an original decision transition system S_T representing TS . At the beginning, we present a significant theorem allowing us to determine whether the new transition added to the original decision transition system S_T satisfies all minimal transition rules, true and realizable in S_T , without computing such rules. Here, the theorem taken from [1] is formulated for a decision transition system and expressed by means of the formalism used in the chapter.

Theorem 3. Let $S_T = (U_T, A \cup A')$ be a decision transition system for a dynamic information system DS , $S_T^* = (U_T^*, A \cup A')$ a decision transition system for an extension of DS , $TranRul(DS)$ a set of all minimal transition rules true and realizable in S_T and $t^* \in U_T^*$. For each $t \in U_T$ let $M_t = \{a \in A : a(t^*) = a(t)\}$ and $P_t^d = \{d(t') : t' \in U_T \text{ and } \forall_{a' \in M_t} a'(t') = a'(t)\}$ for each $d \in A'$. The transition t^* satisfies all rules from $TranRul(DS)$ if and only if for any $t \in U_T$ and $d \in A'$ one of the following requirements is satisfied:

Algorithm 2. Algorithm for an efficient computing a consistency factor of a transition belonging to the extension of a decision transition system

Input : A decision transition system $S_T = (U_T, A \cup A')$, a transition t^* belonging to the extension of S_T .

Output: A consistency factor $\xi_{S_T}(t^*)$ of the transition t^* with the knowledge included in S_T .

```

 $\tilde{U} \leftarrow \emptyset;$ 
for each  $u \in U$  do
  for each  $a \in A$  do
    if  $a(u) \neq a(u^*)$  then
       $| a(u) \leftarrow *;$ 
    end
  end
end

Remove each object  $u \in U$  such that  $\forall_{a \in A} a(u) = *$ ;

for each  $u \in U$  do
   $M_u \leftarrow \{a \in A : a(u) \neq *\};$ 
  for each  $d \in A'$  do
     $P_u^d \leftarrow \{d(u') : u' \in U \text{ and } \forall_{a' \in M_u} a'(u') = a'(u)\};$ 
    if  $\text{card}(P_u^d) = 1 \text{ and } d(u^*) \neq d(u)$  then
       $| \tilde{U} \leftarrow \tilde{U} \cup \{u\};$ 
      break;
    end
  end
end

 $\xi_{S_T}(u^*) \leftarrow 1 - \frac{\text{card}(\tilde{U})}{\text{card}(U)};$ 
```

1. $\text{card}(P_t^d) \geq 2$,
2. $\text{card}(P_t^d) = 1 \text{ and } d(t^*) = d(t)$.

The proof of Theorem 3 is analogous to the proof of Theorem 2.

Example 3. Now, we give a simple example enabling readers to understand the approach proposed in this section. Let us consider a decision transition system S_T shown in Table 7.

Let us have a new transition t_{new} shown in Table 8. We are going to compute consistency of this transition with the knowledge included in the original decision transition system S_T using approach presented in this section. This consistency will be expressed for the new transition by a consistency factor computed according to Algorithm 2.

Decision transition system S'_T with irrelevant values of attributes is shown in Table 9.

The computed sets P_t^d are shown in Table 10.

According to Theorem 3 there does not exist any transition supporting minimal transition rules true and realizable in S_T which are not satisfied by the new transition t_{new} .

Table 7. A decision transition system S_T

$U_T/A \cup A'$	a_1	a_2	a'_1	a'_2
t_1	-1	-1	-1	-1
t_2	-1	-1	1	1
t_3	1	1	1	-1
t_4	1	-1	-1	1
t_5	-1	1	0	0
t_6	-1	1	0	0
t_7	-1	1	-1	0
t_8	0	0	-1	0
t_9	0	1	-1	0
t_{10}	-1	0	-1	0

Table 8. A new transition

$U_T/A \cup A'$	a_1	a_2	a'_1	a'_2
t_{new}	-1	-1	-1	0

Table 9. A decision transition system S'_T

$U_T/A \cup A'$	a_1	a_2	a'_1	a'_2
t_1	-1	-1	-1	-1
t_2	-1	-1	1	1
t_3	*	*	1	-1
t_4	*	-1	-1	1
t_5	-1	*	0	0
t_6	-1	*	0	0
t_7	-1	*	-1	0
t_8	*	*	-1	0
t_9	*	*	-1	0
t_{10}	-1	*	-1	0

Table 10. Sets P_t^d

$U_T/A \cup A'$	a_1	a_2	$P_u^{a'_1}$	$P_u^{a'_2}$
t_1	-1	-1	$\{-1, 1\}$	$\{-1, 1\}$
t_2	-1	-1	$\{-1, 1\}$	$\{-1, 1\}$
t_4	*	-1	$\{-1, 1\}$	$\{-1, 1\}$
t_5	-1	*	$\{-1, 0\}$	$\{0\}$
t_6	-1	*	$\{-1, 0\}$	$\{0\}$
t_7	-1	*	$\{-1, 0\}$	$\{0\}$
t_{10}	-1	*	$\{-1, 0\}$	$\{0\}$

For each transition t , we have $\text{card}(P_t^{a'_1}) \geq 2$ or $\text{card}(P_t^{a'_2}) \geq 2$ or $\text{card}(P_t^{a'_3}) = 1$ and $a'_2(t) = a'_2(t_{\text{new}})$. The set $\text{TranRul}_{t_{\text{new}}}^{\sim}(DS)$ of transition rules not satisfied by t_{new} is not supported by any transition. For the set $\text{TranRul}_{t_{\text{new}}}^{\sim}(DS)$, we have $\text{str}(\text{TranRul}_{t_{\text{new}}}^{\sim}(DS)) = 0$. Therefore, $\xi_{S_T}(t_{\text{new}}) = 1$. According to our approach we can say that the transition t_{new} is consistent to the degree 1 with the knowledge included in the original decision transition system S_T .

6.3 Application: Possibility Distribution of Transitions between States

If we take into consideration a maximal extension of a given dynamic information system DS , we can calculate a consistency factor $\xi_S(u)$ for each object belonging to the extension of an underlying system of DS . If a dynamic information system describes a system of processes \mathbb{S} , we determine a possibility distribution of states in \mathbb{S} . Analogously to possibility distribution of states, we can determine a possibility distribution of transitions between states in \mathbb{S} . For each transition t belonging to the extension of a transition system TS of DS , we can calculate a consistency factor $\xi_{S_T}(t)$ of t with the knowledge included in a decision transition system S_T representing the transition system TS of DS . This knowledge is expressed by means of a set of all minimal transition rules true and realizable in S_T . Since $\xi_{S_T}(t) \in [0, 1]$, we obtain a fuzzy set Ξ_T defined over a set U_T^{\max} , where U_T^{\max} is a decision transition system for the extension of DS . The set U_T^{\max} is the universe of discourse in the case of possible transitions between states. The fuzzy set Ξ_T has the form:

$$\Xi_T = \frac{\xi_{S_T}(t_1)}{t_1} + \frac{\xi_{S_T}(t_2)}{t_2} + \dots + \frac{\xi_{S_T}(t_q)}{t_q}, \quad (30)$$

where $t_1, t_2, \dots, t_q \in U_T^{\max}$, and $\xi(t_i)$ is a consistency factor of t_i with the knowledge included in S_T for each $i = 1, 2, \dots, q$.

Now, we can carry out reasoning analogous to that in Subsection 5.3. A decision transition system $S_T = (U_T, A \cup A')$ includes all transitions between global states observed in \mathbb{S} until now. In many cases, we have observed only a part of all possible transitions. If we take a new transition t_{new} which

Table 11. A decision transition system S_T describing transitions between states in a system of weather processes

$U_T/A \cup A'$	T	H	P	W	T'	H'	P'	W'
t_1	25	40	h	y	30	40	1	n
t_2	30	40	l	n	15	50	1	n
t_3	15	50	l	n	30	80	1	y
t_4	30	80	l	y	25	80	1	y

Table 12. Exemplary minimal transition rules true and realizable in the decision transition system S_T

$(T, 25) \Rightarrow (T', 30)$	$(W, y) \wedge (H, 40) \Rightarrow (T', 30)$
$(W, y) \wedge (T, 30) \Rightarrow (W', y)$	$(H, 80) \Rightarrow (T', 25)$

Table 13. An extension S_T^* of S_T with a possibility distribution of transitions

$U_T^*, A \cup A'$	Transition status	T	H	P	W	T'	H'	P'	W'	$\pi_t(t_i)$
t_1	original	25	40	h	y	30	40	l	n	1.00
t_2	original	30	40	l	n	15	50	l	n	1.00
t_3	original	15	50	l	n	30	80	l	y	1.00
t_4	original	30	80	l	y	25	80	l	y	1.00
t_5	new	30	40	l	n	30	40	l	n	0.75
t_6	new	15	50	l	n	30	40	l	n	0.75
t_7	new	15	50	l	n	15	50	l	n	0.75
t_8	new	25	40	h	y	15	50	l	n	0.75
t_9	new	15	50	l	n	25	80	l	y	0.75
t_{10}	new	30	80	l	y	30	40	l	n	0.75
t_{11}	new	30	80	l	y	15	50	l	n	0.75
t_{12}	new	30	80	l	y	30	80	l	y	0.75
t_{13}	new	25	80	l	y	30	40	l	n	0.75
t_{14}	new	25	80	l	y	25	80	l	y	0.75
t_{15}	new	30	40	l	n	30	80	l	y	0.50
t_{16}	new	30	40	l	n	25	80	l	y	0.50
t_{17}	new	25	40	h	y	30	80	l	y	0.50
t_{18}	new	25	40	h	y	25	80	l	y	0.50
t_{19}	new	25	80	l	y	15	50	l	n	0.50
t_{20}	new	25	80	l	y	30	80	l	y	0.50
t_{21}	new	15	50	l	n	25	40	h	y	0.25
t_{22}	new	25	40	h	y	25	40	h	y	0.25
t_{23}	new	30	40	l	n	25	40	h	y	0.00
t_{24}	new	30	80	l	y	25	40	h	y	0.00
t_{25}	new	25	80	l	y	25	40	h	y	0.00

has not been observed yet in \mathbb{S} , then the following question Q arises: "Is it possible that the transition t_{new} will appear in the system \mathbb{S} in the future?". We wish to answer the question Q on the basis of the possessed information collected until now in the decision transition system S_T . We can determine a possibility distribution $\pi_t(t_i)$ such that $\pi_t(t_i) = \xi_{S_T}(t_i)$, where $t_i \in U_T^{max}$, for a state variable t ranging on U_T^{max} . The state variable t denotes a transition between global states appearing in the system \mathbb{S} in the future. The fuzzy set Ξ_T , determined earlier, represents the set of possible values of "t=a transition between global states from U^{max} ". The quantity $\pi_t(t_i)$ represents the degree of possibility of the assignment $t = t_i$. For each transition $t_i \in U_T$, we have $\xi_{S_T}(t_i) = 1$, thus $\pi_t(t_i) = 1$. This means that "a transition appearing in the

future is t_i " is totally possible. It seems to be obvious because in the set U_T we have collected transitions which have been observed until now in the system \mathbb{S} . Hence, their appearing in the future is totally possible. For the remaining transitions, i.e., from $U_T^{max} - U_T$, we do not have such certainty (such transitions have not been observed yet). Nevertheless, on the basis of the possessed information, we can evaluate the possibility of appearing such transitions in the system \mathbb{S} in the future.

Example 4. Let us consider an information system S describing weather processes from Example 1. If we assume that objects in S are ordered in time, then we obtain the following transition relation T :

$$T = \{(u_1, e_1, u_2), (u_2, e_2, u_3), (u_3, e_3, u_4), (u_4, e_4, u_5)\}. \quad (31)$$

We can construct a decision transition system $S_T = (U_T, A \cup A')$ shown in Table I1. We have four transitions t_1, t_2, t_3, t_4 in S_T .

In Table I2, exemplary minimal transition rules true and realizable in the decision transition system S_T describing weather processes are shown.

An extension $S_T^* = (U_T^*, A \cup A')$ of the decision transition system S_T consists of 25 transitions (all possible transitions between states collected in the underlying system S). For each transition $t \in U_T^*$ we can calculate a consistency factor $\xi_{S_T}(t)$ of t with the knowledge included in S_T and expressed by all minimal transition rules true and realizable in S_T . Consistency factors determine a possibility distribution $\pi_t(t_i)$ such that $\pi_t(t_i) = \xi_{S_T}(t_i)$, where $t_i \in U_T^*$. Results are collected in Table I3.

7 Conclusions

In the chapter, we discussed some issues on extensions of information and dynamic information systems. Consistent and partially consistent extensions of information and dynamic information systems are helpful in prediction problems. On the basis of those extensions we can determine possibility distributions of states and transitions between states over a universe of discourse related to a given system of processes.

Acknowledgments

The work described here has been partially supported by the grant from the University of Information Technology and Management in Rzeszów, Poland.

References

1. Moshkov, M., Skowron, A., Suraj, Z.: On Testing Membership to Maximal Consistent Extensions of Information Systems. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS (LNAI), vol. 4259, pp. 85–90. Springer, Heidelberg (2006)

2. Moshkov, M., Skowron, A., Suraj, Z.: Maximal Consistent Extensions of Information Systems Relative to Their Theories. *Information Science* 178, 2600–2620 (2008)
3. Pancerz, K., Suraj, Z.: Synthesis of Petri Net Models: A Rough Set Approach. *Fundamenta Informaticae* 55, 149–165 (2003)
4. Pancerz, K.: Consistency-Based Prediction Using Extensions of Information Systems - an Experimental Study. In: Proceedings of the Conference HSI 2008, IEEE Catalog Number: 08EX19995C (2008)
5. Pancerz, K.: Extensions of Dynamic Information Systems in State Prediction Problems: the First Study. In: Magdalena, L., Ojeda-Aciego, M., Verdegay, J.L. (eds.) *Proceedings of the Conference IIPMU 2008*, pp. 101–108 (2008)
6. Pawlak, Z.: *Rough Sets - Theoretical Aspects of Reasoning About Data*. Kluwer Academic Publishers, Dordrecht (1991)
7. Pawlak, Z.: Concurrent Versus Sequential the Rough Sets Perspective. *Bulletin of the EATCS* 48, 178–190 (1992)
8. Rząsa, W., Suraj, Z.: A New Method for Determining of Extensions and Restrictions of Information Systems. In: Alpigini, J.J., Peters, J.F., Skowron, A., Zhong, N. (eds.) *RSCTC 2002. LNCS (LNAI)*, vol. 2475, pp. 197–204. Springer, Heidelberg (2002)
9. Skowron, A., Suraj, Z.: Rough Sets and Concurrency. *Bulletin of the Polish Academy of Sciences* 41, 237–254 (1993)
10. Skowron, A., Suraj, Z.: Synthesis of Concurrent Systems Specified by Information Systems. *Institute of Computer Science Research Report* 39/94, Warsaw University of Technology, Poland (1994)
11. Suraj, Z.: Discovery of Concurrent Data Models from Experimental Tables: A Rough Set Approach. *Fundamenta Informaticae* 28, 353–376 (1996)
12. Suraj, Z.: The Synthesis Problem of Concurrent Systems Specified by Dynamic Information Systems. In: Polkowski, L., Skowron, A. (eds.) *Rough Sets in Knowledge Discovery*, vol. 2, pp. 418–448. Physica-Verlag, Berlin (1998)
13. Suraj, Z.: Rough Set Methods for the Synthesis and Analysis of Concurrent Processes. In: Polkowski, L., Tsumoto, S., Lin, T.Y. (eds.) *Rough Set Methods and Applications*, pp. 379–488. Physica-Verlag, Berlin (2000)
14. Suraj, Z.: Some Remarks on Extensions and Restrictions of Information Systems. In: Ziarko, W.P., Yao, Y. (eds.) *RSCTC 2000. LNCS*, vol. 2005, pp. 204–211. Springer, Heidelberg (2001)
15. Suraj, Z., Pancerz, K., Owsiany, G.: On Consistent and Partially Consistent Extensions of Information Systems. In: Ślezak, D., Wang, G., Szczuka, M.S., Düntsch, I., Yao, Y. (eds.) *RSFDGrC 2005. LNCS (LNAI)*, vol. 3641, pp. 224–233. Springer, Heidelberg (2005)
16. Suraj, Z., Pancerz, K.: Some Remarks on Computing Consistent Extensions of Dynamic Information Systems. In: Kwaśnicka, H., Paprzycki, M. (eds.) *Proceedings of the Conference ISDA 2005*, pp. 420–425. IEEE Computer Society, Los Alamitos (2005)
17. Suraj, Z., Pancerz, K.: A New Method for Computing Partially Consistent Extensions of Information Systems: A Rough Set Approach. In: *Proceedings of the Conference IIPMU 2006*, Editions EDK, Paris, pp. 2618–2625 (2006)
18. Suraj, Z., Pancerz, K.: Towards Efficient Computing Consistent and Partially Consistent Extensions of Information Systems. *Fundamenta Informaticae* 79, 553–566 (2007)
19. Zadeh, L.: Fuzzy Sets as the Basis for a Theory of Possibility. *Fuzzy Sets and Systems* 1, 3–28 (1978)

A Probabilistic Approach to the Evaluation and Combination of Preferences

Annibal Parracho Sant'Anna

Universidade Federal Fluminense - Niteroi-RJ - Brazil
tppaps@vm.uff.br

Abstract. A model for the process of evaluating and combining preferences is here proposed. After determining the preferences according to each criterion, these partial preferences are combined into global ones. Different combination rules are set, in a probabilistic framework. Attention is centered to the case of indicators measured in different levels of aggregation.

1 Introduction

This chapter deals with the problem of combining indicators of preference. Attention is centered in the case of indicators measured in different levels of aggregation of the evaluated options. This happens when the same criteria are applied to evaluate in an individual basis and in clusters of homogeneous individuals. It is demonstrated along the chapter that, by means of the determination of the preferences in terms of probabilities of being the preferred option, a probabilistic composition of the indicators may be performed in such a way that the correlations between such indicators may be taken into account.

Several difficulties have always been recognized in the combination of multiple criteria. The invertibility of evaluations on different measurement scales, the lack of precision in the observed values, the dependence between the criteria generate the most important of such difficulties. Notwithstanding, since the need to performance monitoring by numerical indices is increasing, the combination of criteria to rank options with distinct features has been always more employed.

Historically, the importance of the invertibility of evaluations increases as these evaluations extend from the economic sector, where the convertibility effort is most of the time limited to set all values in monetary terms, to other areas where social, environmental and other values are not suitable to conversion to monetary representation.

The difficulty to deal with criteria evaluated in invertible reference settings is clear when the combination involves weighting the criteria. The need to take into account the scale of measurement when setting the weights has been signaled long ago.

The imprecision of the evaluations and the need to take into account uncertainty and derive probabilistic conclusions are other important points of criticism to combined criteria evaluations. In fact there is always some amount of subjectivity in deriving preference for any kind of attribute and the lack of information on the degree of uncertainty in uncertain classifications considerably impairs the use of such classifications.

The transformation into probabilities of being ranked first, proposed in [11], opens a way to overcome such difficulties. It starts by ranking according to the different particular criteria to be combined. The imprecision in such rankings is modeled by considering the assigned ranks as midpoints of statistical distributions. These distributions are, in practice, determined by adding stochastic disturbances to such midpoints. Finally probabilities of attaining the first position are computed. These probabilities can be combined into global measures without the need of assigning weights to the criteria.

This chapter develops decision support systems applying the probabilistic composition to the situation where the individuals evaluated interact inside groups. The individual performances must be evaluated taking into account group features. This makes the composition object of another important source of criticism, due to the possible presence of unaccounted dependence between the variables combined. If the same attributes are measured in the analysis applied to evaluate the performance of individuals isolated and their performance as a group, the same disturbances must be present in the formation of the two evaluations. In the probabilistic approach, the correlation between the criteria may be directly taken into account by determining the global measurement in the form of joint probabilities.

Two examples of application are here considered. The first is to data about clients of a supermarkets chain. The other is about data of graduate courses.

The chapter is structured as follows. In the next Section, the transformation into probabilities of being the first and the different points of view that may be taken in combining such probabilities in global evaluations. Section 3 presents the problem of taking into account collective evaluations, the correlation between the vectors of evaluations that it engenders and the use of probabilistic Malmquist indices to measure the change of the performance through time in this context. Sections 4 and 5 develop the examples of application. Final comments conclude the chapter.

2 Probabilistic Composition of Preferences

The key computation in the evaluation of probabilistic preferences is the determination of the probabilities of each option being the preferred among those in a sample. The probability of considering a particular option as the best one is a natural measure of the decision maker preference for that option. But if we start measuring in another way, to be able to compare and compose the different criteria of preference, we ought to transform our

measurements into probabilities of being preferred. What cannot be measured cannot get managed, but numbers without a measure of their precision cannot be managed either.

From any kind of measurement a ranking of the options can be derived. On the other hand, a ranking with possible ties and empty spaces for presently not available options may represent every distance between the different options that one may have in mind.

After ranking, to determine the probabilities of preference, it rests quantifying the imprecision in the ranks. If the ranks are derived from preferences initially given in qualitative terms, i. e., in terms of common language, such as low, moderate or high preference, the imprecision is usually taken into account by means of the representation through fuzzy intervals ([16]). But it is also present in ordinal and cardinal scales and can be represented analogously.

To compute the probabilities of being the first option all we need, besides an ordering, is a statistical measure of the uncertainty on each position in that ordering. The modeling of the uncertainty may be always done in the measurement with error framework ([15]). In this framework, the rank of each option or any other numerical indication of its position is thought as the mean (or any other location parameter) of a statistical distribution. Once the observed ranks, presented at the beginning as deterministic, are seen as estimates of location parameters, estimates of other parameters can be derived from the same set of observations, although it is difficult to have a number of observations large enough to estimate higher moments with precision in the first applications.

The observed range, i.e., the range of observed values for the criterion in the whole set of examined options, may be used as an estimate for a common range for these distributions. Different assumptions on the form may be taken to complete their modeling. To compensate the lack of empirical information, simplifying and equalizing assumptions are the essence of Fuzzy Sets Theory ([16]). In the present situation, imposing analogously assumptions of independence between the disturbances affecting the different observations and, for the form, for instance, normal distributions with identical variance, or uniform distributions with identical range, or triangular distributions with the same set of possible values, will provide a precise framework.

2.1 Computation of the Probability of Being the Most Preferred Option

To make easier the comparisons, the probabilities of being the most preferred may be computed with respect to a sample of fixed size, randomly generated or withdraw in fixed percentiles of the set of values assigned to the options under evaluation. For instance, this sample may be formed by the nine deciles of the observed distribution. This sample size has the advantage of presenting evaluations always distributed around the value 0.1 that will be given to all options if they are indiscernible.

To set the dispersion parameter, maximum and minimum values for each distribution may be derived from the maximum and minimum values in the set of assigned values.

Let us consider, for instance, the case of only five options being compared and ranked in a Likert scale of five points, representing the five possible distinct evaluations by the numbers 1 to 5. The intermediate deciles will then be 1.5, 2.5, 3.5 and 4.5. The distribution centered in each of these values may be a triangular distribution with extreme values 0.5 and 5.5.

In general, with n options being evaluated by the ordered values $x_{(1)}, \dots, x_{(n)}$, the k -th decile will be represented in the sample by $d_k = x_{(i(k))}$ with $i(k)$ the integer closest to $kn/10$ or the arithmetic mean between $x_{(kn/10-1/2)}$ and $x_{(kn/10+1/2)}$.

If, a triangular distribution ([10]) with constant minimum and maximum, is assumed, as is usual in the fuzzy approach, robust estimates for these minimum and maximum will be given, respectively, by $x_{(1)} - (d_9 - d_1)/(n/(10/8))$ and $x_{(n)} + (d_9 - d_1)/(n/(10/8))$.

It may be assumed, instead of the asymmetric triangular distributions, normal distributions with standard deviation derived from the observed range, as in ([13]). In that case, we may follow the usual practice of deriving an estimate for the dispersion parameter of each disturbance from a measure of the dispersion in the observed sample. Given the small number of options usually evaluated, to follow again the common practice, we may use the sample range and derive an estimate for the standard deviation dividing it by the normal relative range ([9]).

This procedure may be set more formally. Denoting by $d_2(n)$ the normal relative range for samples of size n , if the vector of observations for the i -th preference attribute is (y_{i1}, \dots, y_{in}) , the whole randomization procedure consists in assuming, for all i and k , that the distribution of possible values for the i -th attribute as the k -th option is evaluated is normal with expected value given by y_{ik} and standard deviation given by $\max(y_{i1}, \dots, y_{in}) - \min(y_{i1}, \dots, y_{in})/d_2(n)$.

Another alternative would be to assume a uniform distribution with a fixed range derived in such a way as to allow for all inversions of ranks considered reasonable. This was done in [12]. What variance will be large enough to allow for inversion of all ranks? Fixing, for instance, at $1/n$ the probability of inversion of ranks between the units with the highest and the lowest observed values would complete the statistical modeling under such uniformity assumption.

The probabilities of an option being the first can be computed by integrating with respect to the joint density the probability of such option presenting a value better than that of each other option. To compute this probability we ought to divide the integration interval into sub-intervals bounded by the values in the sample.

Let us consider, for instance, the case of triangular distributions, centered at the observed values and with extremes fixed at $n/(10/8)$ of the absolute

distance between the first and the ninth decile of the observed distribution of the values assigned to n options according to a given criterion, X , and let us assume independence between the disturbances affecting the evaluations of different options according to that criterion. Then, the probability of being the most preferred, for an option whose evaluation is $x_{(i)}$, that ranked i -th in increasing order in the observed sample of evaluations according to the criterion X , will be obtained by adding integrals of terms of the form $\prod(1-(1-x)^2)/(1-a_p)\prod(x^2/a_q)$ where the first product is for $p < j$ and the second for $q > j$, p and q different from i , and for j varying from 0 to n , the number of observations in the sample.

The integration will be with respect to the density of $X_{(i)}$. This density is given, as a function of x , by $2x/x_{(i)}$ for $i > j$ and by $2(1-x)/(1-x_{(i)})$ for $i < j$.

It is also possible to increase or reduce the standard deviation of one or another measure to mirror a stronger or weaker certainty regarding the measures provided about better or worse known production units. Nevertheless, dispersion variations are in general difficult to quantify.

The independence between random errors on the measurements according to a single criterion is also a simplifying assumption. If the options are only ranked by pairwise comparison, it would be more reasonable to assume a negative correlation. To model that precisely, it would be enough to assume identical correlations and to derive this identical value from the fact that the sum of the ranks is a constant. The absolute values of this correlation would, however, quickly decrease as the number of units grows.

2.2 Different Approaches to the Combination of Probabilistic Evaluations

After computing the probabilities of being the preferred according to each criterion, is easy to combine them into a unique measure of global preference. A way to do that consists of treating these probabilities as conditional on the choice of the respective criterion and computing the total probability preference by adding the products of such conditional probabilities by the probabilities of choice of each criterion.

The difficulty in this approach is to determine the marginal probabilities of choice of each criterion. This is specially difficult if the criteria are correlated. But, if it is possible to rank the criteria and model the correlation between them, these probabilities of choice of each criterion may be computed in the same way the probabilities of preference according to each criterion are computed.

Another strategy to combine the probabilistic preference is in terms of joint preference according to the multiple criteria. In this approach, the dependence between the criteria may be directly taken into account.

Different joint probabilities may be employed, depending on the point of view adopted. The different points of view may be characterized in terms of choice between extreme positions in two basic orientation axes. These extreme positions are, in one axis, an optimistic versus a pessimistic position and, in the other, a progressive versus a conservative position.

In the progressive-conservative axis, the progressive evaluator looks after options that are the first in excellence, the conservative evaluator evaluates them by their ability of not minimizing the preference. The term 'conservative' in this terminology is related to the idea of avoiding losses, while the term 'progressive' is related to the idea of improving, of reaching higher patterns.

In the optimistic-pessimistic axis, the optimistic extreme consists of considering enough the satisfaction of only one criterion. All the criteria are taken into account, but the composition employs the connective 'or'. The joint probability computed is that of maximizing (in a progressive composition, or of not minimizing in a conservative one) the preference according to at least one of the multiple criteria.

On the opposite end, the pessimistic preference goes for options that satisfy every criterion. The connective is 'and'. The joint probability computed is that of maximizing (or not minimizing) simultaneously the preference according to all the criteria.

The terms optimistic and pessimistic are related to the idea of confiding that the most favorable or the less favorable criterion, respectively, will prevail.

By combining the positions in the extremes of these two axes, four different measures are generated. In the progressive and optimistic point of view, evaluating each option by its probability of being preferred by at least one criterion, denoting by P_{ik} the probability of the k-th option being the preferred according to the i-th criterion, if the criteria are independent, the final measure of efficiency of this option is given by $1 - \prod(1 - P_{ik})$, for i varying along all considered criteria. The computation for the other approaches is analogous.

If the criteria are divided into groups and different points of view are allowed in the computation of the joint probabilities within each group, the number of possibilities increases. A natural division of the criteria into groups is in a subset of criteria for which the optimum is large and another subset for which optimization means reduction. For instance, criteria of benefits and criteria of disadvantages, criteria related to the production of outputs and criteria related to the use of inputs, criteria related to outcomes and criteria related to costs, and so on. Other measures may also be considered.

The criteria may be divided also into a larger number of independent subsets. It may be, for instance, the case of the global evaluation being given by the probability of presenting the best performance in at least one criterion of each block.

The choice of the composition algorithm will have to take into account practical considerations. There are situations, as in many segments of public administration, where the reference is placed in the worst possible performance. From such worst levels progress starts. In these cases most observed values, and the most reliable ones, are near the minimum efficiency level allowed. With only a few sparse values in the efficient extreme, it becomes safer to evaluate the performance by the probability of staying away from

the inefficiency border than by the proximity to the efficiency frontier. For instance, in such a situation, any value registered by mistake near the value of an option that is, in truth, the most efficient would strongly affect the probability of that best unit reaching the efficiency frontier while it will only slightly affect its probability of being the least preferred.

2.3 Relations between the Probabilistic Composition and Data Envelopment Analysis

Different approaches have been developed through time to compare production units according to their efficiency in extracting the largest possible aggregate of products from the smallest aggregate of employed resources. Data Envelopment Analysis (DEA) is an attractive instrument to reach this goal. It measures efficiency in a realistic way by the distance to the best observed performance and, to take into account that each unit may address a proper market niche, aggregation is performed using for each unit evaluated the most favorable weights.

The typical DEA efficiency evaluation algorithm was developed by [3]. The concept of efficiency applied comes from [5].

The composition in terms of probability of being the first option has in common with Data envelopment Analysis (DEA) the feature of deriving the evaluations from distances to the frontier. The use of DEA in multiple criteria composition may follow the productivity approach of DEA, by first identifying inputs and outputs and then constructing an aggregated index using the common DEA procedure. This corresponds, in the probabilistic composition, to divide the criteria into two blocks, one referring to the frontier of large values and the other to that of small values.

But DEA may also be applied with all the criteria in the same direction, as benefit or cost variables, the criteria aggregation being done in this case by a DEA constant inputs or constant outputs model.

In one or the other case, DEA ranks the units in terms of productivity comparison with a virtual reference option in the frontier. The composition based on the probabilities of reaching the frontier is more robust because it involves comparison with all evaluation units, not only those in the frontier.

From the points of view that may be chosen to combine the probabilistic evaluations, the optimistic and progressive point of view is closer to the DEA point of view. If this is the approach chosen, DEA algorithms may also be employed to combine the partial probabilistic evaluations in a final aggregate value.

Difficulties in the interpretation of the results obtained in some practical situations induced the development of alternative algorithms. A first variation was developed to deal with different scales of operation of the units put in parallel. There are situations in which operation units face orders with a volume determined out of its decision scope, in such a way that they cannot change their size. Their efforts to elevate productivity are driven to minimize the volume of employed resources. Conversely, situations may occur in which

the available resources are out of the range of decision of the production unit. The productivity is then given by the volume of the resulting production. The inclusion in the analysis of a unit with dimensions very different from those found in the rest of the group and which assure it advantages of scale may result in this unit receiving a paradigmatic position that is, in fact, not reasonable to expect to be reached by the other units. Algorithms were developed in [1] to deal with returns to scale. A problem with such algorithms is that any unit with an extreme value becomes necessarily evaluated as fully efficient. The transformation into probabilities of reaching the frontier brings automatically all variables to the same scale.

DEA optimistic foundation of allowing variables weights, the evaluation of each option applying the weights more favorable to it may lead to not taking into account some criteria. In the case of simultaneous evaluation of individual and clustered performances, that will result in the individuals with performance above the average being evaluated by their individual performances while those with performances below the average are evaluated by the aggregate attributes. In the DEA framework an exit to avoid that would be constraining the weights on each criterion to stay below that given to the same criteria when applied to the clusters. In the probabilistic approach a more precise treatment to this problem may be given by taking into account the correlation between the criteria.

Another serious criticism to DEA is driven to the lack of statistical evaluations. Random errors may distort the measures of inputs and outputs. These errors may come from imprecise measurement tools or from conceptual distances between the true inputs and outputs and the variables effectively measured to represent them. The distortions that random disturbances may cause are not squealed by any external sign. The excellence frontiers generated by performances reflecting the effect of large measurements errors cannot be distinguished from those generated by observations accurately measured.

Alternatives have been developed to deal with this difficulty which depend on being able to parametrically model the frontier ([7]) or to statistically model the efficiencies vector ([14]). The probabilistic approach takes into account the random components from the beginning. The errors in the initial measurements are modeled with mild assumptions that will affect in a balanced way the computations of the distances of the different evaluated units to the frontier.

By measuring the distance to the frontier according to each input or output in terms of probability of reaching the frontier, the probabilistic composition preserves DEA advantages of relating the efficiency to observed frontiers and not being influenced by scales of measurement. And by taking into account all variables and all compared options in the evaluation of each option, the probabilistic composition mitigates the influence of extreme observed values. While the frontier of excellence tends to be formed by rare performances, the comparison with a large set of observations with more frequent values makes the evaluation process more resistant to random errors.

3 Modeling Cooperative Attributes

Evaluation systems based on the comparison of individual performances may fail to attend the main objective, of enhancing global improvement, by fostering competitive practices where cooperation would be a more important asset. On the other end, leaving unrecognized individual efforts and evaluating only on the basis of large groups achievements may leave out of the performance evaluation important drives for improvement.

For instance, stimulating the productivity in scientific research by offering grants only to researchers presenting, comparatively, the best results on a list of indicators, simultaneously, stimulates two kinds of attitudes that will harm the development of productive research activity. The first is the detachment of the objectives of the individuals from the objectives of their institutions, which should be the real core of the most important research projects. The second is developing an opposition of each researcher to the success of the pairs which compete for the grants reserved for a same research field.

The evaluation system, even when designed to assign resources to individuals, must take into account variables measuring environmental variables that affect collectively groups of individuals or are affected by the joint action of such groups. By not taking into account social features affecting the individual performances evaluated, the evaluation will be unfair not only to the groups as a whole but even to the individuals compared.

By not taking into account the environmental conditions affecting the activities in the community where they are located, the evaluator that claims to be judging individual productivity may be only measuring individual results attributable to the context where the work is done and not to personal contributions. Sometimes the absolute results are obtained without any productivity of the individual in efficiently exploring resources made available by other sources on which distribution neither the evaluator nor the evaluated person have any interference.

The probabilistic composition provides forms to join, in the same evaluation system, individual and group indicators in such a way that the evaluation of each individual be affected by the group performance but the contributions of the individuals have a significant impact on their particular evaluation.

This system puts together variables measuring individual attributes with variables measuring the same attributes in aggregate units of evaluation. For this reason, positive correlation between the stochastic components of these variables will probably be present.

3.1 Dependence Assumptions

The probability of joint occurrence of two or more events is always smaller or equal to the probability of occurrence of each of them taken isolated. From this follows that the minimum of the isolated probabilities of occurrence is an upper bound for the probability of joint occurrence. On the other hand, for

nonnegatively correlated events, as we may expect to be the events of being the best according to criteria applied to individuals and to clusters to which the same individuals belong, a lower bound to the joint probability is given by that probability computed under the hypothesis of null correlation, i. e., by the product of the isolated probabilities. Thus we have an upper and a lower bound for the probability of joint maximization of preference.

It is interesting to notice that determining the probabilities by the minimum corresponds to the composition of fuzzy logic ([\[17\]](#)) by the necessity and possibility concepts. That means that the fuzzy composition approach corresponds to an extreme of the correlation between indicators of occurrence, the other extreme corresponding to the assumption of independence between the criteria.

The computation of the joint probability by the minimum results also in a ranking procedure similar in spirit to the DEA approach. But instead of looking for the most favorable criterion, the criterion chosen by the maximum correlation assumption is that less favorable to the object of evaluation.

The global ranks derived by independence and by the minimum constitute information that may be used complementarily. Moreover, correlation structures in an intermediary position between these two may also be explored.

Independence between criteria applied to individuals isolated may be assumed and, after computing the joint probability of preference according to these criteria, the criteria related to collective evaluations may enter successively in the computation. The small number of correlations needed in this second stage may be estimated. For instance, a subjective contribution of experts may be employed only to rank the criteria. Based on this ranking, only successive correlations may be estimated.

The key feature of the probabilistic approach to deal with clustering options is the careful handling of the correlation between criteria applied to clusters of options and criteria applied to individual options. A first principle in modeling the correlation in this context will be assuming maximal dependence between cluster indicators and the respective individual indicators. Even if not measuring the same feature, cluster evaluations being more affected by environmental stochastic factors, must be more correlated among themselves and with the individual evaluations than these among themselves.

Another important aspect to be considered is the advantage of assuming independence to let numerical differences possibly registered being fully taken into account. If the importance of the correlations cannot be accessed, this feature would make prevail the decisions derived from the independence assumption.

3.2 Malmquist Indices

Another aspect that may be explored to make the evaluation adequately take into account the context, though centering attention in individual characteristics, is ranking in terms of the evolution of individual positions through time.

The Malmquist productivity index is a measure of relative change through time introduced by [2], following the approach of [8] to measure of evolution of productivity. It sets the evaluation in terms of distance to the best among the observed productivities.

Malmquist indices are quantity indices not depending on revenues or cost shares to aggregate outputs or inputs. The independence on weights or shares is also a characteristic of DEA. Exploring that, [4] has shown that the Malmquist approach of evaluating evolution through time by computing indices to each option relatively to values of the other options fixed on successive time points may be employed in the DEA context. It may be employed by the same way in the probabilistic composition.

To consider the shifts on the frontier, we may use besides the Malmquist indices based on comparison to a fixed sample of the initial or of the final year, a geometric mean of two indices, one relative to the initial frontier and the other relative to the subsequent frontier. The first is calculated dividing by the initial efficiency of the unit under evaluation the efficiency of a hypothetical unit introduced in the initial data set in the place of such unit and with the vector of values observed in it in the subsequent year. In the same way, the second is obtained dividing the efficiency of the unit in the second year by that obtained substituting for their values those of the previous year.

In addition to compare the values associated with the global evolution measures, we may also separate in the index the effect of changes affecting all evaluated options from those proper to each individual evaluated. As in [4], an index of technical change affecting the individual may be obtained dividing the geometric mean Malmquist index by the ratio between the instantaneous evaluations of the individual in the two years.

4 An Example of Large Size

In this section a model for combining in an evaluation system individual and cluster criteria is developed. Three criteria are employed to evaluate clusters formed according to two different classification rules.

The example is built on data of clients of a retail sales chain. data on a total of 5025 clients effectively negotiating with the chain were examined. The objective is to enable the firm to provide customized treatment to different classes of costumers. The same framework can however be employed to model evaluation in many other contexts.

The first aggregate variable, C_1 , is given by a classification on five a priori levels, each with the same number of costumers, determined from the observed value of the transactions of the client with the network. If the system is already active, this first variable may be the classification provided by the model in the last application of the system. If the objective is to reduce inequality, the preference in terms of this variable may be stated in an inverse order.

This volume criterion is complemented by a classification in terms of diversity, C_2 . This second kind of classification is formed, in the present context of network costumers, by ranking the clients according to the number of different sectors of activity of the chain that had transactions with the costumer during the costumer in the last year. In a context of productivity evaluation this second variable may be thought as representing areas of actuation. Those areas where smaller values for the individual indicators are expected might receive higher preference values.

The third variable, C_3 , is designed to determine an intermediary level of aggregation. The clusters for this variable are formed by the intersection of the clusters determined by the two preceding variables. The costumers are now ranked inside the clusters determined by the second variable according to their value in the first, or, equivalently, in the reverse order.

The individual evaluation variables are derived from the Recency, Frequency and Monetary value (RFM) approach to access importance of costumers to the firm ([6]). The first, C_4 , is a recency variable, classifies costumers in decreasing order according to the number of days from the date of the last transaction of the year to the end of the year. The probabilistic transformation is to the probability of minimizing such number. The second, C_5 , is a frequency variable given by the number of visits to the chain during the whole year. And the third, C_6 , is a diversity variable given by the number of distinct products bought by the client during the year. The frontiers of excellence for these two last variables are those of high values.

Independence between the measurements representing the individually accessed variables is a natural assumption because such measurements involve observing behavior at distinct circumstances. Since, in the present case, the individual variables are not directly aggregated into any of the aggregate variables, it is also conceivable that stochastic independence may hold between the two kinds of variables.

Even between the variables measured in an aggregate level, it is disputable if classifications in terms of volume and classifications in terms of diversity may show dependence. It may well be assumed that errors in the measurement of C_1 and C_2 are independent.

Table 1 shows, for two successive years, the correlations between the initial classification in five strata employed in the first criterion and a final classification in five strata of equal size derived from the final ranking resulting from four hypotheses on dependence suggested by the reasoning above developed. The hypotheses confronted are (e , in the labels denoting composition by the minimum of the probabilities, and $*$ denoting composition by the product):

- Dependence between blocks and between aggregate classifications ($D_{AB} = C_1eC_2eC_3eC_4*C_5*C_6$)
- Dependence only between the two blocks ($D_B = C_1*C_2*C_3eC_4*C_5*C_6$)

Table 1. Correlations Between Final and Initial Classifications

Year	D _{AB}	D _B	D _A	I
1	0.66	0.89	0.94	0.91
2	0.81	0.89	0.86	0.90

- Dependence only between aggregate views ($D_A = C_1 e C_2 e C_3 * C_4 * C_5 * C_6$)
- Independence between all the classifications ($I = C_1 * C_2 * C_3 * C_4 * C_5 * C_6$)

Table 1 reveals that the difference between the results derived from distinct assumptions is small. But it is clear that the hypothesis of independence only between the individually evaluated criteria, that means, dependence between the criteria evaluating clusters and dependence between the evaluation according to these criteria and that according to the criteria evaluating the individuals isolated, is the hypothesis that leads to a final classification less correlated to the initial classification. So, to allow for the maximal refreshment of positions, this is the hypothesis to be assumed.

5 An example with Averaged Attributes

Brazil has an influential system of evaluation of graduate courses. The system is managed by CAPES, a public funding agency for higher education that applies the results of the evaluation to base its decisions relative to scholarships and financial support to projects. An institution is allowed to start offering Ph. D. Programs only if its Master Program in the area is graded above good. Courses with low grades are forced into not accepting new students. Other funding organizations also consult CAPES classification. Thus, briefly speaking, this evaluation provides a reference for the whole community of research and higher education in Brazil.

CAPES evaluation system is based on data annually provided by the institutions, automatically summarized in a large set of numerical indicators. The final evaluations are presented in a scale from 1 to 5 for the programs offering only M. Sc. Courses and from 1 to 7 in the case of Ph. D. courses. Such degrees are valid for three years, but annually the partial indicators and comments on them are officially published.

The system was developed in close connection with the academic community and is strongly influenced by the principle of appraisal by the peers. In such a way that, although the final decisions on resources allocation and the final grades of the courses must be approved by a Scientific Committee, considerable power of judgment is concentrated on small committees representing the researchers in each area of knowledge. These committees assemble, as a rule, twice a year: first to review their criteria and weights and later to examine the information gathered about the courses and issue their evaluations.

The committees are constituted through the appointment by CAPES Board of an area representative for a three years term. The remaining members of the committee are chosen by this area representative and dismissible ad nutum. The practical meaning of this appointment system is that the area representatives are the people who must concur with the management ability, define the area goals and understand how they fit in the government global strategy to improve higher education in the country.

This structure has been strong enough to hold through decades. Occasional criticism from experts in evaluation who would like to see decisions more clearly related to a management philosophy, objective goals and well defined liabilities, and from courses managers who do not see their real concerns and achievements effectively taken into evaluation have been easily overridden by the ability of choosing a team of area representatives who correctly mirror the political power balance between the higher education institutions.

This section applies indices of the kind above discussed to evaluate the programs of one research areas during the period from 2001 to 2006. This area includes the programs in Mechanical Engineering, Production Engineering and a series of other areas with small number of courses.

The new indices proposed are designed to consider two main aspects not taken into account in the CAPES system. The first is the interest in stimulating cooperation between programs of the same area or programs geographically close.

The second is to pay more importance to the relative evolution through time, instead of to the relative instantaneous positions. The idea is to compare the programs, first of all, with themselves, the other programs furnishing only a paradigm with respect to which the evolution of each one would be compared. This will be done by employing Malmquist indices.

Since, in the geometric mean Malmquist index, the global evolution of similar courses in the same direction may reduce the value of the individual evaluation, the presence of evaluations of the clusters in the evaluation of the individuals become important.

Indices with these properties may complement the indices of CAPES. Trying to make fairer the comparison, the evaluation of the programs came to adopt in the last years the principle of standardizing size, to rely on the comparison between courses of the same size. This has induced a downsizing philosophy that has extended the competition between the programs into a competition inside the programs, as the exclusion of people from the program may be a form of raising the value of many indicators that take the number of Faculty members as a denominator.

Thus, besides a comparison between the results of the application of this approach and the official classification by CAPES, an evaluation of the influence of size in each of these evaluations may be relevant.

Only two kinds of products are examined here: publication and formation. These are the kinds of products with heavier weights in the computation of the indices of CAPES too.

These concepts are measured by four variables. Two consider publication: total number of papers published in periodicals classified by CAPES in the highest level, and total number of books, chapters of books and papers in other periodicals considered by CAPES of at least a nationally acceptable level.

Two other variables consider formation: total number of D. Sc. theses approved and total number of M. Sc. dissertations approved.

Each program is also evaluated by the average of each of these variables in the cluster of programs of the same knowledge area and in the cluster of programs of the same geographic area. The knowledge areas considered are of industrial engineering, mechanical engineering and other courses evaluated together with those of these two areas by the same committee of CAPES.

The geographic areas are formed by the states with at least five distinct programs among the 61 programs considered, which are located in the South and Southeast regions of the country: Sao Paulo, Rio de Janeiro, Minas Gerais, Rio Grande do Sul, Parana and Santa Catarina, and a seventh group taking together all the other states of the country.

The data analyzed, for the triennials 2001-2003 and 2004-2006, are available at capes.gov.br.

Table 2 presents the global productivities according to a pessimistic and conservative evaluation for 2004-2006 by taking into account only the individual evaluations, assumed independent, and, alternatively, by taking into account also the aggregate indicators. In this last case, these aggregate indicators are assumed to be independent among themselves but dependent in block to the individual indicators. The indices for 2100-2003 present similar values, with correlation of 0.89, 0.91 and 0.82, respectively.

In the composition of these indices, under the independence hypothesis a geometric mean was substituted for the joint probability. This avoids the effect of the number of factors in reducing the product of the probabilities.

Table 2 presents also, in the last column, the geometric mean Malmquist indices for the hypothesis of correlation between blocks. The Malmquist indices for the other hypothesis follow a similar pattern.

Table 3 presents the correlations of the vectors of evaluations according to each of the probabilistic indices in Table 2 with the official evaluations and with the size of the programs in terms of number of Faculty members. As was to be expected, since the kind of consideration they bring to the analysis is not taken into account in the present CAPES evaluation, the first set of correlations shows a decrease as aggregate indicators are taken into account.

It is interesting that the same happens with the correlations with the size of the program. This shows a desirable effect of the inclusion of the aggregate indicators, of transferring to the evaluation of the small programs the good results of the large programs close to them in terms of area of knowledge or geographically. It should be noticed also that, as the downsizing effect above referred affects the small programs as well as the large ones, employing the

Table 2. Global Indices

Inst.		Area	CAPES	No Cl.	independent blocks	Malmquist
UFMG	I	3	0.88	0.91	0.88	1.04
UNIF	I	3	0.88	0.89	0.88	1.04
UFAM	I	3	0.87	0.88	0.87	1.07
UFPB	I	3	0.89	0.89	0.87	1.07
UFRN	I	3	0.88	0.87	0.86	1.06
UFPE	I	5	0.93	0.90	0.87	1.08
UENF	I	3	0.86	0.91	0.86	1.03
PUCR	I	4	0.91	0.90	0.89	1.05
UFF	I	4	0.91	0.93	0.91	1.03
UFRJ	I	5	0.97	0.94	0.93	1.05
UFSM	I	3	0.90	0.91	0.90	1.04
UFRS	I	5	0.92	0.90	0.89	1.02
UFSC	I	3	0.92	0.92	0.92	1.09
UNIP	I	3	0.88	0.93	0.88	1.03
UFCR	I	4	0.95	0.94	0.93	1.03
UNIM	I	4	0.94	0.93	0.93	1.03
USPC	I	4	0.94	0.93	0.93	1.05
USP	I	5	0.94	0.95	0.94	1.04
PUCM	M	4	0.88	0.90	0.88	1.03
UFMG	M	4	0.95	0.93	0.92	1.05
UNIF	M	4	0.91	0.92	0.90	1.04
UFU	M	5	0.95	0.93	0.90	1.04
UFES	M	3	0.85	0.87	0.85	1.04
UFPA	M	3	0.87	0.88	0.87	1.05
UFPB	M	4	0.92	0.89	0.88	1.07
UFPE	M	4	0.88	0.88	0.88	1.06
UFRN	M	4	0.90	0.89	0.87	1.06
UNB	M	4	0.89	0.88	0.88	1.06
PUCP	M	3	0.89	0.90	0.89	1.05
UFPR	M	4	0.92	0.91	0.90	1.06
IME	M	3	0.85	0.90	0.85	1.04
PUCR	M	3	0.94	0.93	0.93	1.03
UFF	M	4	0.88	0.91	0.88	1.03
UFRJ	M	6	0.96	0.94	0.92	1.02
UFRS	M	5	0.95	0.93	0.90	1.02
UFSC	M	6	0.97	0.94	0.91	1.03
UNPB	M	3	0.88	0.93	0.88	1.02
UNPI	M	3	0.89	0.93	0.89	1.06
UNIT	M	3	0.89	0.91	0.89	1.08
UNIV	M	3	0.86	0.92	0.86	1.05
UNPG	M	4	0.95	0.95	0.95	1.03
USP	M	5	0.97	0.94	0.93	1.02
USPC	M	5	0.96	0.96	0.94	1.02
UNIC	M	6	0.98	0.94	0.93	1.02
UNIF	O	3	0.88	0.89	0.88	1.06

Table 2. (*continued*)

Inst.	Area	CAPES	No Cl.	independent blocks	Malmquist
UFBA	O	3	0.87	0.85	0.84
UNB	O	3	0.85	0.86	0.85
UTFP	O	4	0.90	0.89	0.87
PUCP	O	6	0.89	0.87	0.86
CEFR	O	3	0.88	0.89	0.88
UENF	O	3	0.90	0.89	0.89
UFF	O	3	0.91	0.90	0.89
PUCR	O	4	0.87	0.88	0.87
UFRJ	O	4	0.93	0.91	0.91
FURG	O	3	0.88	0.87	0.87
UFSC	O	3	0.86	0.89	0.86
ITA	O	6	0.97	0.93	0.91
USP	O	3	0.90	0.93	0.90
INPE	O	4	0.95	0.92	0.91
UNIC	O	4	0.91	0.92	0.91
USP	O	4	0.89	0.91	0.89

Table 3. Correlations Between Final and Initial Classifications

	nocluster	independent blocks	Malmquist
CAPES	0.73	0.46	0.46
SIZE	0.75	0.52	0.57

number of Faculty members as denominator has not substantially reduced the positive correlation of CAPES degrees with size, still around 0.5.

Malmquist indices show no correlation with CAPES vector of evaluations. It shows also no correlation with size, presenting, in fact, a negative correlation with both. This shows that evaluating in terms of evolution results in an entirely new ranking.

On the other side, the Malmquist indices are above 1 for every program, if the aggregate indicators are taken into account. For the case of employing only the individual measurements, they are in general, smaller, but rarely below 1. This reflects the global increase in production. Comparing in terms of the magnitude of this increase seems to be a much sounder approach to programs evaluation.

A final comment must be made on the approach taken to derive the joint probabilities. The choice of the pessimistic and conservative point of view avoids the risk of having the evaluation distorted by exceptionally large values in isolated variables. By this way, this approach will discourage, for instance, competition for exceptionally large amounts of published articles with disregard to formation or vice-versa.

Another comment should be made on the dependence instances taken in the construction of the indices. The choice of the dependence structure is based on the results of the analysis of the example in the previous section. Besides, the assumption of independence was preferred due to its ability to take into account more precisely the probabilities of being preferred according to each criterion.

Taking into account the correlation between blocks of individual and aggregate evaluations pays due attention to the theoretical dependence between indicators based on the same data and is expected to increase the influence of the aggregate indicators, generating a new classification. This last effect was not found as the different forms of computation of the indices that consider the aggregate data led to similar vectors of evaluations.

6 Conclusion

Along this chapter, a new set of forms of combining evaluations by means of the determination of probabilistic preferences was developed. The probabilistic composition allows to combine evaluations with different levels of aggregation with results easy to interpret as the examples presented demonstrated. It allowed for comparing costumers as well as academic performances under different points of view. It was shown how simple model assumptions allowed for overcoming the absence of previous information about the probability distribution of the stochastic disturbances.

The examples showed also how the probabilistic approach is able to make clear the higher influence of different separate criteria in the final evaluation. Combining different criteria led to different ranking, while the form of combination had a smaller effect.

The applications made involved sets of options of a size extremely larger than usual. The size of 5025 costumers of the first data set is rarely dealt with by multicriteria composition methods. The number of criteria, already larger than usual in the second example, can also be extended without any conceptual change.

The examples studied bring basics frameworks of dependence relations between criteria that can be explored in other applications. Only extreme dependence relations were assumed. Efforts should be taken to obtain a quantitative basis of information on possible intermediary correlation structures.

The application to other instances of the same problem should bring also other opportunities of development, by raising other clustering structures. Important areas of possible application are in the public sector, where evaluation must frequently face the need to take into account criteria unrelated to quantitative attributes. An important feature of the evaluation system here developed is its full independence of the availability of numerical measurements to start with.

References

1. Banker, R.D., Charnes, A.H., Cooper, W.W.: Some Models for Estimating Technical and Scaling Inefficiencies in DEA. *Management Science* 30, 1078–1092 (1984)
2. Caves, D.H., Christensen, L.R., Diewert, W.E.: The Economic Theory of Index Numbers and the Measurement of Input, Output and Productivity. *Econometrica* 50, 1393–1414 (1982)
3. Charnes, A.H., Cooper, W.W., Rhodes, E.: Measuring the Efficiency of Decision Making Units. *European Journal of Operations Research* 2, 429–444 (1978)
4. FäRe, R., Griffel-Tatje, E., Grosskopf, S., Lovell, C.A.K.: Biased Technical Change and the Malmquist Productivity Index. *Scandinavian Journal of Economics* 99, 119–127 (1997)
5. Farrell, M.J.: The measurement of productive efficiency. *Journal of the Royal Statistical Society, A* 120, 449–460 (1957)
6. Hughes, A.: Strategic Database Marketing. McGraw-Hill, New York (2005)
7. Kumbhakar, S.C., Lovell, C.A.K.: Stochastic Frontier Analysis. Cambridge University Press, Cambridge (2000)
8. Malmquist, S.: Malmquist, Index Numbers and Indifference Surfaces. *Trabajos de Estadística* 4, 209–242 (1953)
9. Montgomery, D.C.: Introduction to Statistical Quality Control, 5th edn. J. Wiley, New York (2005)
10. Pedrycz, W.: Why triangular membership functions? *Fuzzy Sets and Systems* 64, 21–30 (1994)
11. Santáanna, A.P., Santáanna, L.A.F.: Randomization as a Stage in Criteria Combining. In: Proceedings of the VII ICIEOM, pp. 248–256 (2001)
12. Santáanna, A.P.: Data Envelopment Analysis of Randomized Ranks. *Pesquisa Operacional* 22, 203–215 (2002)
13. Santáanna, A.P.: Evaluation of Fuzzy Productivity of Graduate Courses. In: Nedjah, N., Mourelle, L.M., Borges, M.N., Almeida, N. (eds.) Intelligent Educational Machines. Springer, Heidelberg (2005)
14. Simar, L., Wilson, P.W.: Sensitivity analysis of efficiency scores: How to bootstrap in nonparametric frontier models. *Management Science* 44, 49–61 (1998)
15. Wichura, M.J.: The Coordinate-Free Approach to Linear Models. Cambridge University Press, New York (2006)
16. Zadeh, L.A.: Fuzzy sets. *Information and Control* 8, 338–353 (1965)
17. Zadeh, L.A.: Fuzzy Sets as the Basis for a Theory of Possibility. *Fuzzy Sets and Systems* 1, 3–28 (1978)

Use of the q -Gaussian Function in Radial Basis Function Networks

Renato Tinós and Luiz Otávio Murta Júnior

Departamento de Física e Matemática
Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto (FFCLRP)
Universidade de São Paulo (USP)
Av. Bandeirantes, 3900
14040-901, Ribeirão Preto, SP, Brazil
rtinos@ffclrp.usp.br, murta@ffclrp.usp.br

Summary. Radial Basis Function Networks (RBFNs) have been successfully employed in several function approximation and pattern recognition problems. In RBFNs, radial basis functions are used to compute the activation of artificial neurons. The use of different radial basis functions in RBFN has been reported in the literature. Here, the use of the q -Gaussian function as a radial basis function in RBFNs is investigated. An interesting property of the q -Gaussian function is that it can continuously and smoothly reproduce different radial basis functions, like the Gaussian, the Inverse Multiquadratic, and the Cauchy functions, by changing a real parameter q . In addition, the mixed use of different shapes of radial basis functions in only one RBFN is allowed. For this purpose, a Genetic Algorithm is employed to select the number of hidden neurons, and center, width and q parameter of the q -Gaussian radial basis function associated with each radial unit. The RBF Network with the q -Gaussian RBF is compared to RBF Networks with Gaussian, Cauchy, and Inverse Multiquadratic RBFs in problems in the Medical Informatics domain.

1 Introduction

Radial Basis Function (RBF) Networks are a class of Artificial Neural Networks where RBFs are used to compute the activation of artificial neurons. RBF Networks have been successfully employed in real function approximation and pattern recognition problems. In general, RBF Networks are associated with architectures with two layers, where the hidden layer employs RBFs to compute the activation of the neurons. Different RBFs have been used, like the Gaussian, the Inverse Multiquadratic, and the Cauchy functions [22]. In the output layer, the activations of the hidden units are combined in order to produce the outputs of the network. While there are weights in the output layer, they are not present in the hidden layer.

When only one hidden layer and output neurons with linear activation function are employed, RBF Networks can be trained in two steps. First, the

parameters of the radial basis units are determined. Then, as the outputs of the hidden units and the desired outputs for each pattern are known, the weights are computed by solving a set of linear equations via least squares or singular value decomposition methods [4]. Thus, gradient descendent techniques are avoided and determining the radial basis units' parameters becomes the main problem during the training phase.

The selection of the parameters of the radial basis units means to determine the number of hidden neurons, the type, widths, and centers of the RBFs. In several cases, the first three parameters are previously defined, and only the radial basis centers are optimized [20]. Besides the centers, the number of hidden neurons [6], [25] and the widths [5] can be still optimized. In general, all the radial units have the same type of RBF, e.g., the Gaussian function, which is chosen before the training.

In [13], the effect of the choice of radial basis functions on RBF Networks was analyzed in three time series prediction problems. Using the k -means algorithm to determine the radial basis centers, the authors compared the results produced by RBF Network with different types of RBFs for different numbers of hidden neurons and widths of the RBFs. In the experiments conducted in [13], all radial units have the same fixed RBF type and width for each RBF Network. The authors concluded that the choice of the RBF type is problem dependent, e.g., while the RBF Network with hidden neurons with Gaussian transfer function presented best performance in one problem, the choice of the Inverse Multiquadratic function was beneficial for another problem.

Here, the mixed use of different shapes of radial basis functions in RBF Networks, i.e., the hidden neurons can have different radial basis functions in a same RBF Network, is investigated. For this purpose, the q -Gaussian function, which reproduces different RBFs by changing a real parameter q , is used. Thus, the choice of the number of hidden neurons, center, type (parameter q), and width of each RBF can be viewed as a search problem.

In this work, a Genetic Algorithm (GA) is employed to select the number of hidden neurons, center, type, and width of each RBF associated with each hidden unit. The methodology used here is presented in Section 5. Before, Section 2 presents the RBF Network model, Section 3 introduces the q -Gaussian function, and Section 4 discusses the use of the q -Gaussian RBF. Section 6 presents and discusses an experimental study with two pattern recognition problems of the Medical Informatics domain, in order to test the performance of the investigated methodology. In the experiments, the RBF Network with the q -Gaussian RBF is compared to RBF Networks with Gaussian, Cauchy, and Inverse Multiquadratic RBFs. Finally, Section 7 presents the final conclusions.

2 Radial Basis Function Networks

RBF are a class of real-valued functions where its output depends on the distance between the input pattern and a point \mathbf{c} , defined as the center of

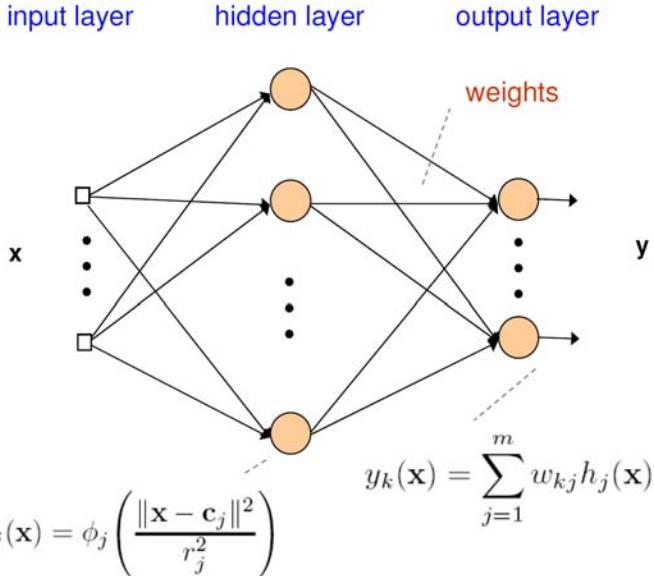


Fig. 1. RBF Network with one hidden layer

the RBF. Moody and Darken proposed the use of RBFs in Artificial Neural Networks (ANNs) inspired by the selective response of some neurons [20]. ANNs where RBFs are used as activation functions are named RBF Networks. The architecture and the learning of RBF Networks are described in the next sections.

2.1 Architecture

RBF Networks can have any number of hidden layers and outputs with linear or nonlinear activation. However, RBF Networks are generally associated with architectures with only one hidden layer without weights and with an output layer with linear activation (Figure 1). Such architecture is employed because it allows the separation of the training in two phases: when the radial units' parameters are determined, the weights of the output layer generally can be easily computed.

The output of the k -th neuron in the output layer of an RBF Network, where $k = 1, \dots, q$, is given by

$$y_k(\mathbf{x}) = \sum_{j=1}^m w_{kj} h_j(\mathbf{x}) \quad (1)$$

where $h_j(\mathbf{x})$ is the activation of the radial unit $j = 1, \dots, m$ for the input pattern \mathbf{x} and w_{kj} is the synaptic weight between the radial unit j and the

output neuron k . The activation of the j -th radial unit is dependent on the distance between the input pattern and the hidden unit center \mathbf{c}_j . Using an Euclidean metric [22], the activation of the j -th radial unit can be defined as

$$h_j(\mathbf{x}) = \phi_j(d_j(\mathbf{x})) \quad (2)$$

where

$$d_j(\mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{c}_j\|^2}{r_j^2}, \quad (3)$$

$\phi_j(\cdot)$ and r_j are, respectively, the RBF and the scalar parameter that defines its width for the j -th radial unit, and $\|\cdot\|$ is the Euclidean norm.

The most common RBF is the Gaussian function, which is given by

$$\phi_j(d_j(\mathbf{x})) = e^{-d_j(\mathbf{x})} \quad (4)$$

Neurons with Gaussian RBF present a very selective response, with high activation for patterns close to the radial unit center and very small activation for distant patterns.

In this way, other RBFs with longer tails are often employed. Two examples of such RBFs are the Cauchy function, given by

$$\phi_j(d_j(\mathbf{x})) = \frac{1}{1 + d_j(\mathbf{x})} \quad (5)$$

and the Inverse Multiquadratic, defined by

$$\phi_j(d_j(\mathbf{x})) = \frac{1}{(1 + d_j(\mathbf{x}))^{1/2}} \quad (6)$$

2.2 Learning

The first step in the training of the RBF Network presented in Figure 11 is the choice of the number of radial units and the parameters of each one of them. The original method proposed in [20] employs the k -means algorithm to determine the RBF center locations. In this case, the number of radial units is equal to k and is determined before the training. The width and the type of the RBFs are fixed too.

Instead of using the k -means to determine the RBF centers, the input patterns of the training set can be used as center locations. The simplest method is to select all the input patterns of the training set as centers of the radial units. However, this method is not generally used because of the large number of radial units employed and the occurrence of overfitting. Subset selection can be an interesting approach to avoid such problems. Thus, besides the center locations, the number of hidden neurons can still be optimized. The problem of finding the best subset of input patterns to be used as radial units' centers is generally intractable when the training set is large. In this way, heuristics can be used to find a good subset.

In [8], Forward Selection and Orthogonal Least Squares were employed to select the centers of the radial units based on the instances of the training set. Besides the centers, the widths of the RBFs can be optimized, like in the Generalized Multiscale RBF Network [5].

In recent years, there is a growing interest in optimizing the radial units' parameters of RBF Networks using Evolutionary Algorithms [14], which are a class of meta-heuristic algorithms successfully employed in several similar search problems.

In [6], a GA is used to select a subset of radial units centers based on the instances of the training set. The chromosome of each individual i of a population is composed of a subset of indexes of the input patterns of the training set. For each index in the chromosome i , a radial unit with center located at the respective input pattern is added to the RBF Network i . For example, if the chromosome of the individual i is $\mathbf{z}_i^T = [2 \ 98 \ 185]$, then a RBF Network is created with three radial units with centers located at the input patterns \mathbf{x}_2 , \mathbf{x}_{98} , and \mathbf{x}_{185} . For the fitness evaluation of the individual i , the respective RBF Network is created and the Akaike's Information Criterion (AIC) is computed. Then, the computed AIC, which has a term that evaluates the RBF Network performance and a term that evaluates the RBF Network complexity, is used as the fitness of the individual i . All radial units have the same type of RBF, which is defined *a priori*. In [16], the radial unit widths are incorporated in the chromosomes too.

The next step is to compute the weights of the RBF Network. When the number of hidden units and the parameters of the RBFs are fixed, then the radial unit activation can be determined for each instance of the training set and the system defined by Eq. 1 can be viewed as a linear model. Minimizing the sum of squared errors, the optimal weight vector of the k -th output neuron [22] can be computed by

$$\hat{\mathbf{w}}_k = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \hat{\mathbf{y}}_k \quad (7)$$

where $\hat{\mathbf{y}}_k$ is the vector of the desired outputs of the k -th neuron for the instances of the training set, \mathbf{H} is the design matrix formed by the activations of the radial units $h_j(\cdot)$ for each instance of the training set, and \mathbf{H}^T denotes the transpose of the matrix \mathbf{H} .

3 The q -Gaussian Function

In this section we describe some theoretical and historical aspects concerning the origins of the q -Gaussian function. It is important to observe that the q -Gaussian is not an alternative to the classic Gaussian function but a parametric generalization of Gaussian function. The main use of the q -Gaussian function is as the probability distribution function that arises naturally when we consider central limit theorem from sum of random variables with global

correlations [28]. In order to understand the origins of the q -Gaussian distribution, it is necessary to understand the non-extensive q -statistics and the generalized q -entropy.

Originally, information entropy was conceived by Shannon intuitively rather than from formal basic principles. The mathematical function that reproduces the adequate behavior of classic information entropy is the logarithmic function. The choice of a logarithmic base in the entropy formula determines the unit for information entropy used. The most common unit of information is the bit, based on the binary logarithm. An interesting and useful property of entropy is the fact that, for a closed dynamic system, the entropy always grows to a maximum value.

The classic formalism has been shown to be restricted to the domain of validity of the Boltzmann-Gibbs-Shannon (BGS) statistics, and it seems to describe nature when the effective microscopic interactions and the microscopic memory are short ranged. Generally, systems that obey BGS statistics are called extensive systems. If we consider that a physical system can be decomposed into two statistical independent subsystems A and B , the probability of the composite system is $p^{A+B} = p^A p^B$. It can be verified that the Shannon entropy has the additivity property

$$S(A + B) = S(A) + S(B) \quad (8)$$

However, for a certain class of physical systems, presumably those with long-range interactions, long time memory and multifractal like macrostates, an extension of this principle can be interesting. Inspired by multifractals concepts, Tsallis has proposed a generalization of the BGS statistics [26], which is based on a generalized entropic form

$$S_q = \frac{1 - \sum_{i=1}^k p_i^q}{1 - q}, \quad (9)$$

where k is the total number of possibilities of the system and the real number q is an entropic index that characterizes the degree of nonadditivity. This expression meets the BGS entropy in the limit $q \rightarrow 1$.

The Tsallis entropy is nonadditive in such a way that for a statistical independent system, the entropy of the system is defined by the following nonadditivity entropic rule

$$S_q(A + B) = S_q(A) + S_q(B) + (1 - q)S_q(A)S_q(B) \quad (10)$$

From this paradigm, a kind of q -mathematics [27], [12], [21], [7] that is appropriate to q -statistics has emerged. By definition, the q -sum of two numbers is defined as

$$x +_q y = x + y + (1 - q)xy \quad (11)$$

The q -sum is commutative, associative, recovers the conventional summing operation if $q = 1$ (i.e. $x +_1 y = x + y$), and preserves 0 as the neutral element (i.e. $x +_q 0 = x$). By inversion, one can define the q -subtraction as

$$x -_q y = \frac{x - y}{1 + (1 - q)y} \quad (12)$$

The q -product for x, y is defined by the binary relation

$$x \cdot_q y = [x^{1-q} + y^{1-q} - 1]^{1/(1-q)} \quad (13)$$

This operation, also commutative and associative, recovers the usual product when $q = 1$, and preserves 1 as the unity. It is defined only when $x^{1-q} + y^{1-q} \geq 1$. Also by inversion, it can be defined the q -division

$$x /_q y = (x^{1-q} - y^{1-q} + 1)^{1/(1-q)} \quad (14)$$

As well known in classical statistical mechanics, the Gaussian maximizes, under appropriate constraints, the classic entropy. The q -generalization of the classic entropy introduced in [26] as the basis for generalizing the classic theory reaches its maximum at the distributions usually referred to as q -Gaussian. This fact, and a number of conjectures [9] and numerical indications [10], suggest that there should be a q -analog of the central limit theorem (CLT) as well. Limit theorems, in particular, the CLTs, surely are among the most important theorems in probability theory and statistics. They play an essential role in various applied sciences as well. Various aspects of this theorem and its links to statistical mechanics and diffusion have been discussed during recent decades as well.

The q -analysis began at the end of the 19th century, as stated by McAnally [18], recalling the work of Rogers [23] on the expansion of infinite products. Recently, however, its development brought together the need for the generalization of special functions to handle nonlinear phenomena [10]. The problem of the q -oscillator algebra [4], for example, has led to q -analogues of many special functions, in particular the q -exponential and the q -gamma functions [18], [1], the q -trigonometric functions [2], q -Hermite and q -Laguerre polynomials [9], [3], which are particular cases of q -hypergeometric series.

4 The q -Gaussian Radial Basis Function

The use of the q -Gaussian function as a radial basis function in RBF Networks is interesting because it allows changing the shape of the RBF according to the real parameter q [29]. The q -Gaussian RBF for the radial unit j can be defined as

$$\phi_j(d_j(\mathbf{x})) = e_{q_j}^{-d_j(\mathbf{x})} \quad (15)$$

where q_j is a real valued parameter and the q -exponential function of $-d_j(\mathbf{x})$ [27] is given by

$$e_{q_j}^{-d_j(\mathbf{x})} \equiv \begin{cases} \frac{1}{(1+(q_j-1)d_j(\mathbf{x}))^{\frac{1}{q_j-1}}} & \text{if } (1+(q_j-1)d_j(\mathbf{x})) \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

An interesting property of the q -Gaussian function is that it can reproduce different RBFs for different values of the real parameter q . For large negative numbers, the function is concentrated around the center of the RBF. When the value of q increases, the tail of the function becomes larger.

In the next, Eq. 16 will be analyzed for $q \rightarrow 1$, $q = 2$, and $q = 3$. For simplicity, the index j and the dependence on \mathbf{x} will be omitted in the following equations. For $q \rightarrow 1$, the limit of the q -Gaussian RBF can be computed

$$\lim_{q \rightarrow 1} e_q^{-d} = \lim_{q \rightarrow 1} \frac{1}{(1 + (q - 1)d)^{\frac{1}{q-1}}} \quad (17)$$

If we write $z = (q - 1)d$, then

$$\begin{aligned} \lim_{q \rightarrow 1} e_q^{-d} &= \lim_{z \rightarrow 0} (1 + z)^{-\frac{d}{z}} \\ \lim_{q \rightarrow 1} e_q^{-d} &= \lim_{z \rightarrow 0} \left((1 + z)^{\frac{1}{z}} \right)^{-d} \end{aligned} \quad (18)$$

The limit of the function $(1 + z)^{\frac{1}{z}}$ is well known and converges to e when $z \rightarrow 0$. Thus,

$$\lim_{q \rightarrow 1} e_q^{-d} = e^{-d} \quad (19)$$

In this way, we can observe that the q -Gaussian RBF (Eq. 16) reduces to the standard Gaussian RBF (Eq. 4) when $q \rightarrow 1$.

Replacing $q = 2$ in Eq. 16, we have

$$e_q^{-d} = \frac{1}{1 + d} \quad (20)$$

i.e., the q -Gaussian RBF (Eq. 16) is equal to the Cauchy RBF (Eq. 5) for $q = 2$.

When $q = 3$, we have

$$e_q^{-d} = \frac{1}{(1 + 2d)^{1/2}} \quad (21)$$

i.e., the activation of a radial unit with an Inverse Multiquadratic RBF (Eq. 6) for d is equal to the activation of a radial unit with a q -Gaussian RBF (Eq. 16) for $d/2$.

Figure 2 presents the radial unit activation for the Gaussian, Cauchy, and Inverse Multiquadratic RBFs. The activation for the q -Gaussian RBF for different values of q is still presented. One can observe that the q -Gaussian reproduces the Gaussian, Cauchy, and Inverse Multiquadratic RBFs for $q \rightarrow 1$, $q = 2$, and $q = 3$. Another interesting property of the q -Gaussian RBF is still presented in Figure 2: a small change in the value of q represents a smooth modification on the shape of the RBF.

In the next section, a methodology to optimize the RBF parameters of the hidden units in RBF Networks via Genetic Algorithms is presented.

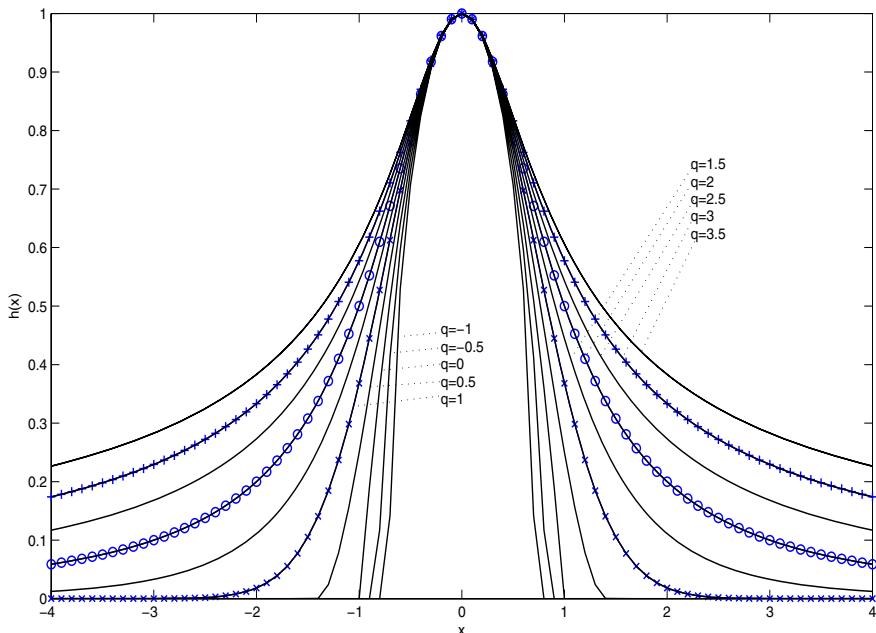


Fig. 2. Radial unit activation in an one-dimensional space with $c = 0$ and $r = 1$ for different RBFs: Gaussian ('x'), Cauchy ('o'), Inverse Multiquadratic for $\sqrt{2}x$ ('+'), and q -Gaussian RBF with different values of q (solid lines)

5 Selection of Parameters of the q -Gaussian RBFs via Genetic Algorithms

In the investigated methodology, a Genetic Algorithm (GA) is used to define the number of radial units m , and the parameters of each RBF related to each hidden unit $j = 1, \dots, m$, i.e., the center, width, and parameter q for each radial unit with q -Gaussian RBF. Algorithm 1 describes the procedure to select the parameters of the radial units. The GA is described in the next.

5.1 Codification

A hybrid codification (binary and real) is employed in the GA used in this work. Each individual i ($i = 1, \dots, \mu$) is described by a vector (chromosome) with $3N$ elements, where N is the size of the training set. The individual i is defined by the vector

$$\mathbf{z}_i^T = [b_1 \ r_1 \ q_1 \ b_2 \ r_2 \ q_2 \ \dots \ b_N \ r_N \ q_N] \quad (22)$$

where b_j is a bit that defines the use of the j -th training pattern as a center of a radial unit. If $b_j = 1$, a radial unit is created with center equal to the training pattern \mathbf{x}_j , and with width and q -parameter respectively given by

Algorithm 1

-
- 1: Initialize the population composed of random individuals $\mathbf{z}_i^T = [b_1 \ r_1 \ q_1 \ b_2 \ r_2 \ q_2 \dots b_N \ r_N \ q_N]$, where N is the size of the training set, the bit b_j ($j = 1, \dots, N$) defines if the j -th training pattern is used as the center of a radial unit with width r_j and q -parameter q_j .
 - 2: **while** (stop criteria are not satisfied) **do**
 - 3: Apply elitism and tournament selection to generate the new individuals
 - 4: Apply crossover and mutation
 - 5: Compute the fitness (Eq. 25) of each individual i in the new population by evaluating the RBF Network defined by the individual i . The number of radial units m_i of the RBF Network related to individual i is equal to the number of ones in the elements b_j ($j = 1, \dots, N$) of the vector \mathbf{z}_i . When $b_j = 1$, a new radial unit is added with center defined by the j -th training pattern, and width and q -parameter given by r_j and q_j . The outputs of the RBF Network are computed using eqs. 11, 2, and 7
 - 6: **end while**
-

the real numbers r_j and q_j . The number of radial units m_i for each individual i is equal to the number of ones in the bits b_j of the vector \mathbf{z}_i . In the first generation, the number of elements b_j equal to 1, i.e. the number of radial units m_i , is generated with mean n_b . Then, the number of radial units m_i is allowed to change by crossover and when a bit b_j is mutated.

For example, if the chromosome of the individual i is given by

$$\mathbf{z}_i^T = [0 \ 0.11 \ 1.26 \ 1 \ 0.21 \ 0.82 \ 1 \ 0.15 \ 1.67 \ 0 \ 0.32 \ 0.96 \ 1 \ 0.02 \ 2.3] \quad (23)$$

where, for simplicity, $N = 5$, then the RBF i is composed of three radial units with: centers located in the input patterns \mathbf{x}_2 , \mathbf{x}_3 , and \mathbf{x}_5 ; widths 0.21, 0.15, and 0.02; q -parameters 0.82, 1.67, and 2.3.

5.2 Operators

Here, the GA uses three standard operators:

Selection

Tournament selection and elitism are employed here. Elitism is employed in order to preserve the best individuals of the population. Tournament selection is an interesting alternative to the use of fitness-proportionate selection mainly to reduce the problem of premature convergence and the computational cost [19]. In tournament selection, two (or more) individuals from the population are randomly chosen with uniform distribution in order to form a tournament pool. When the tournament size is equal to two, a random number with uniform distribution in the range $[0,1]$ is generated and, if this number is smaller than the parameter p_s , the individual with the best fitness from the tournament pool is selected. Otherwise, the remaining individual is selected. This procedure is repeated until the new population is complete.

Crossover

When the standard crossover is applied, parts of the chromosomes of two individuals are exchanged. Two point crossover is applied in each pair of new individuals with crossover rate p_c . Here, the crossover points are allowed only immediately before the elements b_j (see Eq. 22). In this way, individuals exchange all the parameters of a radial unit each time.

Mutation

Two types of mutation are employed. The standard flip mutation is employed with mutation rate p_m in the elements b_j . Thus, if an element b_j is equal to zero, it is flipped to one when it is mutated, what implies in the insertion to the RBF Network of a new radial unit with center at the training pattern \mathbf{x}_j , and with width and q -parameter respectively given by the real numbers r_j and q_j . If an element b_j is equal to one, it is flipped to zero, what implies in the deletion of the radial unit with center in the training pattern \mathbf{x}_j .

When an element r_j or q_j is mutated, its value g_j is changed according to

$$\tilde{g}_j = g_j \exp(\tau_m \mathcal{N}(0, 1)) \quad (24)$$

where τ_m denotes the standard deviation of the Gaussian distribution with zero mean employed to generate the random deviation $\mathcal{N}(0, 1)$. It is important to observe that silent mutations can occur in the elements r_j and q_j when the element b_j is equal to zero.

5.3 Fitness Computation

In the fitness computation, the individual is decoded and the corresponding RBF Network is evaluated. First, the number of hidden units in the RBF Network i (related to individual i), m_i , is defined as the number of ones in the elements b_j of the vector \mathbf{z}_i . Then, the centers of the RBFs are defined. If b_j is equal to one, a center is defined at the training pattern \mathbf{x}_j . The next step is to set the widths and q -Gaussian parameters of each radial unit according to elements r_j and q_j of the chromosome of the individual i . The activations of the hidden units for each instance of the training set and the optimal output weights are then computed (eqs. 2 and 7). Then the RBF Network i can be evaluated.

In this work, the AIC, which evaluates the RBF Network performance and the RBF Network complexity, is employed. In RBF Networks with supervised learning, the AIC [6] is defined as

$$f(i) = N \log \frac{1}{N} \sum_{n=1}^N \left[\left(\hat{\mathbf{y}}(n) - \mathbf{y}(\mathbf{x}(n), \mathbf{z}_i) \right)^T \times \left(\hat{\mathbf{y}}(n) - \mathbf{y}(\mathbf{x}(n), \mathbf{z}_i) \right) \right] + cm_i \quad (25)$$

where $\mathbf{x}(n)$ is the n -th instance of the training set, $\hat{\mathbf{y}}(n)$ is the respective vector of desired outputs, and c is a real number that controls the balance between RBF Networks with small training set errors and with small number of radial units. Here $c = 4$ if $m_{inf} \leq m_i \leq m_{sup}$, and $c = 12$ otherwise. In this way, individual with a small ($m_i < m_{inf}$) or large ($m_i > m_{sup}$) are punished.

6 Experimental Study

In this section, in order to test the performance of RBF Networks with q -Gaussian RBFs (Section 4), experiments with three pattern recognition databases in the Medical Informatics domain are presented. In order to compare the performance of the RBF Network with different RBFs, the same methodology described in last section is applied in RBF Networks with four different RBFs: Gaussian, Cauchy, Inverse Multiquadratic, and q -Gaussian. However, in the experiments with Gaussian, Cauchy, and Inverse Multiquadratic RBFs, the values of the parameter q of each RBF are not present in the chromosome of the individuals of the GA. The shape of the RBFs can change for the RBF Network with the q -Gaussian RBFs (by changing the parameter q), while it is fixed for the experiments with the other three functions.

The databases employed in the experiments were obtained from the *Machine Learning Repository of the University of California - Irvine*. The first database, *Wisconsin Breast Cancer Database* [17], contains 699 instances. Nine attributes obtained from histopathological examination of breast biopsies are considered. Here the RBF Network is employed to classify the instances in two classes, benign and malignant and 16 instances with missing attribute values are discarded [11]. The second database, *Pima Indians Diabetes Database*, contains 768 instances with 8 attributes obtained from clinical information of women of Pima Indian heritage [24]. The objective is to classify the occurrence of diabetes mellitus. In the third database, *Heart Disease Database*, the objective is to classify the presence of heart disease in the patient based on the information of 14 attributes obtained from medical examination. The database originally contains 303 instances, but only 270 are used here [15].

The experiments with the methodology presented in Section 5 applied in the three pattern classification problems presented in the last paragraph are described in the next section. Then, the experimental results and its analysis are presented in Section 6.2.

6.1 Experimental Design

In order to compare the performance of RBF Networks with different RBFs for each database, each GA was executed 20 times (with 20 different random seeds) in the training of each RBF Network. The number of instances in the training set is 50% of the total number of instances, and the same number

is used in the test set. For each run of the GA, the individuals of the initial population are randomly chosen with uniform distribution in the allowed range of values, which are $0.01 \leq r_j \leq 1.0$ for the radial widths and, for the experiments with the q -Gaussian RBF, $0.5 \leq q_j \leq 3.0$.

For all experiments, the population size is set to 50 individuals, the two best individuals of the population are automatically inserted in the next population (elitism), the tournament size is set to 2, $p_s = 0.8$, $p_c = 0.3$, $n_b = 20$ (mean number of radial units in the first generation), $m_{inf} = 15$, $m_{sup} = 60$, $p_m = 0.0025$ (mutation rate for the elements b_j), and $\tau_m = 0.02$ (standard deviation of the Gaussian distribution used to mutate the real elements r_j and q_j). Each GA is executed for 300 generations for the experiments with databases Breast Cancer and Heart Disease and 500 for the experiments with database Pima.

6.2 Experimental Results

The experimental results of the fitness of the best individual in the last generation averaged over 20 runs are presented in tables 1, 2, and 3. The results of the percentage of classification errors for the test sets and the number of radial units of the RBF Network generated from the final best individual averaged over 20 runs are also presented. Figures 3, 4, and 5 show the experimental results of the best-of-generation fitness averaged over 20 runs. From the experiments, some results can be observed and are analyzed as follows.

One can observe that the performance of RBF Networks with different RBFs are different, what was observed in the experiments presented in [13] too. Like in the experiments presented in [13], the influence of the RBF is problem dependent. When the results of the RBF Networks with Gaussian, Cauchy, and Inverse Multiquadratic functions are compared, it is possible to observe that the Gaussian RBFs presents the best performance in the

Table 1. Results of the best individual in the last generation for experiments with Wisconsin Breast Cancer Database

		Radial Basis Function			
		Gaussian	Cauchy	Inv.	Multiq.
		q -Gaussian			
Fitness	Median	-585.94	-696.87	-687.89	-707.08
	Worst	-546.27	-637.17	-640.37	-667.17
	Best	-655.07	-794.26	-726.25	-983.56
	Mean	-591.99	-703.61	-688.31	-732.71
	STD	33.15	52.19	19.82	76.54
Test Set Errors Mean (%)		2.42	1.85	1.79	1.83
	STD	0.58	0.23	0.39	0.54
m	Mean	34.15	25.75	21.20	27.5
	STD	6.09	7.69	4.40	12.91

Table 2. Results of the best individual in the last generation for experiments with Pima Database

		Radial Basis Function			
		Gaussian	Cauchy	Inv. Multiq.	q -Gaussian
Fitness	Median	-428.05	-427.18	-427.79	-428.02
	Worst	-423.97	-422.66	-423.29	-419.92
	Best	-435.06	-430.23	-431.48	-433.57
	Mean	-427.89	-427.34	-427.09	-427.93
	STD	2.67	1.88	2.27	3.47
Test Set Errors	Mean	21.42	20.69	20.64	20.98
	STD	0.77	0.97	0.91	0.99
m	Mean (%)	15.00	15.00	15.00	15.00
	STD	0.00	0.00	0.00	0.00

Table 3. Results of the best individual in the last generation for experiments with Heart Disease Database

		Radial Basis Function			
		Gaussian	Cauchy	Inv. Multiq.	q -Gaussian
Fitness	Median	-120.73	-133.34	-138.76	-136.94
	Worst	-103.92	-128.22	-134.42	-128.53
	Best	-126.72	-143.56	-149.34	-143.71
	Mean	-120.27	-133.93	-139.53	-137.28
	STD	5.35	3.56	3.69	3.89
Test Set Errors	Mean	24.00	16.04	17.59	18.93
	STD	2.73	1.16	1.63	2.98
m	Mean (%)	15.05	15.00	15.00	15.00
	STD	0.22	0.00	0.00	0.00

experiments with database Pima, the Cauchy presents the best performance with database Breast Cancer, while the Inverse Multiquadratic presents the best performance in the experiments with database Heart Disease.

When the shape of the RBFs are compared (Figure 2), one can observe that the Gaussian function presents a higher local activation, which is interesting in the experiments with database Pima, but presents a clear disadvantage in other experiments. It is still possible to observe that the optimization of the widths of the RBFs is beneficial, as the fitness decays during the evolutionary process, but the choice of the RBF generally has a big influence in the fitness performance.

The RBF Network with q -Gaussian RBF presents fitness results better or similar to those obtained by the RBF Network with the Gaussian RBF in the experiment with database Pima and with the Cauchy RBF in the experiment with database Breast Cancer. In the experiment with database Heart Disease, the results obtained by the RBF Network with q -Gaussian RBF are worse

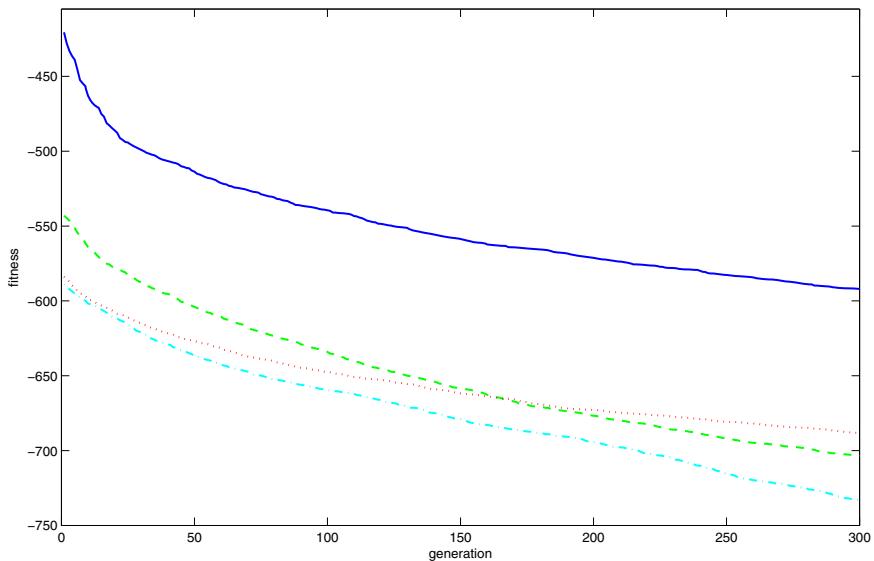


Fig. 3. Mean best-of-generation fitness in the experiments with Wisconsin Breast Cancer Database for RBFs: Gaussian (solid line), Cauchy (dashed line), Inverse Multiquadratic (dotted line), and q -Gaussian (dash-dotted line)

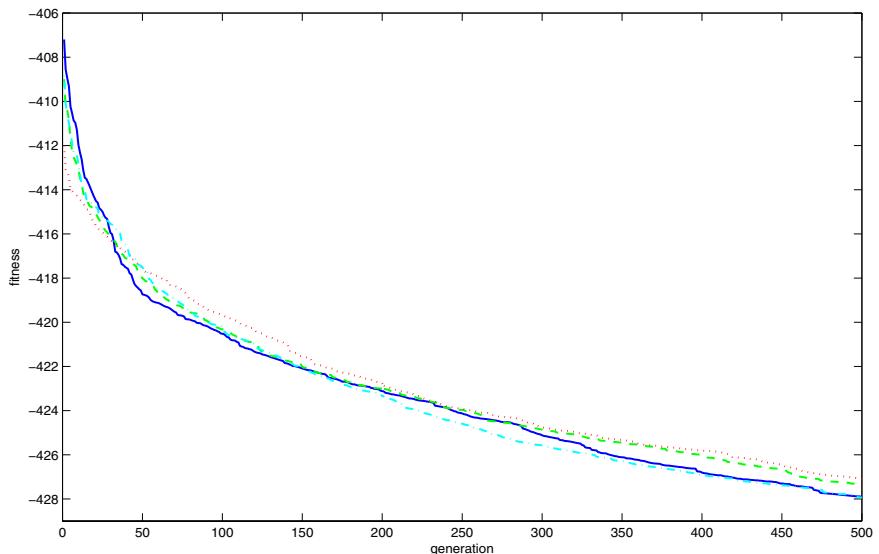


Fig. 4. Mean best-of-generation fitness in the experiments with Pima Database for RBFs: Gaussian (solid line), Cauchy (dashed line), Inverse Multiquadratic (dotted line), and q -Gaussian (dash-dotted line)

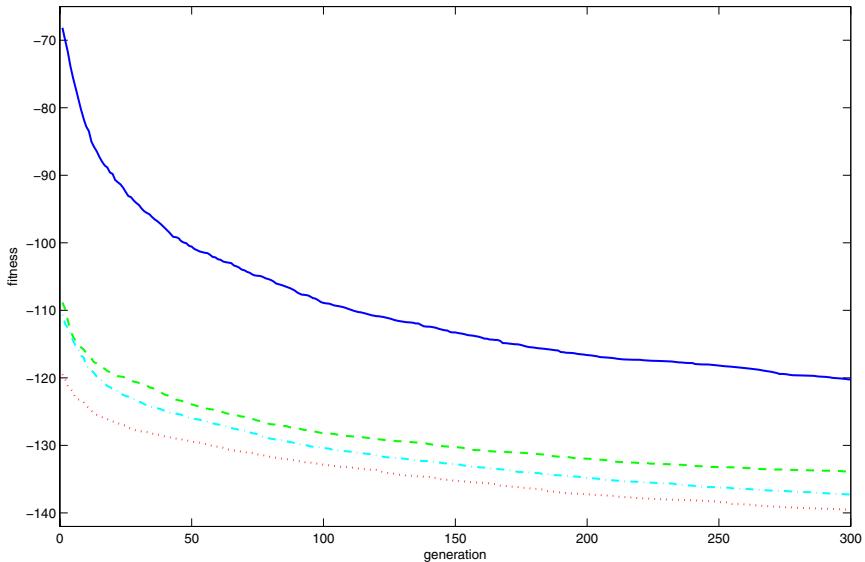


Fig. 5. Mean best-of-generation fitness in the experiments with Heart Disease Database for RBFs: Gaussian (solid line), Cauchy (dashed line), Inverse Multiquadratic (dotted line), and q -Gaussian (dash-dotted line)

than those obtained by the RBF Network with Inverse Multiquadratic RBF. However, the results of the q -Gaussian RBF are better than the results of the Gaussian and Cauchy RBFs.

The good results of the RBF Network with the q -Gaussian RBF can be explained because the GA generally can find good values of the parameter q for the radial units, which generates a better choice for the shape of the RBFs. While the final values of q in the experiments with database Pima are close to 1, i.e., the shape of the q -Gaussian functions is similar to the shape of the Gaussian function, which showed to be a good choice for this database, the values of q are higher in the experiments for the other databases. The better performance of the Inverse Multiquadratic RBF in the Heart Disease Database can be explained by the use of a maximum allowed value of q equal to 3, which is the value of q where the q -Gaussian RBF reproduces the shape of Inverse Multiquadratic RBF (see Figure 2 and Eq. 21). In the experiments with the Heart Disease Database, values of q equal or larger than 3 should be found, which implies in RBFs with longer tails, like the Inverse Multiquadratic RBF. The use of radial units with different RBF shapes (different values of q) is another factor that explains the good performance of the RBF Network with the q -Gaussian RBF.

However, the good performance on the best fitness results does not necessarily imply in good results on the test set errors, what can be observed

in the tables. One can observe that only the patterns of the training set are employed in the fitness function (Eq. 25), what can cause overfitting. In order to minimize this problem, the leave-one-out cross-validation or the generalized cross-validation methods can be used to compute the fitness function [22]. However, the use of such validation methods implies in a larger computational effort.

It is still observable that RBF Networks with small number of radial units are selected by the evolutionary process. From table 2 and 3, one can observe that RBF Networks with 15 radial units (minimum m_i where $c = 4$) are selected for the experiments with database Pima and Heart Disease (one can observe the standard deviation in those experiments is equal to 0, with the exception of one case). The small number of radial units selected by the GA can be explained by the use of Eq. 25, which has a term that evaluates the RBF Network complexity, as fitness function.

7 Conclusions

The use of the q -Gaussian function as a radial basis function in RBF Networks employed in pattern recognition problems is investigated here. The use of q -Gaussian RBFs allows to modify the shape of the RBF by changing the real parameter q , and to employ radial units with different RBF shapes in a same RBF Network. An interesting property of the q -Gaussian function is that it can continuously and smoothly reproduce different radial basis functions, like the Gaussian, the Inverse Multiquadratic, and the Cauchy functions, by changing a real parameter q . Searching for the values of the parameter q related to each radial unit via GAs implies in searching for the best configuration of RBFs to cover the pattern space according to the training set.

The choice of different RBFs is generally problem dependent, e.g., when RBF Networks with Gaussian, Cauchy and Inverse Multiquadratic RBFs are compared in Section 6, each one of these three RBFs presented the best performance in one different experiment. By adjusting the parameter q using the GA, the RBF Network with q -Gaussian RBF can present, in the experiments presented in Section 6, performance similar to those reached by the RBF Network with the RBF that reached the best result. One can observe that the GA can search for the best configuration of the values of q , changing the shape of the RBFs, and allowing good results.

Acknowledgments

This work is supported by *Fundação de Amparo à Pesquisa do Estado de São Paulo* (FAPESP) under Proc. 2004/04289-6 and 2006/00723-9.

References

1. Atakishiyev, N.M.: On a one-parameter family of q-exponential functions. *Journal of Physics A: Mathematical and General* 29(10), L223–L227 (1996)
2. Atakishiyev, N.M.: On the fourier-gauss transforms of some q-exponential and q-trigonometric functions. *Journal of Physics A: Mathematical and General* 29(22), 7177–7181 (1996)
3. Atakishiyev, N.M., Feinsilver, P.: On the coherent states for the q-Hermite polynomials and related Fourier transformation. *Journal of Physics A: Mathematical and General* 29(8), 1659–1664 (1996)
4. Biedenharn, L.C.: The quantum group $\text{suq}(2)$ and a q-analogue of the boson operators. *Journal of Physics A: Mathematical and General* 22(18), I873–I878 (1989)
5. Billings, S., Wei, H.-L., Balikhin, M.A.: Generalized multiscale radial basis function networks. *Neural Networks* 20, 1081–1094 (2007)
6. Billings, S., Zheng, G.: Radial basis function network configuration using genetic algorithms. *Neural Networks* 8(6), 877–890 (1995)
7. Borges, E.P.: A possible deformed algebra and calculus inspired in nonextensive thermostatistics. *Physica A: Statistical Mechanics and its Applications* 340(1–3), 95–101 (2004)
8. Chen, S., Cowan, C.F.N., Grant, P.M.: Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks* 2(2), 302–309 (1991)
9. Floreanini, R., Vinet, L.: Q-orthogonal polynomials and the oscillator quantum group. *Letters in Mathematical Physics* 22(1), 45–54 (1991)
10. Floreanini, R., Vinet, L.: Quantum algebras and q-special functions. *Annals of Physics* 221(1), 53–70 (1993)
11. Fogel, D.B., Wasson, E.D., Boughton, E.M.: Evolving neural networks for detecting breast cancer. *Cancer Letters* 96, 49–53 (1995)
12. Gell-Mann, M., Tsallis, C.: *Nonextensive Entropy - Interdisciplinary Applications*. Oxford University Press, Oxford (2004)
13. Harpham, C., Dawson, C.W.: The effect of different basis functions on a radial basis function network for time series prediction: A comparative study. *Neurocomputing* 69, 2161–2170 (2006)
14. Harpham, C., Dawson, C.W., Brown, M.R.: A review of genetic algorithms applied to training radial basis function networks. *Neural Computing and Applications* 13(3), 193–201 (2004)
15. Liu, Y., Yao, X.: Evolutionary design of artificial neural networks with different nodes. In: Proc. of the IEEE Conference on Evolutionary Computation, ICEC, pp. 670–675 (1996)
16. Maillard, E.P., Gueriot, D.: Rbf neural network, basis functions and genetic algorithms. In: Proc. of the IEEE International Conference on Neural Networks, vol. 4, pp. 2187–2190 (1997)
17. Mangasarian, O.L., Wolberg, W.H.: Cancer diagnosis via linear programming. *SIAM News* 23(5), 1–18 (1990)
18. McAnally, D.S.: Q-exponential and q-gamma functions. i. q-exponential functions. *Journal of Mathematical Physics* 36(1), 546–573 (1995)
19. Mitchell, M.: *An Introduction to Genetic Algorithms*. MIT Press, Cambridge (1996)

20. Moody, J., Darken, C.: Fast learning in networks of locally-tuned processing units. *Neural Computation* 1, 281–294 (1989)
21. Nivanen, L., Le Mehaute, A., Wang, Q.A.: Generalized algebra within a nonextensive statistics. *Reports on Mathematical Physics* 52(3), 437–444 (2003)
22. Orr, M.: Introduction to radial basis function networks. Center for Cognitive Science, Edinburgh University, Scotland, U. K (1996)
23. Rogers, L.J.: Second memoir on the expansion of certain infinite products. *Proceedings of London Mathematical Society* 25, 318–343 (1894)
24. Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., Johannes, R.S.: Using the adap learning algorithm to forecast the onset of diabetes mellitus. In: *Proc. of the Symposium on Computer Applications and Medical Care*, pp. 261–265 (1988)
25. Tinós, R., Terra, M.H.: Fault detection and isolation in robotic manipulators using a multilayer perceptron and a rbf network trained by kohonen's self-organizing map. *Rev. Controle e Automação* 12(1), 11–18 (2001)
26. Tsallis, C.: Possible generalization of boltzmann-gibbs statistics. *Journal of Statistical Physics* 52, 479–487 (1988)
27. Tsallis, C.: What are the number that experiments provide? *Química Nova* 17, 468 (1994)
28. Umarov, S., Tsallis, C., Steinberg, S.: On a q -central limit theorem consistent with nonextensive statistical mechanics. *Milan Journal of Mathematic* (2008), doi:10.1007/s00032-008-0087-y
29. Yamano, T.: Some properties of q -logarithm and q -exponential functions in tsallis statistics. *Physica A* 305, 486–496 (2002)

Part II

**Function Approximation and
Classification: Success Stories and
Real World Applications**

Novel Biomarkers for Prostate Cancer Revealed by (α,β) - k -Feature Sets

Martín Gómez Ravetti, Regina Berretta, and Pablo Moscato

Centre for Bioinformatics, Biomarker Discovery and Information-Based Medicine, The University of Newcastle,
Callaghan, NSW, 2308, Australia

Australian Research Council Centre of Excellence in Bioinformatics

Martin.Ravetti@newcastle.edu.au,
Regina.Berretta@newcastle.edu.au,
Pablo.Moscato@newcastle.edu.au

Summary. In this chapter we present a method based on the (α,β) - k -feature set problem for identifying relevant attributes in high-dimensional datasets for classification purposes. We present a case-study of biomedical interest. Using the gene expression of thousands of genes, we show that the method can give a reduced set that can identify samples as belonging to prostate cancer tumors or not. We thus address the need of finding novel methods that can deal with classification problems that involve feature selection from several thousand features, while we only have on the order of one hundred samples. The methodology appears to be very robust in this prostate cancer case study. It has lead to the identification of a set of differentially expressed genes that are highly predictive of the cells transition to a more malignant type, thus departing from the profile which is characteristic of its originating tissue. Although the method is presented with a particular bioinformatics application in mind, it can clearly be used in other domains. A biological analysis illustrates on the relevance of the genes found, and links to the most current developments in prostate cancer biomarker studies.

1 Introduction

With the label of ‘feature selection’ we encompass several data analysis processes and several types of procedures on which the general objective is to choose, from certain attributes (features) of a given set of samples, those that reveal some particular characteristics present in those samples. These features are generally selected according to certain user-defined criteria, and there are many different approaches we could list. One of the most important common criteria include the need for reduction of the dimensionality of the data, the elimination of irrelevant features for classification, and the selection of relevant features for visualisation and interpretation of high-dimensional datasets. In classification tasks, given a training set of samples with labels

corresponding to different classes, we expect that a good feature selection method will allow us to obtain a better accuracy on predicting the corresponding classes on an independently generated test set of samples.

In this chapter, we are proposing and analysing the performance of a new combinatorial approach (based on a mathematical model called the (α,β) -k-Feature Set Problem) to select the best subset of features, a ‘signature’, that discriminates between two given classes and at the same time improves the performance of the classifiers.

In one recently proposed feature selection taxonomy [1], the approach we will use could be classified as a multivariate filter, because it reduces the dataset size without using any information from the classifier and evaluates a set of features at the same time. An advantage of this method is that it will declare a large number of features as “irrelevant”, by not including them in the feature set finally selected. This means that it will not create a combination of “metafeatures” which will be then of difficult analysis by the biomedical specialist. This is particularly relevant in our bioinformatics and biomarker discovery activities, where all research works would require a follow-up involving a great amount of interdisciplinary effort. We adopt here the definition of biomarker given by the National Institute of Health (US) [2]: “*a characteristic that it is objectively measured and evaluated as an indicator of normal biological processes, pathogenic processes, or pharmacologic responses to a therapeutic intervention.*”

As a test-case for application of our methodology, we will analyse two microarray datasets from prostate cancer samples. This is a challenge for a number of feature selection methods due to the normally low ratio between samples and features. Typically we deal with datasets having one or several hundred of samples and between 10,000 and 40,000 features. Microarrays allow investigators to simultaneously measure the gene expression levels of the whole genome by analysing a blood or tissue samples. With the development of the DNA microarray (or DNA chip) we are witnessing that massive amount of biological datasets are becoming increasingly available and they are useful for investigation with novel methodologies.

There are four main disorders of the prostate: prostatitis, benign prostate hyperplasia, prostarodynia and prostate cancer (PC). It is the latter type of disorder that is the only one which is potentially life-threatening. It is one of the most common types of cancer and it is the second most common cause of death after lung cancer (3,000 men per year) in Australia. Data from Cancer Research UK indicates that: ”Worldwide, more than 670,000 men are diagnosed with prostate cancer every year, accounting for one in nine of all new cancers in males.”¹ It is also one of the most slowly growing cancers and one of its main worrying aspects is that many PCs develop in an asymptomatic way [3]; therefore there is a need for finding novel early detection methods.

¹ [http://info.cancerresearchuk.org/cancerstats/types/prostate/
incidence/](http://info.cancerresearchuk.org/cancerstats/types/prostate/incidence/)

The chapter is organised as follows. In Section 2, we describe our combinatorial approach for feature selection, that uses the (α,β) -k-Feature Set Problem 4 as a mathematical model, as well previous applications of this methodolgy, mostly for microarray data analysis. Computational results and analysis of the prostate cancer datasets are presented in Section 3. Finally, conclusions and discussions are in Section 4.

2 The (α,β) -k-Feature Set Problem

Consider a set of m samples and that each sample has been labelled with the name of one of two possible classes. Each sample will have n values for a set of features. For instance, Table II shows an example with $m = 5$ samples (E_1, E_2, E_3, E_4, E_5) belonging to class Y or W and $n = 7$ boolean-valued features (A, B, C, D, E, F, G), which we will represent with the values 0 or 1 (for False and True, respectively).

Table 1. A numerical example with *seven* features (A, B, C, D, E, F, G) and *five* samples (E_1, E_2, E_3, E_4, E_5) belonging to classes Y or W

	E_1	E_2	E_3	E_4	E_5	
A	0	0	1	0	0	
B	1	1	1	0	1	
C	1	1	0	1	0	
D	1	1	1	0	0	
E	0	1	0	0	0	
F	0	1	1	1	0	
G	0	1	1	0	0	
Class	Y	Y	W	W	W	

Given Table II, we can build the bipartite graph shown in the Figure IIa. In the figure, each black node represents a pair of samples (p, q) that belong to different classes; each of the $n = 7$ white nodes represents a feature i ; and an edge from a white node i to a black node (p, q) exists if the values for the feature i in the pair of samples (p, q) are different. For instance, from Table II, feature A has value 0 in sample E_1 (from class Y) and value 1 in sample E_3 (from class W), so there is an edge from node A to node (E_1, E_3) . In this case we say that the feature A “covers” the pair of samples (E_1, E_3) and then the feature A can help “to explain” why the pair of samples (E_1, E_3) belong to different classes.

The decision problem called *k-Feature Set* [5] asks, given a certain instance, if there exists a set of k features (out of the set of n features given) that can collectively explain why each pair of samples from different classes are not in the same class. Note that the value of k is given as input. Obviously, associated to it there is an optimisation problem, in which k is not given as input. In this optimization variant the aim is to find the set of k features, of

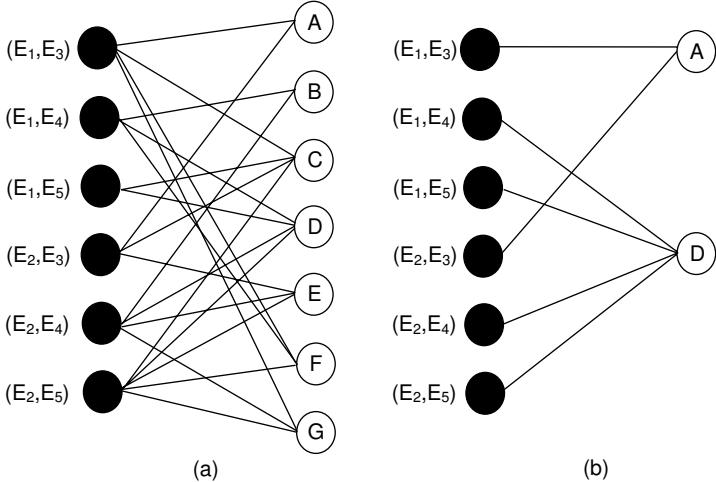


Fig. 1. (a) Bipartite graph built from the example in Table I representing an instance of the k -Feature Set problem. In (b) a 2-feature set. Note that other 2-feature sets also exist ($\{B,C\}$ and $\{C,D\}\}$.

minimum cardinality, that satisfy the requirement of being a k -feature set. If such a minimum k -feature set exists, for the same instance it is now obvious that there is no $k - 1$ -feature set, indicating a kind of natural lower bound on the number of features that can be used to build a statistical model or other type of classifier. In Figure 1b we can see that for the graph constructed using the matrix given in Table I there exists a 2-feature set. We also note that $\{B,C\}$ and $\{C,D\}$ are other two alternative 2-features sets.

We can generalise this basic problem by considering that each pair of samples that belong to different classes should be explained by more than one feature being different. The aim is that this will allow us to obtain more robust signatures and, hopefully, contribute towards a better decision making process. We may also go one step further, and specify that a feature set is also able to explain why two samples are in the same class. This generalisation leads to the (α,β) - k -Feature Set problem [4].

We can extend the graph from Figure 1, adding nodes that represent pairs of samples that belong to the same class (grey nodes in Figure 2). For instance, note from Table I that feature A has the value 0 for samples E_1 and E_2 , both samples belong to class Y . This means that in the graph shown in Figure 2 an edge from node A to the grey node (E_1, E_2) has to be added.

As the k -Feature Set problem has a given value of k as a parameter, the (α,β) - k -Feature Set problem has two extra positive valued parameters, namely $\alpha \geq 1$ and $\beta \geq 0$. We now aim at finding a k feature set that has the property that every pair of samples from different classes is “explained” by at least α features of the k -feature set. We also require a high degree of internal consistency within a class. We require that for any pair of two samples that

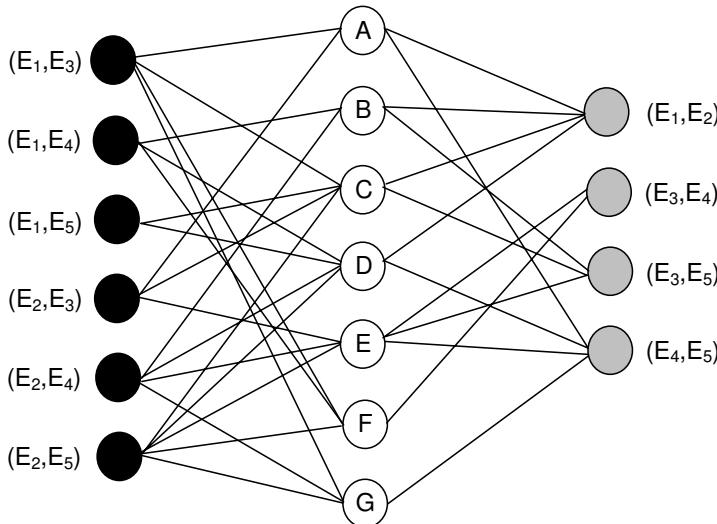


Fig. 2. Graph built from the example in Table II representing an instance of the (α, β) -k-Feature Set problem

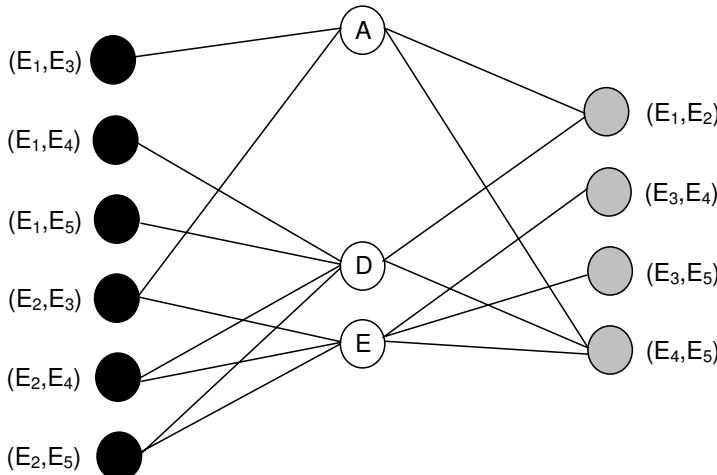


Fig. 3. A feasible solution for the instance of the (α, β) -k-Feature Set problem shown in Figure 2 for $\alpha = 1$ and $\beta = 1$

are in the same class at least β features of the k -feature set need to have the same value. For example, if we consider $\alpha = 1$ and $\beta = 1$, a feasible solution for the instance shown in Figure 2 can be the set $\{A, D, E\}$ (Figure 3). Note that the sets $\{C, D, E\}$, $\{B, C, E\}$, $\{C, D, F\}$, $\{D, E, F\}$ and $\{C, E, F\}$ are other feasible solutions with three features each. For $\alpha = 2$ and $\beta = 2$, a

feasible solution is shown in Figure 4. Note that if $\alpha = 1$ and $\beta = 0$, we have the k -Feature Set problem.

Once again, we have an optimization version associated to this problem. In this case, now with user-given fixed values of α and β , we can aim to minimize the number of features of the k -feature set. We can formulate the optimisation version of this problem as an integer programming model as follows. Let n be the number of features; the objective function we seek to minimise is given by (1), where variable x_i assumes the value 1 if the feature i is selected to be in the k -feature set, and it is zero otherwise. The constraints discussed before are formalised with linear inequalities. Let $a_{ipq} = 1$ if feature i has different values for a pair of samples p and q that belong to different classes ($C_p \neq C_q$); and $a_{ipq} = 0$ otherwise. Analogously, b_{ipq} is 1 if feature i has the same value for the pair (p, q) belonging to the same class ($C_p = C_q$); and zero otherwise. Each black node has a constraint (see (2)) associated with it, which guarantees that each pair of samples (p, q) has at least α edges that link it to nodes which represent features in the k -feature set. Similarly, for each grey node we have a constraint (given by (3)) that guarantees that each pair of samples that belongs to the same class has at least β edges linking it to features in the k -feature set.

$$\text{Min} \sum_{i=1}^n x_i \quad (1)$$

$$\sum_{i=1}^n a_{ipq} x_i \geq \alpha \quad \forall (p, q) \quad C_p \neq C_q \quad (2)$$

$$\sum_{i=1}^n b_{ipq} x_i \geq \beta \quad \forall (p, q) \quad C_p = C_q \quad (3)$$

$$x_i \in \{0, 1\} \quad (4)$$

In terms of the computational complexity of these problems, we can say that the (α, β) - k -Feature Set problem is NP-complete, since the k -Feature Set problem was proven to be NP-Complete by Davies and Russel [6], and is in NP. In addition, Cotta and Moscato proved in 2003 that the parameterized version of the k -Feature Set problem (when the parameter is the cardinality of the feature set) is $W[2]$ -Complete [5]. This result is very important as it provides a tight lower bound in terms of Parameterized Complexity Theory, a field pioneered by R. Downey and M. Fellows [7]. Despite of the computational complexity of the problem, both in the classical and parameterized domain, it is possible to use some safe reductions rules that help to reduce large instances such that they can be solved optimally using, for instance, CPLEX ², a mathematical programming commercial software, which has complete exact methods to solve Integer Programming Models like (1)-(4). The description of the reduction rules for the (α, β) - k -Feature Set problem can be found in [8].

² <http://www.ilog.com/products/cplex>

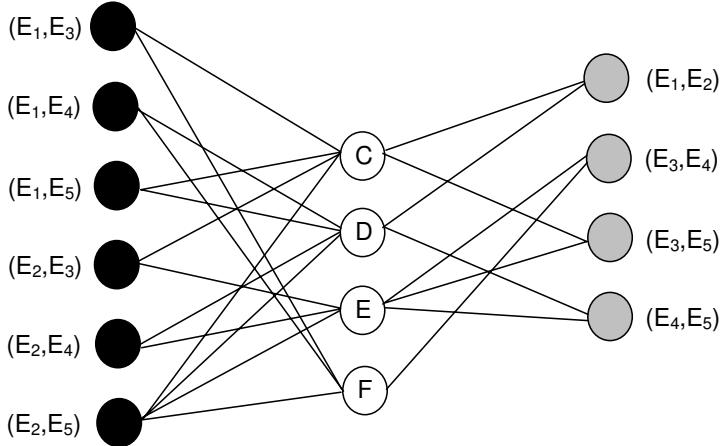


Fig. 4. A feasible solution for the instance of the (α, β) -k-Feature Set problem shown in Figure 2 for $\alpha = 2$ and $\beta = 2$

2.1 The Max Cover (α, β) -k-Feature Set Problem

For the instance shown in Figure 3, if we consider $\alpha = 1$ and $\beta = 1$, then there are six different solutions with minimum cardinality ($k = 3$): they are the sets $\{A, D, E\}$, $\{C, D, E\}$, $\{B, C, E\}$, $\{C, D, F\}$, $\{D, E, F\}$ and $\{C, E, F\}$. Among all optimum solutions for the model (1)-(4), we may yet have another preferred metric to optimise. Note that each feature “explains” a different number of pair of samples (the degree of the white node). For example, the feature A “explain” 2 pairs of samples that belong to different classes (black nodes) and 2 pairs of samples that belong to the same class (grey nodes). One possibility is to choose the solution that maximise the “explanation”, i.e. the solution that has as a property that the sum of the degrees of the features nodes selected is maximum. Note that the solutions $\{A, D, E\}$, $\{B, C, E\}$, $\{C, D, F\}$, $\{D, E, F\}$ and $\{C, E, F\}$, the sum of the degrees is 16, and for the solution $\{C, D, E\}$, this value is 18, so this should be preferred. The Integer Programming model for this new problem can be written as follows.

$$\text{Max} \sum_{i=1}^n d_i x_i \quad (5)$$

$$\sum_{i=1}^n a_{ipq} x_i \geq \alpha \quad \forall (p, q) \quad C_p \neq C_q \quad (6)$$

$$\sum_{i=1}^n b_{ipq} x_i \geq \beta \quad \forall (p, q) \quad C_p = C_q \quad (7)$$

$$\sum_{i=1}^n x_i = k \quad (8)$$

$$x_i \in \{0, 1\} \quad (9)$$

The objective function (5) maximises the weighted sum of the features which is given by the number of pair of samples that feature i “explains”. The constraints (6) and (7) are the same as those used in the previous model. The constraint (8) guarantees that the number of features in the solution will be k , which is given by the resolution of the model (11)-(14). We note that the weights d_i can be selected with other criteria, for example using a statistical score or by other means. We will aim to explore other alternatives in the future and we limit ourselves in this chapter to explore the results using the total number of “explanations” of the feature set.

2.2 A Centric Approach

When we build the graph for the (α, β) -k-Feature Set problem, the grey nodes represent pair of samples that belong to the same class. The objective we seek in constructing the graph this way is to bias the search to obtain feature sets that are relatively homogeneous in the values they present for samples of the same class. However, there are situations in which we want to bias the selection of features that help to explain why two samples belong to a specific class. For instance, we may want to bias the homogeneity of the samples belong to class W in Table II. A simple modification in the construction of the graph can be done by only including the nodes that represent the pair of samples that belong to class W (grey nodes in Figure 5). The bias is then achieved by not considering the pair of samples that belong to class Y .

Once again, consider the example from Table II with the respective graph shown in Figure 5. Consider as well $\alpha = 2$ and we will identify which is the

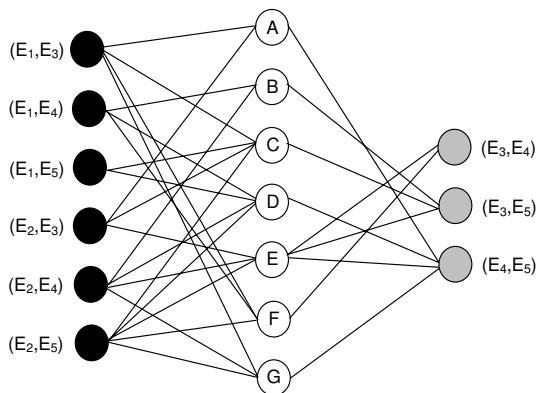


Fig. 5. Graph built from the example in Table II representing an instance of the (α, β) -k-Feature Set problem, but considering grey nodes only from class W

maximum value that β can reach. As we previously remarked, the solution of the model (5)-(8) is the set $\{C, D, E, F\}$ for $\beta = 2$. If we take into account class W only, the optimal solution of the model (5)-(8) will be the same ($\beta = 2$, unchanged). However, when we take into account only class Y , a different graph is generated and the β value can reach $\beta = 4$. In this case the optimal solution is the set $\{A, B, C, D\}$. Notice that using the centric approach, the features A and B belong to the optimal solution, since they are good to explain why the samples 1 and 2 belong to the class Y .

2.3 Previous Applications of the (α,β) -k-Feature Set Problem

Most applications of the (α,β) -k-Feature Set problem were employed to assist microarrays data analysis or other type of biomedical datasets. The only exception is the study presented in [9], where it shows an illustrative example on the application of the (α,β) -k-Feature Set problem in a study that led to rules that successfully predicted the 2004 U.S. Presidential election's outcome, several months ahead of the election, only based on historical information from previous elections.

For microarray data analysis, the first challenge we face is that, contrary to the case of U.S. Presidential election or the examples discussed until now in the chapter, microarray datasets have continue values as measurements of gene expressions. They are not discretized values and the (α,β) -k-Feature Set problem as a model inherently assumes that a feature can only be in a relatively small number of possible states. To solve this problem, in our first studies presented in [10] and [11], we used information from the average and the standard deviation of the expression values over all features (microarray gene probes in this case) and samples to perform the discretization. However, using this approach, features for which the expression values are located within a small interval may be considered not very relevant as discriminators. Another option is to use a methodology that makes an independent analysis for each feature to define a threshold. Towards this end, we used an entropy-based heuristic introduced by Fayyad and Irani [12]. Their method evaluates the amount of information that can be gained when a set of samples are divided into two non-empty sets. It also excludes features that cannot provide a minimum of information to organise the classes, by applying a test based on the Minimum Description Length principle. A detailed description of this method can be found in [13].

Applications of the (α,β) -k-Feature Set problem for microarray data analysis can be found in [10, 11, 8, 14, 15, 13, 16]. In all cases, we have used datasets available in the literature and in the public domain, which is also useful for reproducibility purposes. The first step is the discretization procedure, where [10, 11] used information on the average and the standard deviation and in [8, 14, 15, 13, 16] we used Fayaad and Irani method. Note that when we apply the Fayaad and Irani method, a subset of features (probes in this case)

can be early eliminated for further consideration because of the Minimum Description Length Principle test.

With a dataset composed by discrete values, we then apply the (α, β) -k-Feature Set problem. The approach we have been using is the following. First, we check which is the maximum value α that guarantees that at least there is a k -feature set for that instance of the problem (we name it α_{max}). Then, we solve the optimization version of the (α, β) -k-Feature Set (minimizing the number of features) for $\alpha = \alpha_{max}$ and $\beta = 0$. In that way, we bound the tight lower bound on the number of features k_{opt} that are required to distinguish any pair of samples belonging to different classes with at least α_{max} different feature values. Next, we find the maximum value of β , such that the cardinality of the feature set found (k_{opt}) before is the same ($\beta_{maximal}$). We note that the feature set that does this is not necessarily the same found before. Finally, we solve the Max Cover (α, β) -k-Feature Set Problem using $\alpha = \alpha_{max}$ and $\beta_{maximal}$. For microarray data analysis, in the end of this approach we have a subset of features (probes), which we called the “*genetic signature*”. This approach can help to understand the basic phenomena behind the differential levels of expression of the genes.

The main advantage of this approach against other types of algorithms based on rankings is that here we are looking to the best set of features (for a given α) that discriminate between the classes instead of just selecting the first group of “good” features of a ranking that may not work well together as a signature. Our method provides a mathematical guarantee that any pair of samples that belong to different classes is discriminated by the values at least $\alpha = \alpha_{max}$ features.

In [11, 13, 16] the proposed approach was applied on a different gene expression and protein abundance data sets from brain and blood samples of Alzheimer’s disease diagnosed patients. In [11] the methodology was compared with an Evolutionary Search and statistical methods. The results showed a clear pattern of differential gene expression. In [16] and using the proposed integrative data analysis method, we uncovered a robust 5-protein signature capable to predict if a patient³ will develop Alzheimers disease, with an overall accuracy of 96%. In [10] and [8], we illustrate on the usefulness of the (α, β) -k-Feature Set problem using a dataset known as NCI60 from Cancer Microarray Project, Stanford, available online [17], with the gene expression of 5 different types of cancer (renal, ovarian, leukaemia, colon and melanoma) in 41 cell lines. The results showed a good balance of discrimination between classes as well as a within-class consistency, which complements the conclusions from [17]. In [15] and [14] we applied the methodology on a dataset contributed by [18] containing the gene expression taken from 40 different regions of the brains of a Parkinson-affected rodent and of a control which is labeled “normal”. We compare the results found by our approach with p -values calculated using the Wilcoxon test.

³ Patients with no cognitive impairments. See [16] for details.

3 Case of Study: Finding Biomarkers for Prostate Cancer

With the specific objective of detecting biomarkers based on gene expression data, we are looking for a *small* genetic signature that can help us to accurately classify if a patient has Prostate Cancer (PC) or not. A small, yet robust, set of biomarkers may motivate the validation with other different techniques and be amenable for use on the clinical setting. Here we diverge from the method we described previously and we work not to find α_{max} -based genetic signatures. For the two datasets we studied in this chapter α_{max} take the values of 181 and 1,220 respectively. Those genetic signatures are useful to identify whole pathways which may be deregulated due to the disease process. Instead, in this chapter we will illustrate how we can find small signatures and their relevance for the prediction task.

Next, we describe the datasets we have used, the methodology we have applied and computational results we have achieved. An in-depth biological discussion of the biomarkers supports the relevance of the observed changes in gene expression to discriminate normal and prostate cancer samples. Interestingly enough, the biomarkers found for one of the datasets seem to indicate that normal tissue adjacent to the tumor seems to have also had their expression profile changed for these key biomarkers, which opens the possibility that some of these biomarkers can be used as the basis of novel imagining technologies.

3.1 Datasets

The first dataset has originated from two different research papers: the training set from [19] is composed by 102 prostate samples, 52 tumours and 50 non-tumours samples referred to as normal from now on. Expression profiles for each sample were obtained by using oligonucleotide microarrays containing 12,600 probes (more details in [19]). The associated test set has 34 samples, 25 tumours and 9 normal. The technology used for this experiment was the same but with approximately 10 times higher hybridisation intensity. The complete dataset (training plus test set) was already used in some studies [20, 21, 22], and it is available at the Kent Ridge Biomedical Data Set Repository⁴.

The second dataset is composed by three complementary studies [23, 24]. In this case, cRNA was prepared and hybridised to Affymetrix GeneChip HG-U95av2, HG-U95b and HG-U95c arrays (we refer to [24] for a detailed explanation on the cRNA preparation and gene expression assays). The raw data was downloaded from NCBI Data Set Record⁵ and after combining the three experiments we obtained a dataset with 161 samples and 37,690 probes. In these experiments the samples of tissues considered are: tumours tissue,

⁴ <http://leo.ugr.es/elvira/DBCRepository/index.html>

⁵ www.ncbi.nlm.nih.gov/geo

Table 2. Summary of the datasets used for the computational experiments

	Training sets	Test sets
Dataset 1 (12,600 probes)	102 samples (54PC + 50N)	34 samples (25PC + 9N)
Dataset 2 (37,690 probes)	73 samples (60PC +13N)	31 samples (4N+27PC) 57 samples (APC)

normal tissue from a donor, normal tissue adjacent to tumours and metastatic tissue. As the main purpose of this chapter is to analyse the application of (α,β) -k-Feature Set Problem for classification, we randomly divided the last dataset into a training set and two test sets. The training set contains 73 samples of two classes, patients with cancer (PC) or without cancer (N). We only considered normal tissue from donor as N (we did not consider as ‘N’ the normally appearing tissue samples adjacent to tumours) and tumours and metastatic tissue as PC. The first test set has the same characteristics with 31 samples (4 class N and 27 class PC). The second test set has all the 57 samples of normal tissue adjacent to tumour (APC). The main goal of the second test set is to analyse how effective is the group of biomarkers. By effective we understand the capacity of predicting that a patient has prostate cancer even when analysing normal appearing tissue next to tumour. Table 2 summarises the information about the two datasets used in the computational experiments.

3.2 Methodology

Our methodology consists of a data analysis method that integrates the steps we have described before in the chapter. We use three main steps: discretization of expression values, using Fayyad and Irani’s method described earlier; application of the (α,β) -k-Feature Set problem for feature selection and finally, a classification procedure.

After the first step (discretization of expression values), we have an instance of the (α,β) -k Feature Set problem. As previously discussed, the value of the parameter α allows us to select how robust our signature needs to be. We understand that the robustness of a solution is associated with the minimum number of features used to discriminate any pair of samples belonging to different classes. Higher α values select a larger number of features, which forms a signature especially useful when trying to understand underlying phenomena behind the gene expression levels. Smaller α values give to the user the opportunity of finding smaller signatures but with a still good level of discrimination between classes.

In this chapter we are looking for a small number of biomarkers, so the size of the signature is equally important as its performance for classifying the test set. In both datasets we are going to work with small α values and high β values. By fixing α we are also fixing the size of the signature (k). As previously discussed, a high β helps to select features that characterised the class in analysis (centric approach). In our experiments we want to use the highest β value without increasing the size of the signature (k), which we called $\beta_{maximal}$. After repeating the experiment for both classes the final signature is the union of all the selected features. By making the union of the signatures we are including all the important genes from both classes. It is important to remark that all the feature selection process must be performed without using any information from the test sets and they must only be used to evaluate the classification performance.

For the final classification we use the Prediction Analysis for Microarray software (PAM) [25]. The software allows us to use SAM [26] (Significance Analysis for Microarray) to perform a final selection of the genes (from the set selected by the (α,β) -k Feature Set) and the Nearest Shrunken Centroids Algorithm (NSC) to classify the samples. SAM assigns a score to each feature based on a non-parametric analysis and by using an error examination we are able to select the final group of features to be used in the classification process. The selection of the NSC as a classifier was based on our previous experience with it [27] and as a baseline for comparison and ease of replication of our findings. The NSC algorithm first calculates the shrunken centroid for each class considering the training set. Then the classification takes place by analysing the gene expression profile of each sample on the test set and computing the squared distance to each class centroid; the sample is then classified as belonging to the closest class, for details on this algorithm readers may refer to [25, 28].

3.3 Computational Results

The computational results are presented in two parts, each one using one of the datasets presented previously. We also present a discussion about the genes found with our methodology.

First experiment

The training set for the first dataset contains 12,600 features and 102 samples. After the discretization using Fayaad and Irani method, we ended with a 1,566 features from where the (α,β) -k Feature Set with $\alpha = 2$ and $\beta_{maximal}$, was able to create two 7-gene signatures, each of them centric in one of the two classes considered (PC with $\beta_{maximal} = 3$ and Normal with $\beta_{maximal} = 2$). The final signature consists of the union of both signatures, which in this case we have a total of 14 genes (all the selected features were different).

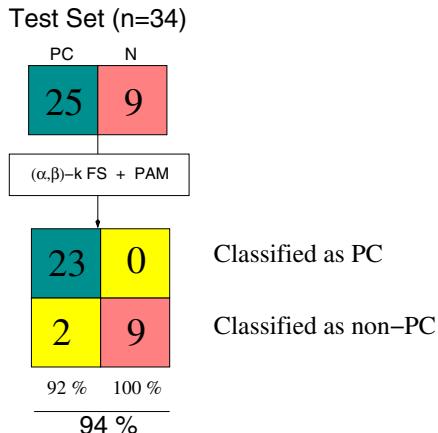


Fig. 6. Confusion matrix for the first experiment. The test set has a total of 34 samples, 25 PC and 9 normal. 23 out of 25 were correctly classified as PC and 9 out of 9 were correctly classified as normal.

By applying SAM and the NSC algorithm with a threshold of 3.88, only 4 gene probes (PTGDS, HPN, GSTM-1/2/4⁶, NELL2) were used to classify the test set with only 2 errors (misclassifications) over a total of 34 samples. Figure 6 shows the confusion matrix for this experiment with an overall accuracy of 94%, 100% of accuracy when predicting a normal patient and 92% of accuracy when predicting prostate cancer.

The same dataset was already used in three recent publications [20, 21, 22]. Their and ours results are compared in Table 3. For Orsenigo [20] and Shah and Kusiak [21] we select the best results presented in their research works. It is worth remarking that we selected the best results from [20, 21] and as a consequence we may be somehow overestimating their true performance. Our results are compared to those in Table 3. This selection is made after knowing the final results; therefore they are not good predictors of the performance of their methodology.

On the same table we also include the results of applying the SAM followed by the NSC algorithm without using (α,β) -k-Feature Set Problem for the feature selection process. For the latter experiment we used the PAM software and the threshold of 7.47. The threshold is selected manually and we choose the threshold that presents the best performance on the training set (small error) and the smallest signature (when the best performance is not unique). Regarding the last test using PAM, it is possible to see how the performance of PAM can be improved by applying the (α,β) -k-Feature Set Problem as a feature selection procedure.

Table 4 presents the selected genes with probes id, gene symbol and gene names and next, a discussion about the genes probes selected are presented.

⁶ The genes GSTM1, GSTM2 and GSTM4 share the same probe (556_s_at).

Table 3. Results comparison between our methodology and three recent works in terms of accuracy to classify the test set. Two classes are considered, prostate cancer (PC) and Normal (N), the results are compared using the number of errors and accuracy for each class and the overall performance.

	PC Class	N Class	Overall
(α,β) -k FS + PAM	2 (92%)	0 (100%)	2 (94%)
Wang [22]	11 (56%)	1 (89%)	12 (65%)
Orsenigo [20]	1 (96%)	1 (89%)	2 (94%)
Shah [21]	11 (56%)	0 (100%)	11 (68%)
PAM	0 (100%)	9 (0%)	9 (73%)

Table 4. Group of genes uncovered considering the first dataset

Probe ID	Gene Symbol	Gene Name
38406_f_at	PTGDS	Prostaglandin D2 synthase 21kDa (brain)
37639_at	HPN (Hepsin)	Hepsin (transmembrane protease, serine 1)
556_s_at	GSTM1/2/4	glutathione S-transferase M1/M2/M4
32598_at	NELL2	NEL-like 2 (chicken)

Biological analysis of the biomarkers found

PTGDS (prostaglandin D2 synthase 21kDa (brain)), was first identified in the brain of mammals (also known as Cerebrin-28, PDGS2, and Glutathione-independent PGD synthetase) and it is assumed that is a PGD(2)-producing enzyme and a retinoid transporter. It is considered as a key enzyme for the metabolism of the arachidonic acid [29] and catalyses the isomerization of PGH(2) to PGD2 [30]. It has been observed that PTGDS may be a chemotherapeutic agent in ovarian cancer [31]. Banerjee *et al.* suggest that the anti-inflammatory action of prostaglandin D(2)-like analogues could be useful for the prevention or delay oral epithelial carcinogenesis [29]. In 2005, Kim *et al.* [32], from the M.D. Anderson Cancer Center, Houston, Texas, have confirmed that PTGDS and prostaglandin D2 (PGD2) metabolites which are produced by normal prostate stromal cells are able to inhibit tumor cell growth. They link it to a peroxisome proliferator-activated receptor gamma (PPARgamma)-dependent mechanism which, in the prostate microenvironment, gives rise to a putatively endogenous mechanism involved in tumor suppression that potentially contributes to the indolence and long latency period of this disease [32]. The PPARgamma link was also strengthened by a recent result by Eichele *et al.* [30] in cervical carcinoma cells (HeLa). They also showed that chemotherapeutics-induced apoptosis was prevented by siRNA targeting PTGDS, indicating that the induction of COX-2 and the synthesis of PTGDS-derived, PPARgamma-activating prostaglandins could be a mechanisms by which several chemotherapeutics (for example, paclitaxel, cisplatin and 5-fluorouracil) can induce apoptosis [30]. These, and other results

[33, 34, 35, 36, 37] indicates that impaired prostaglandin biosynthesis may be linked to tumor progression in prostate cancer.

The presence of GSTM4 in our reduced feature set might not be a surprise for oncologists and molecular biologists. The Glutathione S-transferase (Gst) enzymes are key protectors of the cells macromolecules against both electrophiles and oxidative stress. The GSTM4 gene product belongs to a class (phase II metabolism enzymes) which is frequently involved in molecular studies of detoxification of various drugs [38] and carcinogens (like insecticides, herbicides [39, 40, 41, 42]). It is perhaps not surprising that a polymorphism in this gene has been associated with lung cancer [43].

The role of NELL2 is perhaps the one that is the least understood from the point of view of current biological knowledge. NELL2 is expressed predominantly in brain, but until 2001 it was not reported to be expressed in the prostate. Investigations conducted at Merck Research Laboratories have found that NELL2 mRNA expression is predominantly localized in basal cells of the epithelium, and the discovery that is highly expressed in benign prostatic hyperplasia [44, 45] suggested that it may be a novel prostatic growth factor [44]. Here, however, it is mainly downregulated in prostate tumors in our set. DiLella *et al.* suggest that changes on the levels of NELL2 may lead to alterations in epithelial-stromal homeostasis. Very little is known about its role in cancer, with the exception of its recent inclusion in a selected list of 14 genes regulated by hepatocyte nuclear factor 4 alpha (HNF4 alpha) in renal cell carcinoma. The list, which also includes CDKN1A (p21), suggests that it has a role in the control of cell proliferation [46].

All the biomarkers described before are downregulated in prostate carcinoma as compared with normals. The HPN is the only one from our set that is upregulated. HPN (Hepsin hepsin, transmembrane protease, serine 1, Serine protease hepsin, TMPRSS1) has been found to be upregulated not only in prostate [47], but also in renal, ovarian cancer [48] and it is currently under heavy investigation to its possibility to be used as the target of imaging agents [49]. This biomarker seems to be very robust, which means that its upregulation has been seen in several microarray datasets [50, 51, 52] and other types of studies ([53, 54, 55]). It is interesting that has come up in our study using a single dataset. In addition, the significant association of one single nucleotide polymorphism with Gleason score found by Pal *et al.* indicates a role in prostate cancer risk [56] (see also [57]).

Second experiment

For the second dataset we have one training and two different test sets, each of them containing 37,690 features. Once again our goal is to find biomarkers to classify a patient as PC or N ('PC' for 'Prostate Cancer' or 'N' for 'Normal'). For that reason, on this dataset we removed from the training set all the samples of normal tissue near to tumour, because they belong to patients with PC and its genetic profile is expected to be already altered.

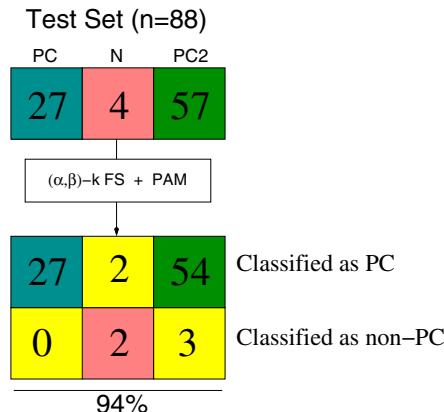


Fig. 7. Confusion matrix for the second experiment. Both test sets have a total of 88 samples, 84 PC and 4 normal. 81 out of 84 were correctly classified as PC and 2 out of 4 were correctly classified as normal.

Using the same methodology, we first run the Fayyad and Irani method that removed 34,245 features. With the 3,445 features remaining we solve the (α,β) -k Feature Set with $\alpha = 5$ and $\beta_{maximal}$, and we uncovered two centric genetic signatures having each seven probes. The union of both signatures produced a 10-gene/probe signature showed in Table 5.

By using the PAM software with a threshold equals to 0, only 2 misclassifications were made in each test set. The threshold equal to zero indicates that the classifier uses all the features to classify the test set. On the first test set, we have an accuracy of 100% for predicting PC and a lower 50% for predicting N. We believe that the low specificity value is due to the small number of normal samples considered (only 4). More tests are needed to clarify the analysis. The second test set contains a unique class containing 57 patients with APC (normal tissue sampled adjacent to tumour). Even when looking at a case with this level of difficulty, the algorithm classifies 54 out of 57 patients as PC. The confusion matrix for the whole experiment is presented in Figure 7. Table 5 introduces the selected genes and next we present a discussion about the genes probes selected.

Biological analysis of the biomarkers found

The transcription factor JUND (jun D proto-oncogene) is upregulated in this dataset of prostate tumors samples. This fact once again point at the presence of unusually high levels of reactive oxygen species. In eukaryotic cells, JUND is a mediator of stress responses, a promoter of differentiation and an inhibitor of cell proliferation and in the androgen-dependent prostate cancer cell line LNCaP it has been suggested as a mediator of the androgen-induced increase in reactive oxygen species [58, 59]. Also using LNCaP, Polytarchou *et al.* observed that exogenous hydrogen peroxide, perhaps one of the major

Table 5. Groups of genes uncovered considering the second dataset

Probe ID	Gene Symbol	Gene Name
38827_at	AGR2	Anterior gradient 2 homolog
56161_at	MLPH	Melanophilin
676_g_at	IFITM1/2/3	Interferon induced transmembrane protein 1, 2 or 3
41483_s_at	JUND	Jun d proto-oncogene
41745_at	IFITM3	Interferon induced transmembrane protein 3
35276_at	CLDN4	Claudin 4
57239_at	TRB1	Tribbles homolog (drosophila)
61614_at	WDR1	Wd repeat domain 1
43827_s_at	LDHB	Lactate dehydrogenase b
63885_at	ZRANB2	zinc finger, RAN-binding domain containing 2

reactive oxygen species, stimulates proliferation and migration of these cells through in a mechanism that involves the activation of activator protein-1 and JUND binding [60].

AGR2 (anterior gradient 2 homolog (*Xenopus laevis*)) is an androgen-inducible secretory protein has been previously shown to highly expressed not only in prostate [61, 62, 63] but also in adenocarcinomas of the pancreas, breast and esophagus [64]. While the prognostic significance of its gene expression is debated ([65, 62]), it has been highlighted as a biomarker of interest for the identification of circulating tumor cells (CTCs) in peripheral blood.

Upregulation of CLDN4 in primary and metastatic tumor samples has been very recently observed at both RNA and protein levels [66]. CLDN4 codes for an integral membrane cell-junction protein and with CLDN3 they play a key role in the control and selectivity of paracellular transport [67]. CLDN4 is the receptor for *Clostridium perfringens* enterotoxin. CLDN4 is also up-regulated in cancers of the bladder, thyroid, fallopian tubes, ovary [68] (but we also note a recent comparative result with other claudins in [69]), stomach, colon, breast, uterus [70, 71, 72, 67], prostate [71, 72], pancreas [73, 74], lung adenocarcinoma cells and type II alveolar pneumocytes [75], intrahepatic cholangiocarcinoma [76] and squamous cell carcinoma of the tongue [77].

C8FW/TRIB1 (tribbles homolog 1 (*Drosophila*)) has been recently proposed as a putative tissue and blood biomarker to predict chronic antibody-mediated rejection, which is an active immune-mediated form of chronic allograft failure associated with a poor prognosis [78], a cooperating agent in acute myeloid leukaemia [79, 80, 81], overexpressed in follicular cancer in malignant thyroid nodular tissue [82], linked to ovarian cancer [83].

Also presented high expression in our signature is ZRANB2/ZNF265 which has been previously shown to be upregulated in grade III ovarian serous papillary carcinoma [84].

Some genes are downregulated including a probe that belongs to either IFITM3 (interferon induced transmembrane protein 3 (1-8U)) or IFITM2

(interferon induced transmembrane protein 2 (1-8D)), WDR1 (WD repeat domain 1 Actin-interacting protein 1, AIP1), and LDHB (lactate dehydrogenase B). The latter is a gene that has also been reported as downregulated in the highly metastatic prostate cancer cell line LNCaP-LN3 when compared with the less metastatic LNCaP [85, 86].

4 Discussion and Conclusion

The two molecular signatures that our method has uncovered have shown to be good predictors of the change of the molecular profile of the normal prostate cells to a malignant phenotype. Some of them, like CLDN4, are also interesting since they can offer the possibility of being used for new imaging procedures. This would be very beneficial, for example, to identify tissues that look normal but may have already changed in a significant way. It is also remarkable that our second signature has been able to predict (as “prostate cancer-derived samples”) a total of 54 out of 57 that are “normally appearing” but are next to tumor tissue. This may indicate that there are genes that are differentially expressed, here identified as putative biomarkers, and which may be related to primordial changes. As a consequence, this discovery may help for early intervention procedures. This hypothesis, that there are significant changes in normally appearing tissue next to a tumor, will require further investigation.

In spite of the success in finding these biomarkers, we may question ourselves: “*are there unifying mechanisms that have been uncovered by these biomarker identification processes ?*”. There is no easy answer for this question but we can provide a glimpse of the way ahead. Using the online systems identification bioinformatics tool call GATHER⁷, and according to TRANSFAC [87], 13 of the 14 genes from the union of the two signatures (the set composed of PTGDS, GSTM, GSTM2, GSTM4, NELL2, HPN, JUND, AGR2, CLDN4, C8FW, ZRANB2, WDR1, and LDHB) has an overrepresentation of binding motifs for MAF (v-MAF musculoaponeurotic fibrosarcoma oncogene homology (avian), Hs.30250).

The name MAF does not immediately recall a well-known association with prostate cancer, so we conducted a literature search on it. It heterodimerizes with FOS and JUN [88]. We found that it has been previously reported as associated with PDGFR-beta (platelet-derived growth factor receptor beta), a receptor tyrosine kinase overexpressed in a subset of prostate cancers [89]. Another discovery is perhaps more central to understand this disease, we also found that results on animal studies have previously shown that MAF is a gene that is known to be responsive to androgen. The criteria of its selection for a selected panel in Ref. [90] is very clear: “(i) overexpression after androgen treatment; (ii) described to be androgen responsive in different species; and (iii) reported in independent studies”. The Italian team aimed

⁷ <http://gather.genome.duke.edu/>

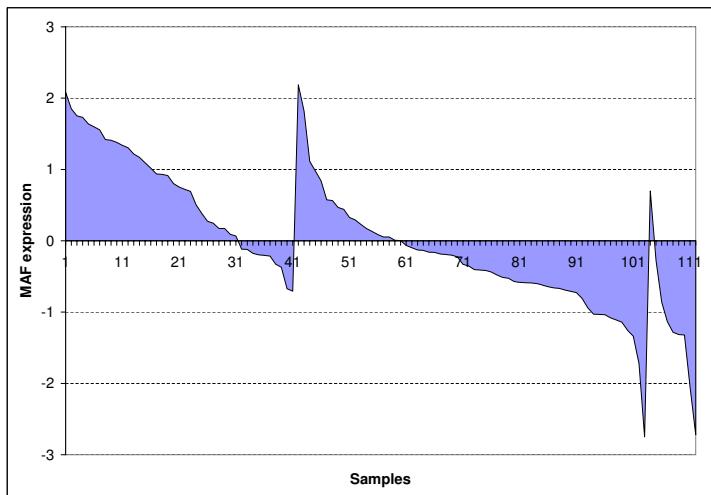


Fig. 8. Expression of MAF (v-MAF musculoaponeurotic fibrosarcoma oncogene homology) on a set of prostate samples from Lapointe *et al* [92]. Each integer on the x-axis corresponds to a sample index, and they have been ordered to show the particular profile of MAF. The first 41 samples (starting from the left) corresponds to normal prostate samples, MAF expresses above the mean value in most of them. The expression of MAF is then reduced in the prostate tumor samples (the next 52 samples), and MAF is highly downregulated in the lymph node metastasis samples (the rightmost nine samples).

at identifying the effects of gene expression in the bovine prostate after two different androgens (boldenone or testosterone) are administered. Toffolatti *et al.* conclude: “These results suggest that testosterone could be a stronger activator of MAF expression than boldenone and could contribute to identify different treatment classes.” A natural research question naturally arises: Can MAF be an early marker of the transition of prostate cancer samples to an androgen independent phenotype? [91].

It is perhaps prudent to see if there exist a separation of primary prostate tumors in terms of MAF expression pattern, and how does it compare with normal samples in that dataset. In Lapointe *et al.* [92] we have another highly-cited contribution which has an associated dataset that has proved valuable in the quest of finding good biomarkers for prostate cancer. Fig. 1 of [92] of that paper presents a hierarchical cluster analysis of the profiled gene expression in 62 primary prostate tumors, 41 normal prostate specimens and nine lymph node metastases. The authors have selected 5,153 variably expressed probes corresponding to genes (rows). The gene expression ratios were mean-centered and are depicted by a log2 pseudocolor scale. Two probes correspond to MAF (N34436, AA043501). In Figure 8, we present the average of the expression of MAF (averaged over the two probes). We have arranged the samples such that

the first 41 entries correspond to the normal prostate samples and, within this group, we have arranged them in decreasing order of averaged MAF expression. It is clear from the figure that most of the MAF samples have relatively higher values (in comparison with the mean). The next 62 samples (from left to right along the x-axis coordinate), correspond to the 62 primary prostate tumor samples. Again, we have ordered them according to decreasing average MAF expression. It is now clear that there seems to be two types of primary tumors. This is interesting as now we know that MAF is an androgen responsive gene and perhaps these two types can be stratified by MAF's expression. The final nine samples (the nine rightmost samples in the figure) correspond to the lymph node metastases. With the exception of one, all lymph node metastases have a low values of MAF.

The results of this chapter show that our methodology seems to be an important contribution to identify biomarkers. Due to its generality, it is certainly a data mining method that can be applied to other problem domains. However, it also shows that more research questions have arisen from this study. For instance, how to find a principled approach to identify the best values of α and β , and how they relate to a given type of classifier. In addition, we pose the research quest of how to incorporate other forms of information available in databases (like protein-protein interaction) to develop biomarker signatures that relate with known biochemical processes. This is an exciting new area, on “integrative bioinformatics”, and its development will finally enable us to identify the “oncosystems” which lay beneath the differential patterns of gene expression.

References

1. Saeys, Y., Inza, I., Larrañaga, P.: A review of feature selection techniques in bioinformatics. *Bioinformatics* 23(19), 2507–2517 (2007)
2. B. D. W. Group: Biomarkers and surrogate endpoints: preferred definitions and conceptual framework. *Clin. Pharmacol. Ther.* 69(3), 89–95 (2001)
3. The prostate cancer foundation of australia (08/08/2008 2007)
4. Cotta, C., Sloper, C., Moscato, P.: Evolutionary search of thresholds for robust feature set selection: application to the analysis of microarray data. In: Raidl, G.R., Cagnoni, S., Branke, J., Corne, D.W., Drechsler, R., Jin, Y., Johnson, C.G., Machado, P., Marchiori, E., Rothlauf, F., Smith, G.D., Squillero, G. (eds.) *EvoWorkshops 2004*. LNCS, vol. 3005, pp. 21–30. Springer, Heidelberg (2004)
5. Cotta, C., Moscato, P.: The k -Feature Set problem is $W[2]$ -complete. *Journal of Computer and System Sciences* 67(4), 686–690 (2003)
6. Davies, S., Russell, S.: NP-completeness of searches for smallest possible feature sets. In: *Proceedings of the AAAI Symposium on Relevance*, pp. 41–43 (1994)
7. Downey, R., Fellows, M.: Parameterized Complexity. Monographs in Computer Science. Springer, Heidelberg (1999)
8. Berretta, R., Mendes, A., Moscato, P.: Selection of discriminative genes in microarray experiments using mathematical programming. *Journal of Research and Practice in Information Technology* 39(4), 231–243 (2007)

9. Moscato, P., Mathieson, L., Mendes, A., Berretta, R.: The electronic primaries: Predicting the u.s. presidency using feature selection with safe data reduction. In: Estivill-Castro, V. (ed.) Twenty-Eighth Australasian Computer Science Conference (ACSC 2005). CRPIT, vol. 38, pp. 371–380. ACS, Newcastle (2005)
10. Berretta, R., Mendes, A., Moscato, P.: Integer programming models and algorithms for molecular classification of cancer from microarray data. In: Estivill-Castro, V. (ed.) Twenty-Eighth Australasian Computer Science Conference (ACSC 2005). CRPIT, vol. 38, pp. 361–370. ACS, Newcastle (2005)
11. Moscato, P., Berretta, R., Hourani, M., Mendes, A., Cotta, C.: Genes related with Alzheimer's disease: A comparison of evolutionary search, statistical and integer programming approaches. In: Rothlauf, F., Branke, J., Cagnoni, S., Corne, D.W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G.D., Squillero, G. (eds.) EvoWorkshops 2005. LNCS, vol. 3449, pp. 84–94. Springer, Heidelberg (2005)
12. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: IJCAI, pp. 1022–1029 (1993)
13. Berretta, R., Costa, W., Moscato, P.: Combinatorial optimization models for finding genetic signatures from gene expression datasets. In: Keith, J.M. (ed.) Bioinformatics, Volume II: Structure, Function and Applications, Methods in Molecura Biology, ch. 19, pp. 363–378. Humana Press (2008)
14. Hourani, M., Mendes, A., Berretta, R., Moscato, P.: Genetic biomarkers for brain hemisphere differentiation in parkinson's disease. In: AIP Conference Proceedings, vol. 952(1), pp. 207–216 (2007)
15. Hourani, M., Berretta, R., Mendes, A., Moscato, P.: Genetic signatures for a rodent model of parkinson's disease using combinatorial optimization methods. In: Keith, J.M. (ed.) Bioinformatics, Volume II: Structure, Function and Applications. Structure, Function and Applications, Methods in Molecura Biology, vol. II, pp. 379–392. Humana Press (2008), doi:10.1007/978-1-60327-429-6_20
16. Ravetti, M.G., Moscato, P.: Identification of a 5-protein biomarker molecular signature for predicting alzheimer's disease, PLOS One (accepted)
17. Ross, D., Scherf, U., Eisen, M., et al.: Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics* 24(3), 227–235 (2000)
18. Brown, V., Ossadtchi, A., Khan, A., Cherry, S., Leahy, R., Smith, D.: High-throughput imaging of brain gene expression. *Genome Research* 12(2), 244–254 (2002)
19. Singh, D., Febbo, P.G., Ross, K., Jackson, D.G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A.A., D'Amico, A.V., Richie, J.P., Lander, E.S., Loda, M., Kantoff, P.W., Golub, T.R., Sellers, W.R.: Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell* 1(2), 203–209 (2002)
20. Orsenigo, C.: Gene selection and cancer microarray data classification via mixed-integer optimization. In: Marchiori, E., Moore, J.H. (eds.) EvoBIO 2008. LNCS, vol. 4973, pp. 141–152. Springer, Heidelberg (2008)
21. Shah, S., Kusiak, A.: Cancer gene search with data-mining and genetic algorithms. *Computers in Biology and Medicine* 37(2), 251–261 (2007)
22. Wang, H.-Q., Wong, H.-S., Huang, D.-S., Shu, J.: Extracting gene regulation information for cancer classification. *Pattern Recognition* 40(12), 3379–3392 (2007)

23. Chandran, U.R., Ma, C., Dhir, R., Bisceglia, M., Lyons-Weiler, M., Liangand, W., Michalopoulos, G., Becich, M., Monzon, F.A.: Gene expression profiles of prostate cancer reveal involvement of multiple molecular pathways in the metastatic process. *BMC Cancer* 7(64)
24. Yu, Y.P., Landsittel, D., Jing, L., Nelson, J., Ren, B., Liu, L., McDonald, C., Thomas, R., Dhir, R., Finkelstein, S., Michalopoulos, G., Becich, M., Luo, J.-H.: Gene expression alterations in prostate cancer predicting tumor aggression and preceding development of malignancy. *J. Clin. Oncol.* 22(14), 2790–2799 (2004)
25. Tibshirani, R., Hastie, T., Narasimhan, B., Chu, G.: Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences of the United States of America* 99(10), 6567–6572 (2002)
26. Tusher, V.G., Tibshirani, R., Chu, G.: Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America* 98(9), 5116–5121 (2001)
27. Gomez Ravetti, M., Moscato, P.: Identification of a 5-protein biomarker molecular signature for predicting alzheimer's disease. *PLOS One* 3(9), e3111 (2008)
28. Tibshirani, R., Hastie, T., Narasimhan, B., Chu, G.: Class prediction by nearest shrunken centroids, with applications to dna microarrays. *Statistical Science* 18(1), 104–117 (2003)
29. Banerjee, A.G., Bhattacharyya, I., Vishwanatha, J.K.: Identification of genes and molecular pathways involved in the progression of premalignant oral epithelia. *Mol. Cancer Ther.* 4(6), 865–875 (2005)
30. Eichele, K., Ramer, R., Hinz, B.: Decisive role of cyclooxygenase-2 and lipocalin-type prostaglandin d synthase in chemotherapeutics-induced apoptosis of human cervical carcinoma cells. *Oncogene* 27(21), 3032–3044 (2008)
31. Su, B., Guan, M., Xia, J., Lu, Y.: Stimulation of lipocalin-type prostaglandin d synthase by retinoic acid coincides with inhibition of cell proliferation in human 3ao ovarian cancer cells. *Cell Biol. Int.* 27(7), 587–592 (2003)
32. Kim, J., Yang, P., Suraokar, M., Sabichi, A., Llansa, N., Mendoza, G., Subbarayan, V., Logothetis, C., Newman, R., Lippman, S., Menter, D.: Suppression of prostate tumor cell growth by stromal cell prostaglandin d synthase-derived products. *Cancer Res.* 65(14), 6189–6198 (2005)
33. Park, J.M., Kanaoka, Y., Eguchi, N., Aritake, K., Grujic, S., Materi, A.M., Buslon, V.S., Tippin, B.L., Kwong, A.M., Salido, E., French, S.W., Urade, Y., Lin, H.J.: Hematopoietic prostaglandin d synthase suppresses intestinal adenomas in apcmin/+ mice. *Cancer Res.* 67(3), 881–889 (2007)
34. Richard, C.L., Lowthers, E.L., Blay, J.: 15-deoxy-delta(12,14)-prostaglandin J(2) down-regulates CXCR4 on carcinoma cells through PPARgamma- and NFkappaB-mediated pathways. *Exp. Cell Res.* 313(16), 3446–3458 (2007)
35. Chen, Y., Perussia, B., Campbell, K.: Prostaglandin d2 suppresses human nk cell function via signaling through d prostanoid receptor. *J. Immunol.* 179(5), 2766–2773 (2007)
36. Cao, H., Xiao, L., Park, G., Wang, X., Azim, A.C., Christman, J.W., van Breemen, R.B.: An improved lc-ms/ms method for the quantification of prostaglandins e(2) and d(2) production in biological fluids. *Anal. Biochem.* 372(1), 41–51 (2008)

37. Torres, D., Paget, C., Fontaine, J., Mallevaey, T., Matsuoka, T., Narumiya, T.M.S., Capron, M., Gosset, P., Faveeuw, C., Trottein, F.: Prostaglandin d2 inhibits the production of ifn-gamma by invariant nk t cells: consequences in the control of b16 melanoma. *J. Immunol.* 180(2), 783–792 (2008)
38. Watson, M., Lind, M., Smith, L., Drew, P., Cawkwell, L.: Expression microarray analysis reveals genes associated with in vitro resistance to cisplatin in a cell line model. *Acta Oncol.* 46(5), 651–658 (2007)
39. Guy, C.A., Hoogendoorn, B., Smith, S.K., Coleman, S., O'Donovan, M.C., Buckland, P.R.: Promoter polymorphisms in glutathione-s-transferase genes affect transcription. *Pharmacogenetics* 14(1), 45–51 (2004)
40. Denson, J., Xi, Z., Wu, Y., Yang, W., Neale, G., Zhang, J.: Screening for inter-individual splicing differences in human gstm4 and the discovery of a single nucleotide substitution related to the tandem skipping of two exons. *Gene.* 379, 14855 (2006)
41. Efferth, T., Volm, M.: Glutathione-related enzymes contribute to resistance of tumor cells and low toxicity in normal organs to artesunate. *Vivo* 19(1), 225–232 (2005)
42. Knight, T., Choudhuri, S., Klaassen, C.: Constitutive mrna expression of various glutathione s-transferase isoforms in different tissues of mice. *Toxicol Sci.* 100(2), 513–524 (2007)
43. Liloglou, T., Walters, M., Maloney, P., Youngson, J., Field, J.K.: A t2517c polymorphism in the gstm4 gene is associated with risk of developing lung cancer. *Lung Cancer* 7(2), 143–146 (2002)
44. DiLella, A.G., Toner, T.J., Austin, C.P., Connolly, B.M.: Identification of genes differentially expressed in benign prostatic hyperplasia. *J. Histochem Cytochem.* 49(5), 669–670 (2001)
45. Luo, J., Dunn, T.A., Ewing, C.M., Walsh, P.C., Isaacs, W.B.: Decreased gene expression of steroid 5 alpha-reductase 2 in human prostate cancer: implications for finasteride therapy of prostate carcinoma. *Prostate* 57(2), 134–139 (2003)
46. Grigo, K., Wirsing, A., Lucas, B., Klein-Hitpass, L., Ryffel, G.U.: Hnf4 alpha orchestrates a set of 14 genes to down-regulate cell proliferation in kidney cells. *Biol. Chem.* 389(2), 179–187 (2008)
47. Wu, Q., Parry, G.: Hepsin and prostate cancer. *Front Biosci.* 12, 5052–5059 (2007)
48. Matsuo, T., Nakamura, K., Takamoto, N., Kodama, J., Hongo, A., Abrzua, F., Nasu, Y., Kumon, H., Hiramatsu, Y.: Expression of the serine protease hepsin and clinical outcome of human endometrial cancer. *Anticancer Res.* 28(1A), 159–164 (2008)
49. Kelly, K.A., Setlur, S.R., Ross, R., Anbazhagan, R., Waterman, P., Rubin, M.A., Weissleder, R.: Detection of early prostate cancer using a hepsin-targeted imaging agent. *Cancer Res.* 68(7), 2286–2291 (2008)
50. Magee, J.A., Araki, T., Patil, S., Ehrig, T., True, L., Humphrey, P.A., Cattonalona, W.J., Watson, M.A., Milbrandt, J.: Expression profiling reveals hepsin overexpression in prostate cancer. *Cancer Res.* 61(15), 5692–5696 (2001)
51. Huppi, K., Chandramouli, G.V.: Molecular profiling of prostate cancer. *Curr. Urol. Rep.* 5(1), 45–51 (2004)
52. Xu, L., Tan, A.C., Naiman, D.Q., Geman, D., Winslow, R.L.: Robust prostate cancer marker genes emerge from direct integration of inter-study microarray data. *Bioinformatics* 21(20), 3905–3911 (2005)

53. Riddick, A.C., Barker, C., Sheriffs, I., Bass, R., Ellis, V., Sethia, K.K., Edwards, D.R., Ball, R.Y.: Banking of fresh-frozen prostate tissue: methods, validation and use. *BJU Int.* 91(4), 315–324 (2003)
54. Stephan, C., Yousef, G.M., Scorilas, A., Jung, K., Jung, M., Kristiansen, G., Hauptmann, S., Kishi, T., Nakamura, T., Loening, S.A., Diamandis, E.P.: Hepsin is highly over expressed in and a new candidate for a prognostic indicator in prostate cancer. *J. Urol.* 171(1), 187–191 (2004)
55. Fromont, G., Chene, L., Vidaud, M., Vallancien, G., Mangin, P., Fournier, G., Validire, P., Latil, A., Cussenot, O.: Differential expression of 37 selected genes in hormone-refractory prostate cancer using quantitative taqman real-time rt-pcr. *Int. J. Cancer.* 114(2), 174–181 (2005)
56. Pal, P., Xi, H., Kaushal, R., Sun, G., Jin, C.H., Jin, L., Suarez, B.K., Catalona, W.J., Deka, R.: Variants in the HEPSIN gene are associated with prostate cancer in men of european origin. *Hum. Genet.* 120(2), 187–192 (2006)
57. Burmester, J.K., Suarez, B.K., Lin, J.H., Jin, C.H., Miller, R.D., Zhang, K.Q., Salzman, S.A., Reding, D.J., Catalona, W.J.: Analysis of candidate genes for prostate cancer. *Hum Hered.* 57(4), 172–178 (2004)
58. Heinrich, R., Ben-Izhak, E.L.O., Aronheim, A.: The c-Jun dimerization protein 2 inhibits cell transformation and acts as a tumor suppressor gene. *J. Biol. Chem.* 279(7), 5708–5715 (2004)
59. Mehraein-Ghom, F., Lee, E., Church, D.R., Thompson, T.A., Basu, H.S., Wilding, G.: Jund mediates androgen-induced oxidative stress in androgen dependent Incap human prostate cancer cells. *Prostate* 68(9), 924–934 (2008)
60. Polytarchou, C., Hatziapostolou, M., Papadimitriou, E.: Hydrogen peroxide stimulates proliferation and migration of human prostate cancer cells through activation of activator protein-1 and up-regulation of the heparin affin regulatory peptide gene. *J. Biol. Chem.* 280(49), 40428–40435 (2005)
61. Zhang, J.S., Gong, A., Cheville, J.C., Smith, D.I., Young, C.Y.: Agr2, an androgen-inducible secretory protein overexpressed in prostate cancer. *Genes Chromosomes Cancer.* 43(3), 249–259 (2005)
62. Zhang, Y., Forootan, S.S., Liu, D., Barraclough, R., Foster, C.S., Rudland, P.S., Ke, Y.: Increased expression of anterior gradient-2 is significantly associated with poor survival of prostate cancer patients. *Prostate Cancer Prostatic Dis.* 10(3), 293–300 (2007)
63. LI, L.I.K., Shishkin, S.S., Khasigov, P.Z., Dzeranov, N.K., Kazachenko, A.V., Toropygin, I., Mamikina, S.V.: Identification of agr2 protein, a novel potential cancer marker, using proteomics technologies, [article in russian]. *Prikl Biokhim Mikrobiol.* 42(4), 480–484 (2006)
64. Wang, Z., Hao, Y., Lowe, A.W.: The adenocarcinoma-associated antigen, agr2, promotes tumor growth, cell migration, and cellular transformation. *Cancer Res.* 68(2), 492–497 (2008)
65. Kristiansen, G., Pilarsky, C., Wissmann, C., Kaiser, S., Bruemmendorf, T., Roepcke, S., Dahl, E., Hinzmann, B., Specht, T., Pervan, J., Stephan, C., Loening, S., Dietel, M., Rosenthal, A.: Expression profiling of microdissected matched prostate cancer samples reveals CD166/MEMD and CD24 as new prognostic markers for patient survival. *J. Pathol.* 205(3), 359–376 (2005)
66. Landers, K.A., Samaratunga, H., Teng, L., Buck, M., Burger, M.J., Scells, B., Lavin, M.F., Gardiner, R.A.: Identification of claudin-4 as a marker highly over-expressed in both primary and metastatic prostate cancer. *Br. J. Cancer* 99(3), 491–501 (2008)

67. Kim, S.O., Lee, I.J., Choi, Y.H.: Genistein reduced the invasive activity of human breast carcinoma cells as a result of decreased tight junction permeability and modulation of tight junction proteins. *Cancer Lett.* (Epub ahead of print)
68. Hough, C.D., Sherman-Baust, C.A., Pizer, E.S., Montz, F.J., Im, D.D., Rosen-shein, N.B., Cho, K.R., Riggins, G.J., Morin, P.J.: Large-scale serial analysis of gene expression reveals genes differentially expressed in ovarian cancer. *Cancer Res.* 60(22), 6281–6287 (2000)
69. Kleinberg, L., Holth, A., Trope, C.G., Reich, R., Davidson, B.: Claudin upregulation in ovarian carcinoma effusions is associated with poor survival. *Hum Pathol.* 39(5), 747–757 (2008)
70. Long, H., Crean, C.D., Lee, W.H., Cummings, O.W., Gabig, T.G.: Expression of clostridium perfringens enterotoxin receptors claudin-3 and claudin-4 in prostate cancer epithelium. *Cancer Res.* 61(21), 7878–7881 (2001)
71. Morin, P.J.: Claudin proteins in human cancer: promising new targets for diagnosis and therapy. *Cancer Res.* 65(21), 9603–9606 (2005)
72. Hewitt, K.J., Agarwal, R., Morin, P.J.: The claudin gene family: expression in normal and neoplastic tissues. *BMC Cancer* 6, 186 (2006)
73. Nichols, L.S., Ashfaq, R., Iacobuzio-Donahue, C.A.: Claudin 4 protein expression in primary and metastatic pancreatic cancer: support for use as a therapeutic target. *Am J. Clin. Pathol.* 121(2), 226–230 (2004)
74. Foss, C.A., Fox, J.J., Feldmann, G., Maitra, A., Iacobuzio-Donohue, C., Kern, S.E., Hruban, R., Pomper, M.G.: Radiolabeled anti-claudin 4 and anti-prostate stem cell antigen: initial imaging in experimental models of pancreatic cancer. *Mol. Imaging.* 6(2), 131–139 (2007)
75. Hanada, S., Maeshima, A., Matsuno, Y., Ohta, T., Ohki, M., Yoshida, T., Hayashi, Y., Yoshizawa, Y., Hirohashi, S., Sakamoto, M.: Expression profile of early lung adenocarcinoma: identification of mrp3 as a molecular marker for early progression. *J. Pathol.* 216(1), 75–82 (2008)
76. Nishino, R., Honda, M., Yamashita, T., Takatori, H., Minato, H., Zen, Y., Sasaki, M., Takamura, H., Horimoto, K., Ohta, T., Nakanuma, Y., Kaneko, S.: Identification of novel candidate tumour marker genes for intrahepatic cholangiocarcinoma. *J. Hepatol.* 49(2), 207–216 (2008)
77. Bello, I.O., Vilen, S.T., Niinimaa, A., Kantola, S., Soini, Y., Salo, T.: Expression of claudins 1, 4, 5, and 7 and occludin, and relationship with prognosis in squamous cell carcinoma of the tongue. *Hum. Pathol.* 39(8), 1212–1220 (2008)
78. Ashton-Chess, J., Giral, M., Mengel, M., Renaudin, K., Foucher, Y., Gwinner, W., Braud, C., Dugast, E., Quillard, T., Thebault, P., Chiffolleau, E., Braudeau, C., Charreau, B., Soulillou, J., Brouard, S.: Tribbles-1 as a novel biomarker of chronic antibody-mediated rejection. *J. Am. Soc. Nephrol.* 19(6), 1116–1127 (2008), <http://jasn.asnjournals.org/cgi/content/abstract/19/6/1116>
79. Röthlisberger, B., Heizmann, M., Bargetzi, M.J., Huber, A.R.: Trib1 overexpression in acute myeloid leukemia. *Cancer Genet Cytogenet.* 176(1), 58–60 (2007)
80. Rücker, F.G., Bullinger, L., Schwaenen, C., Lipka, D.B., Wessendorf, S., Fröhling, S., Bentz, M., Miller, S., Scholl, C., Schlenk, R.F., Radlwimmer, B., Kestler, H.A., Pollack, J.R., Lichter, P., Döhner, K., Döhner, H.: Disclosure of candidate genes in acute myeloid leukemia with complex karyotypes using microarray-based molecular characterization. *J. Clin. Oncol.* 25(9), 1151–1152 (2007)

81. Keeshan, K., Shestova, O., Ussin, L., Pear, W.S.: Tribbles homolog 2 (trib2) and hoxa9 cooperate to accelerate acute myelogenous leukemia. *Blood Cells Mol. Dis.* 40(1), 119–121 (2008)
82. Puskas, L.G., Juhasz, F., Zarva, A., Hackler Jr., L., Farid, N.R.: Gene profiling identifies genes specific for well-differentiated epithelial thyroid tumors. *Cell Mol. Biol.* 51(2), 177–186 (2005)
83. Puiffe, M.L., Page, C.L., Filali-Mouhim, A., Zietarska, M., Ouellet, V., Tonin, P.N., Chevrette, M., Provencher, D.M., Mes-Masson, A.M.: Characterization of ovarian cancer ascites on cell invasion, proliferation, spheroid formation, and gene expression in an *in vitro* model of epithelial ovarian cancer. *Neoplasia* 9(10), 820–829 (2007)
84. Manss, A.H., Morris, B.J.: ZRANB2: Structural and functional insights into a novel splicing protein. *Int. J. Biochem. Cell Biol.* 40(11), 2353–2357 (2008)
85. Leiblich, A., Cross, S.S., Catto, J.W., Phillips, J.T., Leung, H.Y., Hamdy, F.C., Rehman, I.: Lactate dehydrogenase-b is silenced by promoter hypermethylation in human prostate cancer. *Oncogene* 25(20), 2953–2960 (2006)
86. Glen, A., Gan, C.S., Hamdy, F.C., Eaton, C.L., Cross, S.S., Catto, J.W., Wright, P.C., Rehman, I.: Itraq-facilitated proteomic analysis of human prostate cancer cells identifies proteins associated with progression. *J. Proteome Res.* 7(3), 897–907 (2008)
87. Wingender, E.: The TRANSFAC project as an example of framework technology that supports the analysis of genomic regulation. *Brief Bioinform.* 9(4), 326–332 (2008)
88. Kataoka, K., Noda, M., Nishizawa, M.: Maf nuclear oncprotein recognizes sequences related to an ap-1 site and forms heterodimers with both fos and jun. *Mol. Cell. Biol.* 14(1), 700–712 (1994)
89. Hofer, M., Fecko, A., Shen, R., Setlur, S., Pienta, K., Tomlins, S., Chinnaiyan, A., Rubin, M.: Expression of the platelet-derived growth factor receptor in prostate cancer and treatment implications with tyrosine kinase inhibitors. *Neoplasia* 6(5), 503–512 (2004)
90. Toffolatti, L., Gastaldo, L.R., Patarnello, T., Romualdi, C., Merlanti, R., Montesissa, C., Poppi, L., Castagnaro, M., Bargelloni, L.: Expression analysis of androgen-responsive genes in the prostate of veal calves treated with anabolic hormones. *Domest. Anim. Endocrinol.* 30(1), 38–55 (2006)
91. So, A., Gleave, M., Hurtado-Col, A., Nelson, C.: Mechanisms of the development of androgen independence in prostate cancer. *World J. Urol.* 23(1), 1–9 (2005)
92. Lapointe, J., Li, C., Higgins, J., van de Rijn, M., Bair, E., Montgomery, K., Ferrari, M., Egevad, L., Rayford, W., Bergerheim, U., Ekman, P., DeMarzo, A., Tibshirani, R., Botstein, D., Brown, P., Brooks, J., Pollack, J.: Gene expression profiling identifies clinically relevant subtypes of prostate cancer. *Proc. Natl. Acad. Sci. USA* 101(3), 811–816 (2004)

A Tutorial on Multi-label Classification Techniques

André C.P.L.F. de Carvalho¹ and Alex A. Freitas²

¹ Department of Computer Science, University of São Paulo, São Carlos, SP, Brazil
andre@icmc.usp.br
<http://www.icmc.usp.br/~andre>

² Computing Laboratory, University of Kent, Canterbury, CT2 7NF, UK
A.A.Freitas@kent.ac.uk
<http://www.cs.kent.ac.uk/~aaf>

Summary. Most classification problems associate a single class to each example or instance. However, there are many classification tasks where each instance can be associated with one or more classes. This group of problems represents an area known as multi-label classification. One typical example of multi-label classification problems is the classification of documents, where each document can be assigned to more than one class. This tutorial presents the most frequently used techniques to deal with these problems in a pedagogical manner, with examples illustrating the main techniques and proposing a taxonomy of multi-label techniques that highlights the similarities and differences between these techniques.

1 Introduction

Machine Learning (ML) is a sub-area of Artificial Intelligence (AI) concerned with the induction of a model through a learning process. A particular area of ML, named Inductive Learning, consists of techniques that induce these models by using a set of previously known instances or examples, called training instances. After the model has been induced, it can then be applied to new, previously unseen, data.

ML models have been applied to a wide range of tasks. These tasks can be broadly divided into five main classes: Association, Classification, Regression, Clustering and Optimization tasks. This paper is concerned with classification tasks, which can be formally defined as:

Given a set of training examples composed of pairs $\{x_i, y_i\}$, find a function $f(x)$ that maps each attribute vector x_i to its associated class y_i , $i = 1, 2, \dots, n$, where n is the total number of training examples.

Once it has been trained, the classification model can have its predictive accuracy estimated by applying it to a set of new, previously unknown, examples. Its accuracy measure for these new instances estimates the generalization ability (predictive accuracy) of the classification model induced.

Classification problems can be categorized according to the number of class labels that can be assigned to a particular input instance. The most common

approach is to have mutually exclusive classes. For example, suppose a document classification problem where each document should be classified according to the language it was written. If a document could be written in just one idiom and the possible idioms were Chinese, English, French, German, Portuguese and Spanish, each document would be classified in one and only one of these six classes. In this case, each input instance is assigned to only one of the possible classes. This is known as single-label classification. Most of the classification problems investigated in ML are single-label classification problems.

However, there is a large number of relevant problems where each instance can be simultaneously associated with more than one class. These problems, where the classes are not disjoint, are known as multi-label classification problems.

The majority of the works on multi-label classification started as an attempt to deal with ambiguities found in document classification problems [51]. In a document categorization problem, each document may simultaneously belong to more than one topic or label. For example, a document can be classified as belonging to Computer Science, Physics and Application, another document can be assigned to the areas of Biology and Theory and a third can be a Mathematics document related to an Application in Physics. This problem would then have at least six classes or labels (Computer Science, Physics, Application, Biology, Theory and Mathematics). Even now, text classification is the main application area of multi-label classification techniques [21][24][26][28][29][30][36][42][48][50]. However, relevant works can also be found in areas like bioinformatics [11] [51], [14], medical diagnosis [25], scene classification [4][37] and map labeling [53].

Different approaches have been proposed in the literature for dealing with multi-label problems. One of them combines single-label classifiers to deal with the multi-label classification task. A second approach modifies single-label classifiers, by the adaptation of their internal mechanisms, to allow their use in multi-label problems. A third group proposes new algorithms specifically designed to deal with multi-label problems.

This text is organized as follows. In the next section, the main methods found in the literature for dealing with multi-label classification are organized and described. First, the authors define the structure and main characteristics of multi-label problems, without worrying about the learning algorithms used. Later, the authors discuss some algorithm specific approaches. Section 3 discusses how new instances can be classified in a multi-label environment. Section 4 has the final considerations and main conclusions of this work.

2 Categorizing Multi-label Classification Problems

In a trained classifier, a probability can be associated with each one of the existent classes and then be used for the classification of a new example. Thus, if the problem has N classes, a probability p_i , $1 \leq i \leq N$, where $0 \leq p_i \leq 1$, is assigned to each class. If the system is trained for single-label classification, there is a restriction that $\sum p_i = 1$. For a multi-label problem, this restriction is not adopted.

According to [13], binary classification, multi-class classification and ordinal regression problems can be seen as special cases of multi-label problems where the number of labels assigned to each instance is equal to 1.

Some of the solutions to multi-label problems are restricted to binary classification. However, the largest number of methods found in the literature are used for are multi-class problems. For multi-class problems, the main focus of this text, the original multi-label problem is converted to one or more single-label problems.

In order to illustrate the different methods found in the literature for multi-label classification problems, these methods are organized in a hierarchical structure in Figure 1.

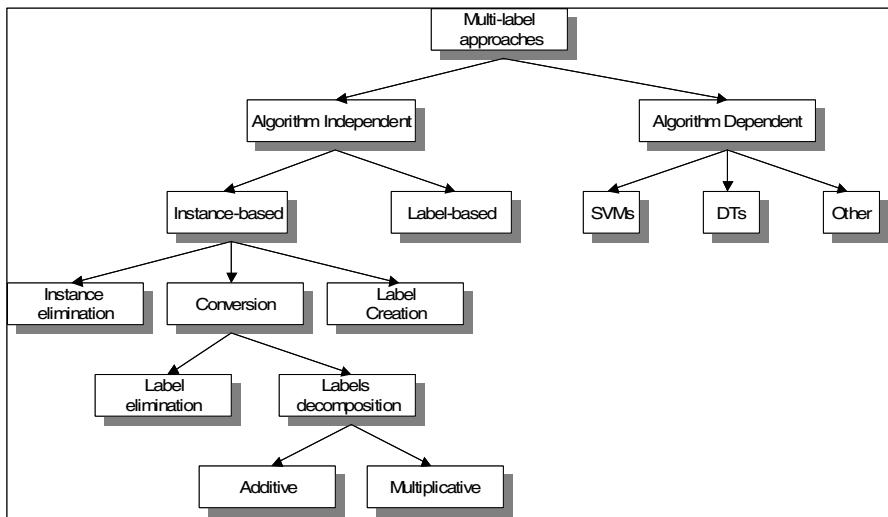


Fig. 1. Methods used in Multi-Label Classification Problems

According to Figure 1, the existing methods can be divided into two main approaches: algorithm independent and algorithm dependent. The next sections describe the main characteristics of the methods belonging to each approach.

2.1 Algorithm Independent Approach

The algorithm independent multi-label methods can be used with any learning algorithm. In this approach, a multi-label classification problem is usually dealt with by transforming the original problem into a set of single-label problems. This transformation can be based on either the class labels, named label-based, or the instances, named instance-based.

Label-based transformation

In the label-based transformation, N classifiers, where N is the number of classes, are used in the multi-label problem. Each classifier is associated with one of the

classes and trained to solve a binary classification problem, its class against the others. For this reason, this approach is also known as the binary approach or cross-training [4]. Any classifier can be used for binary classification. Many popular classifiers can deal only with binary classification problems.

As an example of the use of this approach, suppose a multi-label problem, illustrated by Figure 2, with 3 classes or labels. Since one classifier should be associated with each class, 3 classifiers would be trained. The multi-label problem with 3 classes is then divided into 3 binary problems, one for each class. The i^{th} classifier would be trained to classify examples from the i^{th} class as positive and examples from the remaining classes as negative. Therefore, each classifier would be specialized for a particular class. After the classifiers are trained, whenever a previously unknown example is presented, the classes whose classifier produced a positive label are assigned to it [1].

One of the first works in multi-label classification was due to [25]. In this work, the authors investigated the use of Decision Trees, DTs, in multi-label problems. They proposed a tree-based model, named MULTI- α , which divides the original multi-label problem into N single-label sub-problems, where N is the number of classes. Thus, for each class C_i , $1 \leq i \leq N$, it generates a decision tree using the classes C_i and $\neg C_i$. The outputs above a threshold value are assumed to be correct. The set of outputs produced by the individual classifiers provide the system's final decision. The method was evaluated in a medical diagnosis problem. It is possible to see that this method is similar to one of the algorithm-independent methods, the Label-based transformation.

Similarities can be found between the label-based transformation and the one-against-all approach employed for multi-class problems [23]. However, the one-against-all approach is employed to allow the solution of problems with more than two classes using binary classifiers. Another difference is that, in a multi-class problem, each example is assigned to only one class.

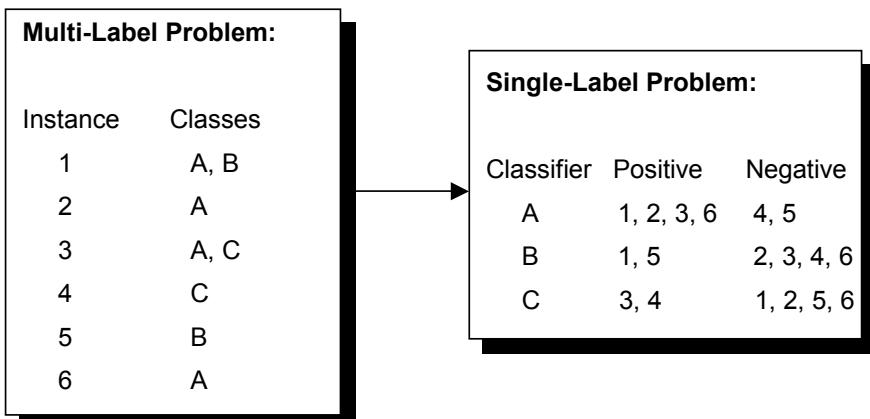


Fig. 2. Label-based transformation

This approach assumes that the labels of an instance are independent among themselves, which is not always true. By ignoring the possible correlation between labels, this approach may lead to poor generalization ability.

A label-based transformation is reversible, since it is possible to recover the original multi-label problem from the new single-label problem. It requires N classifiers, where N is the number of classes.

Instance-based transformation

In the transformation based on instances, named instance-based, the set of labels associated to each instance is redefined in order to convert the original multi-label problem into one or more single-label problems. In this redefinition, one or more classification problems can be produced. Different from label based transformations, which produce only binary classification problems, instance based transformations may produce both binary and multi-class classification problems.

Three different groups of strategies have been proposed in the literature for instance-based transformation:

- Elimination of multi-label instances;
- Creation of new single-labels using the existent multi-labels, here named conversion
- Conversion of multi-label instances into single-label instances:
 - Simplification;
 - Decomposition:
 - Additive;
 - Multiplicative.

Instance elimination is the simplest, but probably the least effective instance-based strategy. It does not solve the original multi-label problem. The elimination of those instances with more than one label will change the current problem into another, much simpler problem, possibly not as relevant as the previous one. An example of the use of this approach is shown in the Figure 3. According to this figure, the multi-label instances, 1 and 3, are eliminated in order to transform the original multi-label problem into a single-label problem. The negative aspect of this approach is that the instances eliminated can represent relevant information to characterize the problem domain.

For protein classification, for example, many proteins have more than one function. How would the user be able to predict the other functions? The elimination of these proteins from the data set would significantly reduce the significance of the model induced by the classifier. Since it is not possible to find out, in the new single-label problem, which instances were eliminated, this method is irreversible. It does not change the number of required classifiers.

There are other methods reported in the literature that, although classifier independent, aim to improve the performance by pre-processing the data set rather

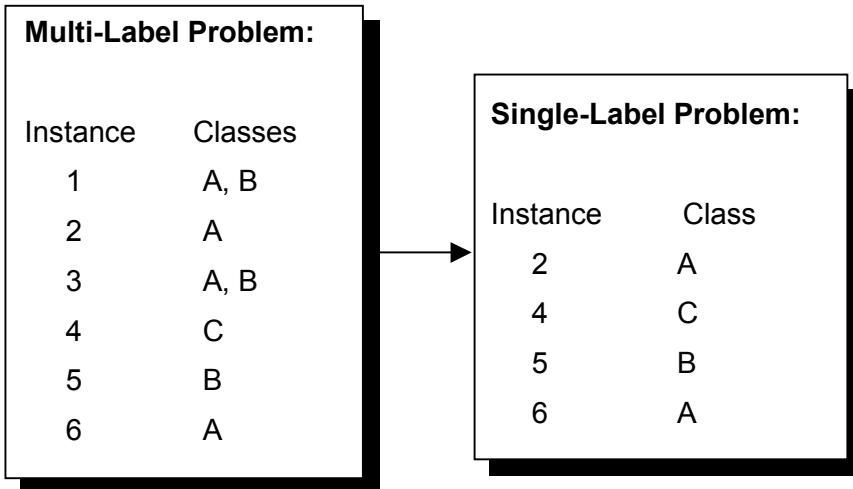


Fig. 3. Elimination of instances with more than one label

than naively eliminating all multi-label instances. In [20], the authors propose the removal of the instances close to the decision hyperplane and the elimination of the instances in the confusing classes. The confusing classes are defined using the confusion matrix.

When label creation is adopted, each possible combination of more than one class is converted to a new single class (label). The combination of the original classes can largely increase the number of classes and result in some classes with very few instances. This problem becomes increasingly worse as the number of possible labels for each instance increases. Figure 4 illustrates this approach.

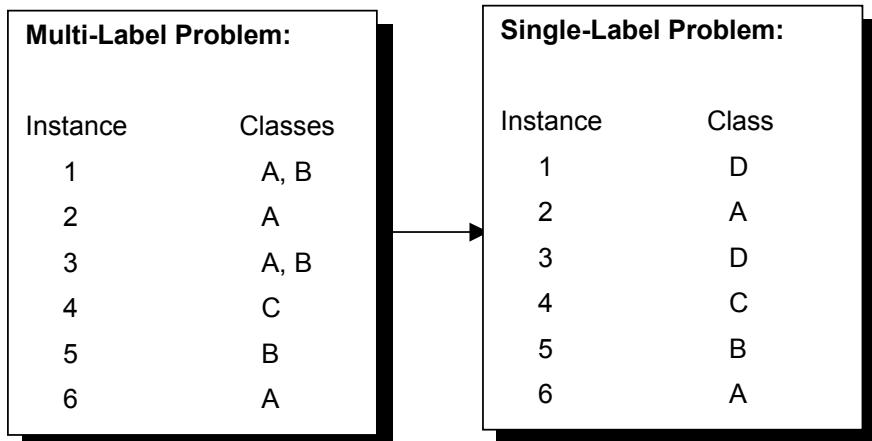


Fig. 4. Creation of new classes

It can be easily observed in this figure that the labels of the two multi-label instances, 1 and 3, which were A and B in both cases, were combined to create a new label, D.

The labels associated with each instance in the original multi-label problem are not lost in the creation of the new labels for the single-label problem. The number of classifiers is the same in both problems if a multi-class classifier is used. However, if a binary classifier is used, the number of classifiers required increases, by comparison with the original multi-label problem.

For the case of label conversion, there are two variations. The first variation transforms each multi-label instance into a single label instance. It is named label simplification. In the second variation, named label decomposition, each multi-label instance is decomposed into a set of single-label instances.

When transforming a multi-label instance into a single-label one, if the instance has more than one label, one of its labels is selected. The other labels are just eliminated. Two alternatives can be followed for the label selection. This procedure can either use a deterministic criterion, selecting from the labels associated with the instance the most likely to be true, or randomly select one of the labels. Figure 5 shows an example of this approach.

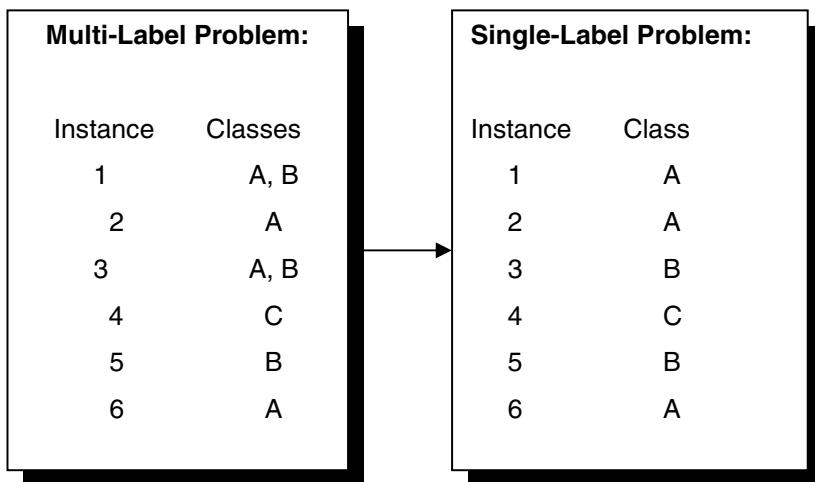


Fig. 5. Transform by label elimination of a multi-label problem into a set of single-label problems

It is easy to see in this example the simplification of the two multi-label instances, 1 and 3, by randomly selecting one of the labels, in both cases A and B, associated with each of them. As a result, the label A was randomly assigned to the instance 1 and the label B was randomly associated with the instance 2.

The selection of one of the labels will over-simplify the problem. Suppose that the classification problem involves the functional classification of a protein. A protein with more than one function would be classified as having just one of the functions, thus ignoring possibly relevant information.

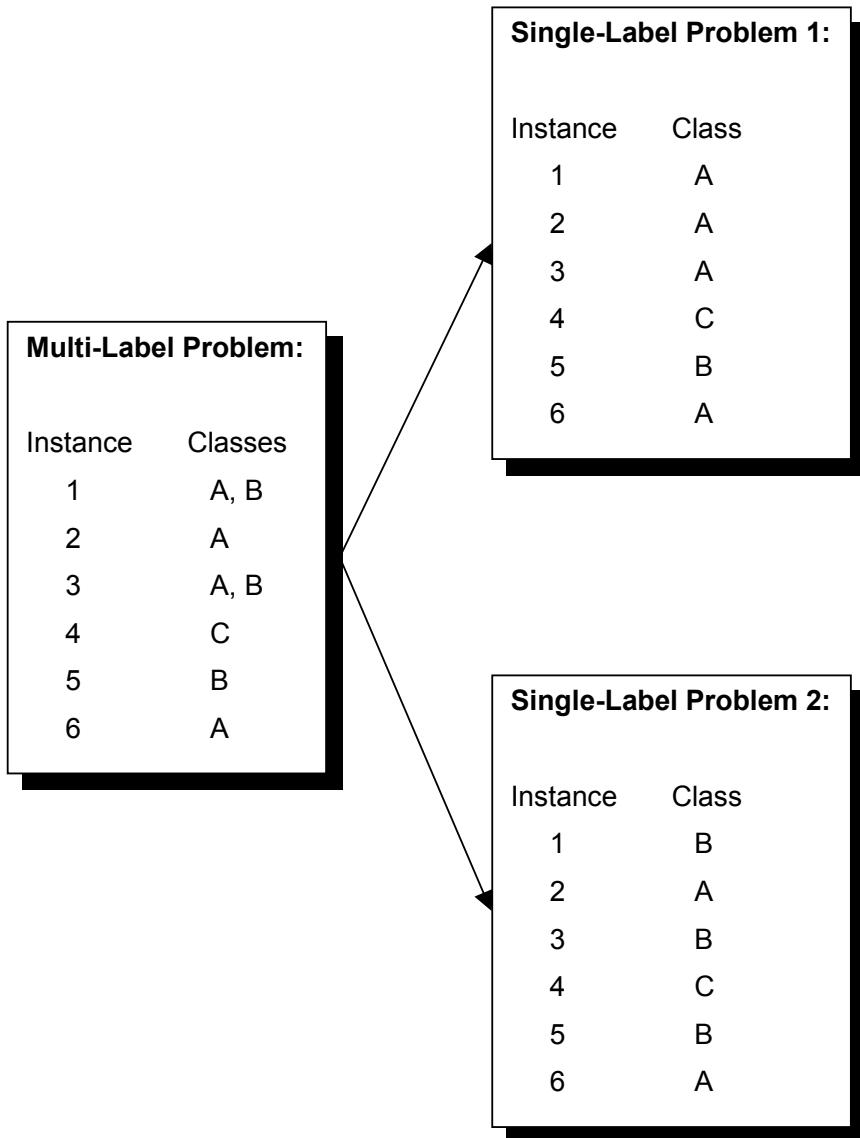


Fig. 6. Transform by decomposition of a multi-label problem into a set of single-label problems

If the deterministic criterion is adopted, it is possible to return to the original multi-label problem from the new single-label problem. If the random criterion is chosen, this return is not possible. The same number of classifiers is generally used in the multi-label and the single-label problems.

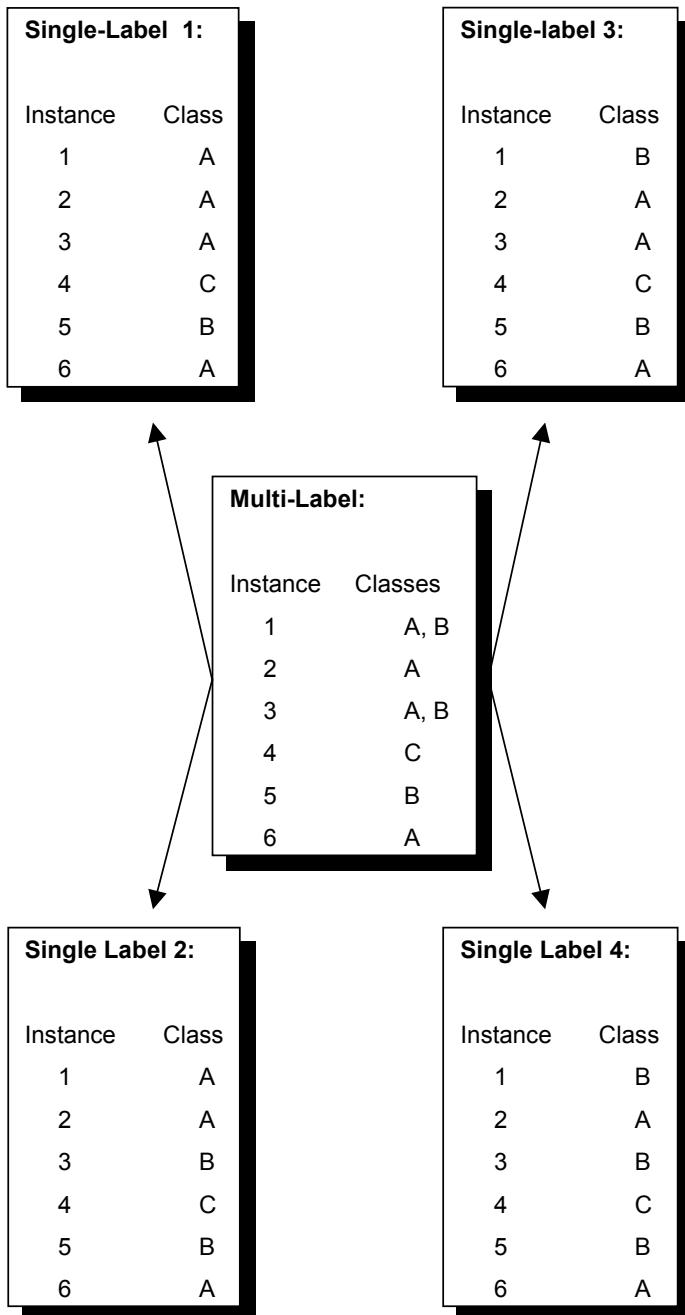


Fig. 7. Decomposition of a multi-label problem into a set of single-label problems according to the multiplicative method

In the decomposition approach, the original multi-label problem with N classes and M instances is divided into K sets of single-label problems. The value of K varies from 1, when no instance has more than one label, to $(N-1)^M$, if all the instances have $N-1$ labels. Two alternatives can be employed for this approach: the additive method and the multiplicative method.

In the additive method, for each instance, each of the possible labels is considered to be the positive class in sequence. Therefore, the number of classifiers is given by $\sum_i (l_i - 1)$, where l_i is the number of labels in the i^{th} instance. Thus, if the labels A, B and C appear in the multi-label instances, when the classifier for the class A is trained, all the multi-label instances that have the label A become single-label instances for the class A. The same happens for the other labels. This method was proposed in [37] and is named cross-training.

The number of classifiers, K , is equal to the number of labels that belong to at least one multi-label instance. This method allows the recovery of the original multi-label problem from the new single-label problems. Figure 6 illustrates this method. For this situation, the number of classifiers is given by $1 + 1 = 2$.

For classifiers based on density estimation, this method may favor the multi-label instances [37]. The authors argue that the multi-label instances are likely to be closer to the decision boundaries, making the use of SVMs very suitable. They also say that if the proportion of multi-label samples is too high, a sampling technique can be employed to use a subset of them for each classifier.

The second decomposition method, here named multiplicative, is similar to another approach employed to divide multi-class problems into a set of binary problems, the one-against-one approach [23]. In this case, a combination of all the possible single-label classifiers is used.

The number of classifiers is given by $\prod_i l_i$, which is the product of the number of labels for each instance. Figure 7 illustrates this approach. In this case, the number of classifiers would be equal to $2 \times 1 \times 2 \times 1 \times 1 = 4$. This method is clearly not scalable, since the number of classifiers grows exponentially with the number of labels in the instances. It is easy to see that the previous additive method produces a subset of the single-label problems generated by this method. This method is reversible, allowing the restitution of the original multi-label problem.

Although the multiplicative decomposition method minimizes the deficiencies of those previous approaches where labels were combined or eliminated, the former, like the label creation method, does not take into account the interactions/correlations that can exist between the labels of a particular instance.

As seen in this section, different methods have been proposed in the literature for the algorithm independent approach. Table 1 summarizes the main characteristics of these methods.

According to this table, where L represents the Number of labels and l_i the number of labels in the i^{th} instance, the methods differ, mainly, in the reversibility, number of classifiers used and size of the data set after the transformation.

In [20], the authors change the input instances, represented by feature vectors, in order to explore the co-occurrence of relationships among the classes. They do so by expanding a feature set, adding a new feature for each label. Next, the

Table 1. Summary of the algorithm-independent methods

Transformation Approach	Transformation Reversibility	Number of classifiers	Number of instances
Label-based	Yes	L	Same
Instance Elimination	No	Same	Reduced
Label Creation	Yes	Same	Same
Label Elimination	depends on the elimination criterion	Same	Same
Label Decomp. Add.	Yes	$\sum (l_i - 1)$	Increased
Label Decomp. Mult.	Yes	$\prod l_i$	Increased

algorithm-dependent methods are introduced. In [19], dependencies between the different labels are explored through a collective approach. It does so by learning parameters for each possible pair of labels.

2.2 Algorithm Dependent Approach

As the name of this approach suggests, the methods following this approach have been proposed to specific algorithms. The advantage of this approach is that, by concentrating on a particular algorithm, the method may present a better performance in difficult real-world problems than the algorithm independent approaches.

Decision Trees

An extension of the alternating decision tree learning algorithm [17] for multi-label classification is also proposed in [12]. The alternating decision tree learning algorithm induces Alternating Decision Trees, a generalization of DTs. Its inductive principle is based on boosting. The proposed multi-label version is based on AdaBoost [16] and ADTBoost [17]. This multi-class algorithm extends ADTs by decomposing multi-class problems using the one-against-all approach.

In another work with DTs [11], the authors modify the C4.5 algorithm [32] for the classification of genes according to their function. A gene of the yeast *S. cerevisiae* may simultaneously belong to more than one class. Thus, this is a typical multi-label problem. The C4.5 algorithm uses a measure of entropy to define the tree nodes. This measure was originally defined for single-label problems. The authors modified the formulae in order to allow its use in multi-label problems. Another modification was the use of leaves of the tree to represent a set of class labels. When the leaf reached in the classification of an instance contains a set of classes, a separate rule is produced for each class. The authors claim that they could also have produced rules that predict a set of classes and improve the comprehensibility of the rules generated.

Support Vector Machines

Several of the recent works in multi-label classification employ Support Vector Machines (SVMs) [46]. SVMs are Large Margin Classifiers (LMC) [2] that minimize the ranking loss. LMC are ML techniques that place the decision frontier in a position that maximizes the distance between itself and the patterns belonging to each class.

The binary decomposition approach for multi-label problems has been partially studied in [33]. In this work, the authors investigate the use of SVMs for the multi-label classification of gene functional categories. The authors used a heterogeneous data set, generated by the combination of two data sets: gene expression data and phylogenetic profiles. According to the authors, this combination provided a more accurate picture of overlapping subsets of the gene functional classes. As a result, it leads to a better classification performance. They also observe that this improvement is not uniformly distributed among the different classes, thus the combination should only be tried if there is evidence of its benefits.

In [13], a similar method based on SVMs is proposed by the authors. In this paper, the authors also propose a new feature selection method for multi-label data sets. In another paper from the same authors, [14], they propose Rank-SVM, a linear model based on Kernel functions. As the name might suggest, this model follows the ranking approach and minimizes the ranking loss. For this model, the authors define a ranking system, which orders the labels according to their output value, and a predictor for the number of labels to be selected, named threshold-based method. This model is compared against a Binary-SVM model for multi-label classification and Boostexter using a bioinformatics data set, the Yeast data set. This data set contains the gene expression levels and phylogenetic profiles of selected genes. The target function is the prediction of the functional classes of a gene. In the experiments carried out, Rank-SVM outperformed the other two models.

One more method based on SVMs is proposed in [1]. When SVMs are employed for multi-label classification problems, the classification task is divided among several SVMs. The processing time is proportional to the number of kernel computations performed. The authors employed modified SVMs, which allows the simultaneous training of a set of SVM classifiers by using a single optimization procedure. In their approach, a single optimization procedure for the classifiers allows a shared use of the kernel matrix information among them. As a result, a reduction in the learning complexity and training time are obtained, without loss in the classification performance. The performance of the proposed model was evaluated using a set of documents in a text mining task.

A set of SVMs was also adopted by [40], where a multiclass problem was decomposed into a set of binary problems using the one-against-all strategy. Experiments were performed using a data set of protein subcellular localization prediction. Kernel functions are also used in [31] [34] and [35].

Other Techniques

Zhang and Zhou propose in [51] a new multi-label learning algorithm based on K-NN, named ML-kNN. This model uses a lazy-learning approach. For each instance, the labels associated with the k-nearest neighbor instances are retrieved. A membership counting function is employed to count the number of neighbors associated with each label. The maximum a posteriori principle is used to define the label set for a new instance. The authors compare the performance of their algorithm against SVMs, ADTBoost.MH and BoosTexter. They use in the comparison the Hamming Loss, One-error, Coverage, Ranking Loss and Average Precision. In the experiments, they used the same Yeast gene functional data set used by [13]. In this data set, the maximum number of labels can be larger than 190. The results were very similar to those obtained by the other approaches.

Specific parametric mixture models are proposed by [44] [45] for multi-label and multi-class classification. The method was used for document classification using web pages. The experimental results were compared to several ML techniques, like Naive Bayes, K-NN and SVMs.

Two extensions of the Adaboost algorithm to enable their efficient use in multi-label problems are proposed and investigated in [38] [39]. The first extension is a modification of the evaluation of the prediction performance of the induced model by checking its ability to predict the correct set of labels for an input instance. The second extension changes the goal of the learning model to be the prediction of a ranking of labels for each input instance. The model is evaluated by its ability to correctly predict the high-ranking labels. These methods were evaluated using document classification data. Another work based on boosting was investigated in [49]. In this work, the author proposed an ensemble approach that is independent on the base classifier used. The proposed approach was applied to synthetic data and real multimedia data.

A multi-label learning algorithm based on class association rules is proposed in [41]. The algorithm, named multi-class multi-label associative classification (MMAC), is divided into three modules: rules generation, recursive learning and classification. Three measures for accuracy evaluation were also investigated in this work.

In [18], a maximal figure-of-merit (MfoM) learning algorithm initially proposed by three of the authors to binary classification is generalized for multi-label problems. The algorithm is experimentally compared with other ML algorithms in a text classification task.

In [22] the authors propose a new multi-label approach for dealing with data flows. For such, they use active learning in order to perform online multi label learning. Their approach is evaluated in a content-based video search application.

Finally, in [53], a classification algorithm based on entropy is used for information retrieval. In this work, the authors use their model to explore correlations among categories in multi-labelled documents.

3 Performance Evaluation

Finally, it is necessary to define how to evaluate the classification results. Different from single-label classification, where the classification of an instance is either correct or wrong, in multi-label tasks, the result can also be partially correct (or partially wrong). These would be the cases where the classifier correctly produces at least one of the correct labels but either misses one or more of the labels that should be assigned or includes one or more wrong labels in the list of assigned labels.

A few measures have been proposed and investigated to evaluate multi-label classifiers [25] [4] [37][41] [51]. The evaluation criteria can be based on either the multi-label classification made by a classifier, which uses the labels produced by the classifier for a given instance, or a ranking function, which uses the ranking position associated with each label by the classifier for a particular instance.

A similar division is proposed in [14], which divides the methods used to define the cost function into binary approach and ranking approach. In the binary approach, an output vector with the number of elements equal to the number of classes is used. Given an input vector, a sign function defines the value of each element of the output vector. Those elements with positive values are the labels for the input instance. Thus, a binary classifier can be used for each output element or class. Figure 8 illustrates this binary representation approach. It is interesting to notice that a neural network with three output nodes could be easily trained with a data set based on this representation.

Multi-Label Problem:		Output vector:		
Instance	Classes	A	B	C
1	A, B	1	1	0
2	A	1	0	0
3	A, B	1	1	0
4	C	0	0	1

Fig. 8. Binary representation for a multi-label problem

It is important to observe although a sign function is similar to a threshold function, being equal if the threshold value is zero, a heuristic can be followed to define the threshold value. For example, it can be adaptively defined or associated with the prior probability of the classes.

For classification-based evaluation, a common metric is the Hamming Loss. In the case of a binary encoding of the labels, like in Figure 8, the Hamming Loss

measures the number of times a pair (instance, label) is misclassified. For such, it uses the average binary error. The smaller the Hamming loss, the better the situation. The perfect situation occurs when its value is equal to 0.

In the ranking approach, it is assumed that the number of labels to be associated with the input instance, L , is previously known. When an input instance is presented to the multi-label classifier, the L labels with the highest output value are selected. An example of this system is the algorithm Boostexter [39].

For ranking-based evaluation, the metrics frequently employed in the literature are One-Error, Coverage and Average Precision. The One-Error measurement measures the number of times the label with the best rank computed by the classification algorithm is not in the set of correct labels of the input instance [37]. Another measurement, Coverage, says how far, on average, it is necessary to go down on the list of labels ordered by rank in order to include all the labels that should have been assigned to the input instance. The third method, Ranking Loss, calculates the average proportion of pairs that are not correctly ordered. The fourth metric, Average Precision, was originally proposed for Information Retrieval. It evaluates the average proportion of labels ranked above a particular desired label and that belong to the set of desired labels.

In [40], the performance was measured by two criteria, the prediction of the number of classes and the set of classes or labels associated with each test example.

4 Related Work and Discussion

Framework proposals for multi-label problems can be found in [4][43] and [52]. The first framework was presented by [4]. In this framework, the authors describe the initial training approaches to deal with multi-label classification and organize them into 4 major groups. They also discuss alternative testing criteria for the evaluation of multi-label classifiers and propose new evaluation metrics. The authors compared the different models and testing criteria using the evaluation measurements proposed in a scene classification problem. In [43], the authors also present experimental results comparing several multi-class methods.

Another work comparing different approaches for multi-label classification is presented in [27]. This paper includes a experimental comparison of six approaches using two data sets: bioinformatics and scene analysis. Several measures are used in this comparative work.

This paper advances the work in [4] [43] and [52] by proposing a new framework to categorize the methods proposed in the literature for multi-label classification and expanding the review of the current works in this area.

Several approaches for multi-label classification combine multi-label classification with hierarchical classification [11] [5] [6]. In hierarchical classification problems, the classes are disposed in a hierarchical structure. For this class of problems, the classes can be seen as nodes in either a tree-like or a direct acyclic graph (DAG) structure [15]. Several applications in text processing and bioinformatics combine these two issues [3][5] [6] [34] [35].

Recently, population-based meta-heuristics, like evolutionary computation [47] and ant colony optimization [10] have been used for multi-label classification problems.

Another promising work is the use of ranking with multi-label classification, where the classifier should predict not only the classes associated with an instance, but the order these classes are associated with the instance [7] [8] [9].

Finally, we believe that there will be a clear increase in the number of real multi-label classification problems and challenges, particularly in the area of bio-informatics, and this is therefore a promising research topic in Machine Learning.

Acknowledgements

The work was partly supported by the Brazilian Research Agencies CNPq and FAPESP.

References

1. Aiolfi, F., Sperduti, A.: Multiclass Classification with Multi-Prototype Support Vector Machines. *Journal of Machine Learning Research* 6, 817–850 (2005)
2. Bartlett, P., Peter, B., Bartlett, J., Schölkopf, B., Schuurmans, D., Smola, A.J.: Advances in Large-Margin Classifiers. The MIT Press, Cambridge (2000)
3. Barutcuoglu, Z., Schapire, R.E., Troyanskaya, O.G.: Hierarchical multi-label prediction of gene function. *Bioinformatics* 22, 830–836 (2006)
4. Boutell, M., Shen, X., Luo, J., Brown, C.: Multi-label semantic scene classification. Technical Report, Department of Computer Science University of Rochester, USA (2003)
5. Blockeel, H., Bruynooghe, M., Dzeroski, S., Ramon, J., Struyf, J.: Hierarchical multiclassification. In: Proceedings of the ACM SIGKDD 2002 Workshop on Multi-Relational Data Mining (MRDM 2002), Edmonton, Canada, pp. 21–35.
6. Blockeel, H., Schietgat, L., Struyf, J., Dzeroski, S., Clare, A.: Decision Trees for Hierarchical Multilabel Classification: A Case Study in Functional Genomics. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS, vol. 4213, pp. 18–29. Springer, Heidelberg (2006)
7. Brinker, K., Fürnkranz, J., Hüllermeier, E.: A Unified Model for Multilabel Classification and Ranking. In: ECAI 2006, pp. 489–493 (2006)
8. Brinker, K., Hüllermeier, E.: Case-Based Multilabel Ranking. In: IJCAI, pp. 702–707 (2007)
9. Brinker, K., Hüllermeier, E.: Label Ranking in Case-Based Reasoning. In: Weber, R.O., Richter, M.M. (eds.) ICCBR 2007. LNCS, vol. 4626, pp. 77–91. Springer, Heidelberg (2007)
10. Chan, A., Freitas, A.A.: A new ant colony algorithm for multi-label classification with applications in bioinformatics. In: Genetic and Evolutionary Computation 2006 Conference (GECCO 2006), Seattle, USA, pp. 27–34 (2006)
11. Clare, A.J., King, R.D.: Knowledge discovery in multi-label phenotype data. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, p. 42. Springer, Heidelberg (2001)

12. de Comite, F., Gilleron, R., Tommasi, M.: Learning Multi-label Alter-nating Decision Trees from Texts and Data. In: Perner, P., Rosenfeld, A. (eds.) MLDM 2003. LNCS, vol. 2734, pp. 251–274. Springer, Heidelberg (2003)
13. Elisseeff, A., Weston, J.: Kernel methods for multi-labelled classification and categorical regression problems. Technical Report. BIOwulf Technologies (2001)
14. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Neural Information processing Systems. NIPS, vol. 14 (2001)
15. Freitas, A.A., de Carvalho, A.C.P.L.F.: A Tutorial on Hierarchical Classification with Applications in Bioinformatics. In: Taniar, D. (ed.) Research and Trends in Data Mining Technologies and Applications, pp. 175–208. Idea Group (2007)
16. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P.M.B. (ed.) EuroCOLT 1995. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995)
17. Freund, Y., Mason, L.: The alternating decision tree learning algorithm. In: Proceedings of the Sixteenth International Conference on Machine Learning, ICML, pp. 124–133 (1999)
18. Gao, S., Wu, W., Lee, C.-H., Chua, T.-S.: An MFoM Learning Approach to Robust Multiclass Multi-Label Text Categorization. In: Proceedings of the International Conference on Machine Learning (ICML 2004), Banff, Canada, pp. 329–336 (2004)
19. Ghamrawi, N., McCallum, A.: Collective Multi-Label Classification. In: Proceedings of the Fourteenth Conference on Information and Knowledge Management (CIKM), pp. 195–200 (2005)
20. Godbole, S., Sarawagi, S.: Discriminative Methods for Multi-labeled Classification. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS, vol. 3056, pp. 22–30. Springer, Heidelberg (2004)
21. Gonçalves, T., Quaresma, P.: A preliminary approach to the multi-label classification problem of Portuguese juridical documents. In: Pires, F.M., Abreu, S.P. (eds.) EPIA 2003. LNCS (LNAI), vol. 2902, pp. 435–444. Springer, Heidelberg (2003)
22. Hua, X., Qi, G.: Online multi-label active annotation: towards large-scale content-based video search. In: Proceeding of the 16th ACM international Conference on Multimedia. MM 2008, Vancouver, British Columbia, Canada, October 26 - 31, pp. 141–150. ACM, New York (2008)
23. Hsu, C.-W., Lin, C.-J.: A comparison of methods for multi-class support vector machines. IEEE Transactions on Neural Networks 13(2), 415–425 (2002)
24. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
25. Karalić, A., Pirnat, V.: Significance level based multiple tree classification. Informatica 15(5), 12 Pages (1991)
26. Lauser, B., Hotho, A.: Automatic multi-label subject indexing in a multi-lingual environment. In: Koch, T., Sølvberg, I.T. (eds.) ECDL 2003. LNCS, vol. 2769, pp. 140–151. Springer, Heidelberg (2003)
27. Li, T., Zhang, C., Zhu, S.: Empirical Studies on Multi-label Classification. In: Proceedings of the 18th IEEE international Conference on Tools with Artificial intelligence. ICTAI, November 13 - 15, pp. 86–92. IEEE Computer Society, Washington (2006)
28. Luo, X., Zincir-Heywood, A.N.: Evaluation of Two Systems on Multi-class Multi-label Document Classification. In: Hadid, M.-S., Murray, N.V., Raś, Z.W., Tsumoto, S. (eds.) ISMIS 2005. LNCS (LNAI), vol. 3488, pp. 161–169. Springer, Heidelberg (2005)

29. McDonald, R., Crammer, K., Pereira, F.: Flexible Text Segmentation with Structured Multilabel Classification. In: Proceedings of the Human Language Technology Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP, 2005), Vancouver, Canada (2005)
30. McCallum, A.: Multi-label text classification with a mixture model trained by EM. In: AAAI 1999 Workshop on Text Learning (1999)
31. Micchelli, C.A., Pontil, M.: Kernels for Multi-task Learning. In: Saul, L.K., Weiss, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems. NIPS 2004, vol. 17, pp. 921–928. MIT Press, Cambridge (2005)
32. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
33. Pavlidis, P., Weston, J., Cai, J., Grundy, W.: Combining microarray expression data and phylogenetic profiles to learn functional categories using support vector machines. In: RECOMB, pp. 242–248 (2001)
34. Rousu, J., Saunders, C., Szedmak, S., Shawe-Taylor, J.: Learning Hierarchical Multi-Category Text Classification Models. In: 22nd International Conference on Machine Learning (ICML 2005), Bonn, Germany, pp. 745–752 (2005)
35. Rousu, J., Saunders, C., Szedmak, S., Shawe-Taylor, J.: Kernel-based Learning of Hierarchical Multilabel Classification Models. Journal of Machine Learning Research 7, 1601–1626 (2006)
36. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
37. Shen, X., Boutell, M., Luo, J., Brown, C.: Multi-label machine learning and its application to semantic scene classification. Storage and Retrieval Methods and Applications for Multimedia. In: Yeung, M.M., Lienhart, R.W., Li, C.-S. (eds.) Proceedings of the SPIE, vol. 5307, pp. 188–199 (2003)
38. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Machine Learning 37(3), 297–336 (1999)
39. Schapire, R.E., Singer, Y.: BoosTexter: A boosting-based system for text categorization. Machine Learning 39(2/3), 135–168 (2000)
40. Su, C.-Y., Lo, A., Lin, C.-C., Chang, F., Hsu, W.-L.: A Novel Approach for Prediction of Multi-Labeled Protein Subcellular Localization for Prokaryotic Bacteria. In: Computational Systems Bioinformatics Conference, CSB Workshops, Palo Alto, USA, pp. 79–82 (2005)
41. Thabtah, F.A., Cowling, P., Peng, Y.: MMAC: A New Multi-Class, Multi-Label Associative Classification Approach. In: Perner, P. (ed.) ICDM 2004. LNCS, vol. 3275, pp. 217–224. Springer, Heidelberg (2004)
42. Tikk, D., Biró, G.: Experiments with multi-label text classifier on the Reuters collection. In: Proc. of the International Conference on Computational Cybernetics (ICCC 2003), Siófok, Hungary, pp. 33–38 (2003)
43. Tsoumakas, G., Katakis, I.: Multi-Label Classification: An Overview. International Journal of Data Warehousing and Mining 3(3), 1–13 (2007)
44. Ueda, N., Saito, K.: Parametric mixture models for multi-topic text. In: Neural Information Processing Systems 15 (NIPS 15), pp. 737–744. MIT Press, Cambridge (2002)
45. Ueda, N., Saito, K.: Single-shot detection of multi-category text using parametric mixture models. In: ACM SIG Knowledge Discovery and Data Mining (SIGKDD 2002), pp. 626–631 (2002)

46. Vallim, R.M.M., Goldberg, D.E., Llorà, X., Duque, T.S.P.C.: A New Approach for Multi-label Classification Based on Default Hierarchies and Organizational Learning, IWLCS. In: The 11th International Workshop on Learning Classifier Systems, part of the Genetic and Evolutionary Computation 2008 Conference (GECCO 2008), Atlanta, Georgia, USA (accepted) (2008)
47. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)
48. Xu, Y.-Y., Zhou, X.-Z., Guo, Z.-W.: Weak learning algorithm for multi-label multi-class text categorization. In: International Conference on Machine Learning and Cybernetics, 2002. Proceedings, vol. 2, pp. 890–894 (2002)
49. Yan, R., Tesic, J., Smith, J.R.: Model-shared subspace boosting for multi-label classification. In: Proceedings of the 13th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining. KDD 2007, San Jose, California, USA, August 12–15, pp. 834–843. ACM, New York (2007)
50. Yu, K., Yu, S., Tresp, V.: Multi-label informed latent semantic indexing. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 258–265 (2005)
51. Zhang, M.-L., Zhou, Z.-H.: A k-nearest neighbor based algorithm for multi-label classification. In: Proceedings of the 1st IEEE International Conference on Granular Computing (GrC 2005), Beijing, China, pp. 718–721 (2005)
52. Zhou, Z.: Mining Ambiguous Data with Multi-instance Multi-label Representation. In: Alhajj, R., Gao, H., Li, X., Li, J., Zaïane, O.R. (eds.) ADMA 2007. LNCS, vol. 4632, p. 1. Springer, Heidelberg (2007)
53. Zhu, B., Poon, C.K.: Efficient Approximation Algorithms for Multi-label Map Labeling. In: Aggarwal, A.K., Pandu Rangan, C. (eds.) ISAAC 1999. LNCS, vol. 1741, pp. 143–152. Springer, Heidelberg (1999)
54. Zhu, S., Ji, X., Xu, W., Gong, Y.: Multi-labeled Classification Using Maximum Entropy Method. In: Proceedings of Annual ACM Conference on Research and Development in Information Retrieval (SIGIR 2005), pp. 274–281, Salvador, Brazil (2005)

Computational Intelligence in Biomedical Image Processing

Felix Bollenbeck¹ and Udo Seiffert²

¹ Leibniz Institute for Plant Genetics and Crop Plant Research, Gatersleben, Germany

bollen@ipk-gatersleben.de

² Fraunhofer Institute IFF, Magdeburg, Germany

udo.seiffert@iff.fraunhofer.de

Summary. The description of phenotypical properties of living organisms has emerged as an urging topic in biology. In the context of automated high-throughput processing of large stacks of serial section light-microscopic images the recognition and segmentation of all prevailing biological structures and tissues is a crucial task for high-throughput processing of such data.

Where in image segmentation contexts many algorithms are based on a global, often physical model of an image, employed in variational formulations, supervised classifier schemes are based on individual local data, sub-images or pixels, i.e. points in a feature space, which are processed individually. Global approaches often explicitly constraint desired properties, e.g. closed contours, where they lack the possibility of learning expert knowledge from reference data as known in supervised computational intelligence paradigms.

In this study, we show that the segmentation of biological images, characterized by non-uniform image features, significantly benefits from combining global physical models and local feature-based supervised classification. We employ an entropy-based voting of *optimal* feed-forward networks by cross-validation architecture selection and global registration-based segmentation.

Apart from the theoretical framework and experimental studies, we demonstrate the usefulness of a combined approach in an application for automated generation of high-resolution 3-D models from serial section data.

1 Introduction Section

1.1 Processing of Biomedical Image Data

More and more applications of computational intelligence paradigms have been concerned in the processing and understanding of data sampled from living things. Along with huge progress in high-throughput assays on quantitative molecular properties, and according challenges with data analysis, there also is strong interest for high quality data for visual inspection of

intact structures in organisms. Such data is especially insightful, when displaying observations in their natural spatial or even spatio-temporal context.

While in medical imaging such data is often sampled on macroscopic scale, e.g. as prerequisite to operative interventions, microscopic imaging in biology often aims at understanding the functional interplay of organs and structures. Since three-dimensionally resolved quantitative assays of molecular properties have become widely available, existing works successfully address the integration of structural and quantitative data generating virtual transparent images of living organisms on a microscopic scale.

The exploration of mammalian organism is located on the forefront of technical advance for obvious reasons, works on humans [1] and mice [38] deliver impressive examples. Nevertheless recent work on invertebrate [35] as well as plant organs [16][37] have been published, demonstrating that three-dimensional analysis and visualization of these organism is of great interest to scientists.

While individual imaging modalities differ in their respective demands consistent, automated and efficient processing and analysis of biomedical image data is characterized by the following observations:

- Image data is subject to biodiversity
- Imaging conditions are often difficult to standardize
- Identification of relevant structures often needs expert knowledge

The observation of inevitable diversity amongst specimen can generally be considered of wanted feature of imaging and should be incorporated into analysis and modelling image data. The removal of noise caused by inhomogeneous imaging conditions or scanner defects, on the other hand is highly feasible, demanding well tuned algorithms and statistical methods. Due to a certain degree of fuzziness or ambiguity of relevant structures in biological material, their identification often requires *a priori* or expert knowledge. Here, computational intelligence paradigms and algorithms resembling such expert knowledge are crucial for automated processing.

In this context, our work is concerned with the construction of structural 3D models based on light microscopic (LM) images of plants. Robust model construction particularly involves the identification and labelling of relevant tissues classes, while we observe the mentioned inhomogeneity with sampled data. By exploiting the synergy of supervised classifiers and free-form deformation models we aim at computationally efficient tissue identification. This is also a key aspect to automated model generation in regards high reproducibility and efficiency

1.2 Related Work

Image registration and segmentation are probably the most prominent problems in image processing and computer vision. Particularly in image segmentation, on which we focus herein, a high number of algorithms and methods

have been described. Theoretical frameworks employed include graph theory [43], (markovian) statistics [49], variational methods [11], and supervised classification schemes [50]. In the context of biomedical imaging, image segmentation has received great attention with the increasing availability of image-based diagnostics in the past decade. The authors of [46] provide an extensive compilation of algorithms and applications.

Variational methods relate to contour evolution or deformation of a reference segmentation. Geodesic curve representations have received great attention in biomedical segmentation, particularly in binary segmentation scenarios (see [26], [44], and [48] for recent studies), while free-form deformation of labelfields are successfully applied multi-class applications (see [40], [39]). In [39] the authors introduce a method to improve overall accuracy by combining multiple variational tasks in a classification-*boosting* manner.

Approaches addressing segmentation by classification in an extracted feature space, using non-linear separation as described in [23], can discriminate more complex features such as different textures. Particularly artificial neural networks have extensively been used in the field (see for an overview [36] and [51] for a recent example). In previous works in the context of automated 3D model generation [7], [8] we have investigated the performance of neural network classifiers for tissue segmentation delivering high-accuracy segmentations with LM image data. Methods towards the integration of neural networks and variational methods exist, e.g described in [25] and [39], though not employing networks for discriminative tasks.

2 A Digital Grain Project

2.1 Serial Section Imagery

Amongst other, native three-dimensionally resolved imaging, such as confocal laser scanning microscopy or nuclear imaging, three-dimensional reconstruction from serial section imaging is a well established technique, holding the advantage to capture histological detail (cells, fibres, etc.) in the resolution available to light-microscopy [41]. For the possibility to precisely identify tissues and structures on a cellular level it is considered indispensable by biologists, despite serial section data comes with high demands in processing, reconstruction, and classification of large stacks of section images, implying an urgent need for automated processing. Works based on serial section imagery of mammalian organisms (human [1], rat [41], mice [22]) as well as recent works on plants [16] generally show high resolutions and level of detail, thereby producing amounts of data easily extending several gigabytes for a single specimen. Since the object of interest is essentially destroyed for digitization, the standardisation and re-assembly of raw-data is a prerequisite to further analysis. Processing of serial section data comprises three major tasks:

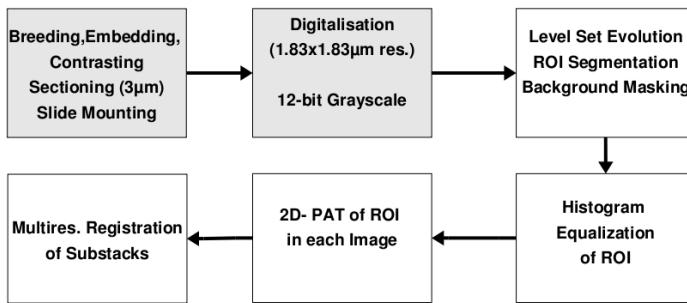


Fig. 1. Workflow of data acquisition and preprocessing steps (biologically specific preparation and digitizing are grayed)

- Removal of noise and disturbances caused by sectioning and slide mounting
- Reconstruction of three-dimensional coherence of sections
- Identification and labelling of biologically relevant material

While noise removal and alignment of images (*registration*) can be considered somewhat classical image processing problems, here we focus on the recognition of biologically relevant materials giving rise to a classification problem and appropriate algorithms for resembling biological expert knowledge. Nevertheless since sensible preprocessing of raw-data is crucially connected to the successful application of computational intelligence paradigms, we are giving a brief description of the pipeline.

We introduce a novel algorithm for fast and accurate recognition of tissue types which combines supervised image feature-based classification schemes and a variational intensity registration based method. By introducing supervised machine learning paradigms we aim at improving automated segmentation in an application comprising three-dimensional modelling based on serial section data of developing barley grains. In the modelling we are considering inter-individual diversities amongst specimen, necessitating to consider and process a multitude of specimen for each 3-D model. Thereby large data volumes in the magnitude of several tens of thousands of section images are generated, the total amount of image data captured is more than 100 GB.

Manual sectioning and mounting of slices on glass slides by a technician produces dust particles and artifacts, which appear mostly as high frequency noise and also larger objects in digitized slice scans. To remove artifacts and background noise, the *region-of-interest* is segmented, masked and embedded in a uniform background. Thresholding or other low-level segmentation algorithms fail due to textured nature of images, while segmentation based on morphological operations and blob-analysis is susceptible to varying image quality.

Variational Geodesic Active Contours

Active contour models are considered very robust frameworks for segmenting objects in images in the presence of noise. From the formulation by *Kass et al.* [21] using a parametric model of a dynamic curve, implicit (geodesic) representation of curves as level-sets solved either in a PDE [9] or variational framework [52] have been introduced, allowing topological changes and efficient numerics.

The description of the curve as the zero-level set of an embedding function $C(T) = \{(x, y) | \phi(t, x, y) = 0\}$ delivers the well known level-set evolution equation

$$\frac{\partial \phi}{\partial t} + F|\nabla \phi| = 0 \quad (1)$$

which necessitates re-initialization of ϕ during evolution. The initialisation problem leads to small time-steps and unstable curve evolution, apart from the re-initialisation to a signed-distance function being computationally costly. The authors of [27] have introduced a variational formulation solving the re-initialisation problem by introducing a penalising term

$$P(\phi) = \int_{\Omega} \frac{1}{2} (|\nabla \phi| - 1)^2 dx dy \quad (2)$$

into the energy functional forcing the level-set function to a signed distance function during evolution.

This improvement of the traditional formulation makes it especially applicable for background masking of large sets of serial section images for two main reasons:

1. A significantly larger timestep for evolution can be chosen
2. The initial level set function does not have to be a signed distance function

While (1.) the choice of a more aggressive step-width is certainly beneficial, the relaxed constraints (2.) on initialisation can be exploited to apply the variational formulation in a multi-scale multi-resolution image pyramid framework. By starting the evolution on larger scales s (the variance of a gaussian kernel) with coarser spatial resolutions r , successively moving to smaller scales with higher resolution thereby controlling the step-width, the number of iterations as well as the iteration time can be reduced to reduce overall time drastically (see fig. 2).

Principal Axes Transform

Although standardization of first order image moments using the well known *principal axes transform (PAT)* [3] is somewhat straightforward, it is nevertheless a vital step for further analysis, bringing geometrical features of image data into spatial correspondence.

While the estimation of moments based on image intensities is considered susceptible to outliers in terms of image distortions, Gaussian and Cauchie

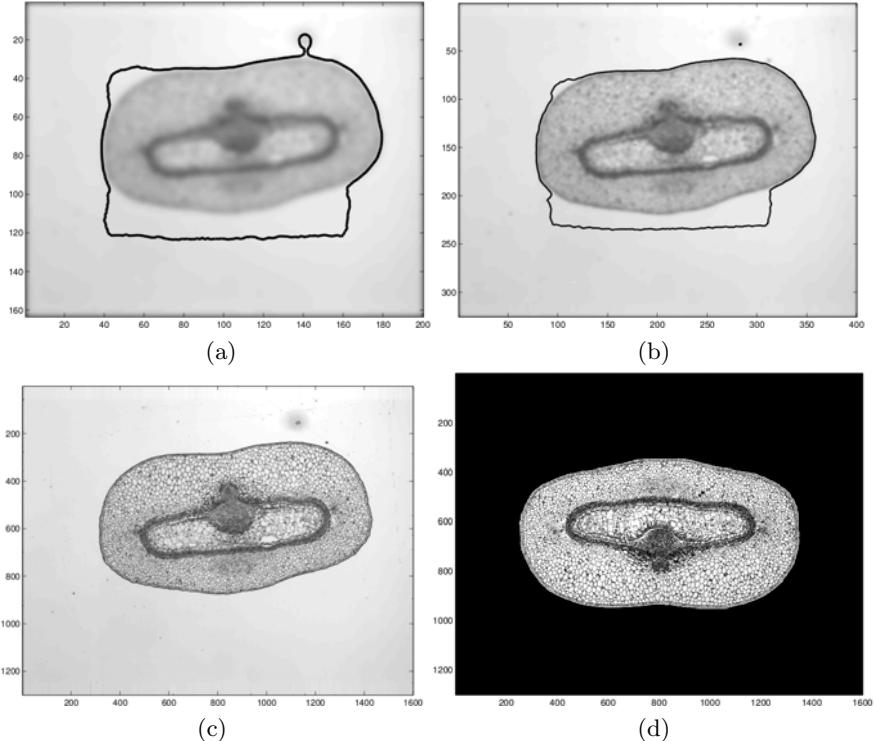


Fig. 2. Preprocessing of Images: Segmentation of *ROI* with *Geometric Active Contours* in a multi-scale multi-resolution framework. Figures 2a, 2c show the curve evolution with different scales s and resolutions r . 2a: $s = 21.9$, $r = 14.64\mu m/px$, 2b: $s = 5.4$, $r = 3.66\mu m/px$, 2c: $s = 3.3$, $r = 1.83\mu m/px$. 2d: A masked image with standardized moments.

priors have been suggested [41], damping the influence of gridpoint intensities far away from the mass centroid. The exact delineation of the ROI mask with active contouring however allows the estimation of image moment based on the binary mask, thereby circumventing the problem of possibly inhomogeneous intensity distribution *within* the object of interest. By applying the *PAT* to section stack images, yielding uniform geometry, further processing, i.e. based on extracted features is significantly improved as well as a good *bulk initialisation* for the subsequent stack reconstruction is established.

2.2 Three-Dimensional Reconstruction from Serial Section Series Images

The advantage of serial section LM imagery in high resolution of histological structures comes with the disadvantage of high costs in the reconstruction of data. The object of interest is physically destroyed for digitization, whereby

the three-dimensional coherence is lost. Due to manual handling and mounting of sections to glass slides are generally miss-aligned when digitized. In order to re-establish this coherence for a three-dimensional reconstruction, where an interactive processing of thousands of section images being clearly unfeasible, each slice image needs to be automatically transformed towards an optimal correspondence of histological structures. In order to reconstruct the sectioned object in z-direction the whole image stack is registered by finding an optimal superposition of all slice images in the stack.

Matching two-dimensional section images based on landmarks is highly discouraged: While the manual identification of landmarks is time consuming and subjective, automated landmark detection is difficult in biomedical images especially in the absence of unique structures. While transforming images based on intensities is clearly more costly in finding an optimal transformation, it can be generally considered less error prone than matching images based on landmarks which could possibly miss-identified beforehand. Non-rigid transformations are not considered here because it is desired that the inter-individual anatomical variability between individuals remains unaffected by the registration process [30].

Intensity Based Registration

In the context of intensity-based image registration a broad range of image-to-image metrics are described in the literature [32]. While statistical and information theoretic metrics [29] addressing images possibly coming from different domains (*inter-modality* registration), images $R, T : \Omega \subset \mathbb{R}^2 \mapsto \mathbb{R}$ with similar intensity distributions allow an intuitive and computational cheap metric by direct comparison of quadratic difference intensity magnitude at gridpoints

$$D(T, R) := \frac{1}{2} \int_{\Omega} (R(x) - T(x))^2 dx \quad (3)$$

An optimal superposition of all stack images $\mathbf{R} := (R_1, \dots, R_M)$ is found by minimizing the sum of squared differences of images intensities over the space of linear transformations $\varphi = (\varphi_1, \dots, \varphi_M)$ for each image:

$$D(R \circ \varphi) := \sum_{i=1}^M \int_{\Omega} (R(x)_{i-1} \circ \varphi_{i-1} - R(x)_i \circ \varphi_i)^2 dx \stackrel{!}{=} \min \quad (4)$$

Extended Image Metric

Stack registration based on pairwise alignment is critical, since small miss-alignments are propagated through the whole image stack. By extending the image metric to incorporate intensity values of images within a neighborhood N

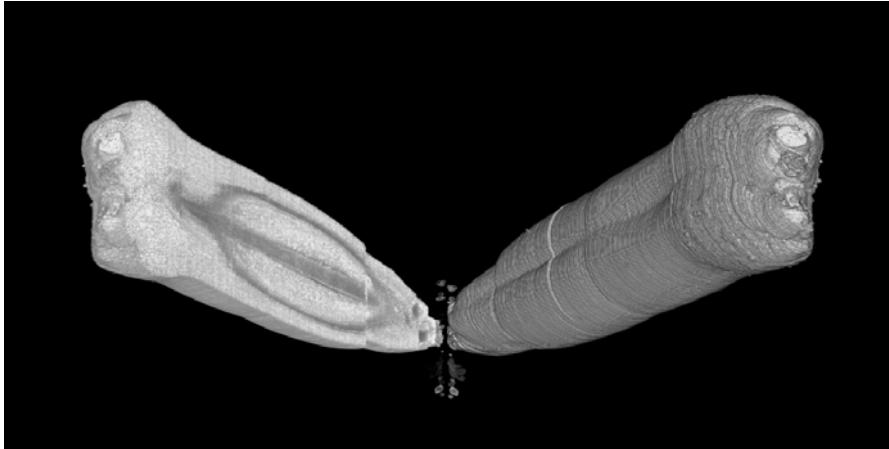


Fig. 3. Reconstruction by stack registration: A volume rendering of aligned intensity images. A virtual longitudinal cutting plane on left side shows the correct alignment of the internal grain structures. (The dataset is approximately 4 GB large).

$$D(R \circ \varphi) := \sum_{i=1}^M \sum_{j=i-N}^{i-1} \int_{\Omega} (R_j(x) \circ \varphi_j - R_i(x) \circ \varphi_i)^2 dx \stackrel{!}{=} \min \quad (5)$$

a more robust registration is obtained.

Multiresolution Framework

Due to the textured nature of images, the surface of metric values in parameter space is unsMOOTH, possibly causing the search stuck in local optima. Performing registration in a multi-resolution image hierarchy has the advantage of successively improving the registration result by choosing appropriate parameters for each scale, thereby

- Avoiding high metric frequencies using large scales initially
- Optimizing in narrowed search space for smaller scales

while being computationally much more efficient.

Grid Search

Exhaustive search, sampling the whole search space on a regular grid can be considered unfeasible with large images for long computing times, but on the other hand guarantees a global optimum at sufficient grid density. While numerical optimization schemes tend to get stuck in local optima near the initialization point, i.e. flipped images, an exhaustive search on larger scales and smaller resolutions finds the *correct* transform even if metric values differ only slightly.

Numerical Optimization in Transform Space

The global optimum found via an extensive search on a low resolution and large scales is used to initialize the registration on finer level. For further refinement of the registration result, now gradient descent optimization is employed. Depending on the resolution on scale level of the hierarchy, optimization parameters such as step length are annealed when switching to smaller scales.

Gradient descent optimizers are generally susceptible to different scales of the input variables. Rotation and translation have dynamic ranges differing in two magnitudes. Algorithms specifically addressing this problem have been proposed using evolutionary optimization strategies [45], generally showing slower convergence compared with gradient descents. We therefore employ a rescaling of the gradient vector for a gradient descent search.

The employed registration scheme is robust to image distortions, since the annealed step width on finer levels prohibit to skew the overall alignment.

Three-Dimensional Histological Voxel Data

By using the described preprocessing and registration pipeline the three-dimensional correspondence of histological structures in section images is restored. Reconstructed grains can be used to provide arbitrary microscopic views of internal cell structures, as shown in fig. 3. Intact data is a prerequisite for further processing, such as automated recognition and segmentation of biologically relevant materials.

3 A Hybrid Algorithm for Automated Tissue Recognition

In the following we describe a hybrid algorithm to recognize and segment any desired number of tissue classes in images by combining global and local segmentation approaches. Segmentation of image data can be considered a major task in how a machine possibly interacts in its environment. Here a first abstraction from a two-dimensional intensity distribution from the scanner array towards *contents* of an image is obtained.

Image segmentation in computational intelligence has received great attention, either as generating data input to intelligent systems and learning classifiers or by employing computational intelligence schemes to efficiently process image data.

While the image segmentation problem has been formulated and addressed from various domains, such as graph theory, diffusion processes, markov models etc., it can generally be considered a machine learning problem of extracting meaningful information of image raw-data.

From a machine learning viewpoint, a sensible criterion would be to divide the large amount of segmentation approaches into supervised and unsupervised algorithms. Granularity on the other hand is interesting: While some

algorithms handle a global representation of an image, i.e. a graph structure, or as function values for a PDE approach in the evolution of a closed curve, others work on local data, sub images, individual pixels and extracted features, i.e. ranging from simple intensity thresholding to supervised classifiers like neural networks or support vector machines.

Global approaches generally have the advantage of allowing the incorporation of constraints and a priori information in the recognition and delineation of image regions, such as evolving a closed curve, desired size and shape etc.. Often the behavior of such algorithms is interfaced by a small set of control parameters, by which the identification of an object is controlled.

Correct estimation of these parameters is often critical when global segmentation algorithms are not employed in a supervised framework. Further a uniform global representation of the (filtered) image can be ambiguous, which causes global methods to fail, this particularly holds in the case of biomedical images.

Segmentation based on local data however allows the extraction of sufficiently discriminative image features for segmentation by classifying local feature vectors. When employed in classification framework, a discriminant feature space can be compiled systematically, using adequate classifiers, samples do not have to be linearly separable. In the literature supervised classification schemes are successfully applied mostly in applications where the identification of regions can be considered hard [12].

Pixel-wise classification generally is prone to over-segmentation, producing distorted topologies in applications. Post-processing, or *cleaning* of speckled labelling is *ad hoc* and can be considered heuristical.

Combining the advantages of both global and local approaches is therefore highly feasible. We introduce a hybrid algorithm based on a global statistical segmentation and pixel-wise classification based on extracted image features.

3.1 Multi Class Image Segmentation

Segmentation is commonly defined as partitioning an image into non-overlapping regions that are homogeneous with respect to a certain criterion [33]. The homogeneity criterion is generally defined by the application, or „what we define it to be“ [28]. While most segmentation problems are formulated to address finding and extracting regions belonging to *one class* w.r.t to a certain criterion, the general case is to partition the image into an arbitrary number of classes.

Formally, a mapping $S : \Omega \mapsto \{1, \dots, M\}$ of the domain Ω of an image $I : \Omega \subset \mathbb{R}^2 \mapsto \mathbb{R}$ to a set of M *class labels*, where $S_i := S(\Omega) = i$ and $\bigcup_{i=1, \dots, M} S_i = \Omega$ and $S_i \cap S_j = \emptyset, i \neq j$.

In the processing of biomedical image data, segmentation often is the crucial step which necessitates expert knowledge in most cases. Here the image-raw data is abstracted for the retrieval of relevant data within the domain of discourse, and is the basis for further quantification and analysis.

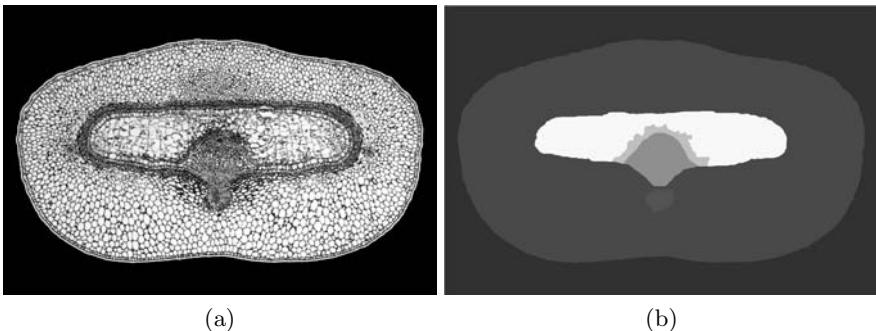


Fig. 4. A sample segmentation of a serial section image. **4a** Standardized intensity image of a *barley grain* dataset. **4b** Segmentation result displaying biologically relevant regions in the image.

An automatic segmentation of sections is characterized by several requirements:

- A multitude of tissues must be recognized
- Images lack clearly defined edges and structures
- The identification of tissue types needs expert knowledge

which necessitates the use of algorithms incorporating *a priori* information for robust multiclass-segmentation, where solely intensity-based techniques are clearly deficient. A possible encoding of *a priori* information (e.g. class numbers and densities) are manual reference segmentations of sample images, thereby facilitating segmentation algorithms resembling expert knowledge autonomously without human interaction or control being necessary.

3.2 Global Registration-Based Statistical Segmentation

We solve the segmentation task via a registration problem, exploiting a result of the sectioning: Although the segmentation of an individual image is hard, neighbouring images are topologically similar in their respective tissue mapping. We therefore transform a reference segmentation based on the similarity of the reference image slice to the target image slice using a variational approach.

For section images $R, T : \Omega \subset \mathbb{R}^2 \mapsto \mathbb{N}^+$ with a reference segmentation $S : R \subset \Omega \mapsto \{1, \dots, M\}$ of R a transformation of S to segment R correctly is found by an optimal deformation of R to T . Given the reference intensity image R and template intensity image T , the goal is to find a transformation to maximize the similarity between both images. The transformation u is now non-parametric or *free-form* in contrast to the affine transformation for the stack registration, allowing arbitrary pixel displacements.

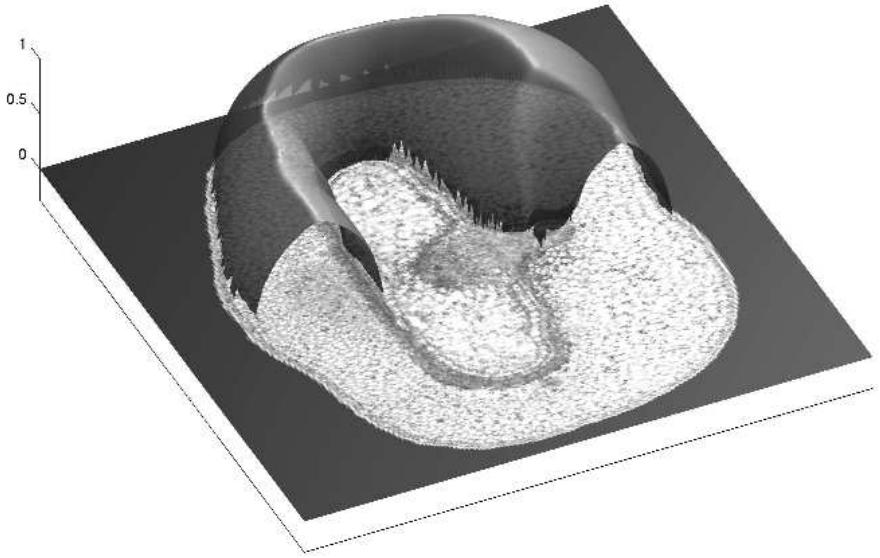


Fig. 5. Confidence of the global segmentation: Border regions are prone to errors, resulting in lower confidence values, which are plotted by a sliced surface overlay of confidence values

The problem of finding an optimal deformation u maximizing an image-to-image metric D is ill-posed, and therefore needs to be regularized, leading to a *deformable registration* problem [32] [6]

$$\mathbf{J}(u) := D(R, T; u) + \alpha S(u) \stackrel{!}{=} \min. \quad (6)$$

For the regularizer or smoothing term S we elide the physically motivated *elastic* potential introduced by [6] in favour of the optical-flow based diffusion regularizer

$$S(u) = \frac{1}{2} \sum \int_{\Omega} \| \nabla u \|^2 d\mathbf{x} \quad (7)$$

introduced by [13] [14] [34]. The use of SSD as an image metric has been proposed [42] [17] for intra-modality registration, and is used here. Delivering explicitly smooth displacement fields, the diffusion-registration problem can be solved in $\mathcal{O}(N)$ per registration step using state-of-the-art solution schemes as in [24] [34], on which we rely. As a result of the registration procedure we do not employ the registered (transformed) template image, but the transformation u , in form of a displacement field. The displacement field u which constitutes an optimal transformation of T to match R with respect to the regularization is applied to transform S to obtain a segmentation of T .

The evaluation of the deformation of the reference segmentation leads to labellings lying in between the discrete label set $\{1, \dots, M\}$ of the M distinct

materials. Therefore the obtained segmentation S^* is mapped back to the original label interval by direct binning.

Segmentation based on a global, variational approach allows the incorporation of statistical, *a priori* information [26][5]. We in turn use a geometric prior $P(S)$ to assign a posterior to pixels $x \in \Omega$ belonging to the segmented class for a segmentation $P(S|I)$.

$P(S|I)$ is modelled with a distribution based on the geometric center of each segment S_i , $i = 1, \dots, M$ assigning $x \in \Omega$ a probability $p(x = S_i|I)$ based on the distance transform to the centroid of S_i

$$p(x = S_i|I) := \begin{cases} 0 & S(x) \circ u = i \\ f(d(x; S, u)) & \text{else} \end{cases} \quad (8)$$

Where f is a decay function based on segments obtained by the deformation approach employing reference S and displacement field u for x .

Thereby a statistical representation of the deformation-based segmentation is obtained. This is based on the observation, that the global segmentation algorithm is more likely prone to errors close to the border of segments, since the deformation is driven by the intensities in *all* gridpoints, producing a lesser degree of accuracy in those regions, which can in turn be captured using the employing the statistical segmentation approach (see figure 5).

3.3 Local Approach by Supervised Neural Networks for Image Segmentation Based on Image Features

Artificial Neural Networks (ANN) have delivered a powerful paradigm to signal and image processing problems. Since the introduction of the back-propagation algorithm more than twenty years ago they have been successfully applied in almost any image processing task, the authors of [36] provide a detailed review and bibliography of algorithms and applications.

As stated by *Kolmogorov's* theorem, multilayer neural networks can theoretically perform any regression or discrimination task given correct weight adaption and architecture [20], while being computationally very efficient due to the intrinsic parallelism.

In image segmentation, various types of neural networks are used for classification of (sub) images with different granularity. Supervised feed-forward ANN have been successfully used in pixel-wise classification based on extracted features [2][18]. By learning and adapting from training data, such approaches are highly beneficial with data being fuzzy to a certain degree for their ability to generalize. This holds particularly with biomedical image data, where structures are subject to natural diversity and therefore ambiguous, and can often only be correctly identified by an expert.

Other classifiers, such as *support vector machines* (SVM) or prototype-based classifiers also have been successfully used in image segmentation, particularly for their ability to utilize kernel methods and arbitrary similarity metrics. Nevertheless SVM are disregarded here for not allowing intrinsic

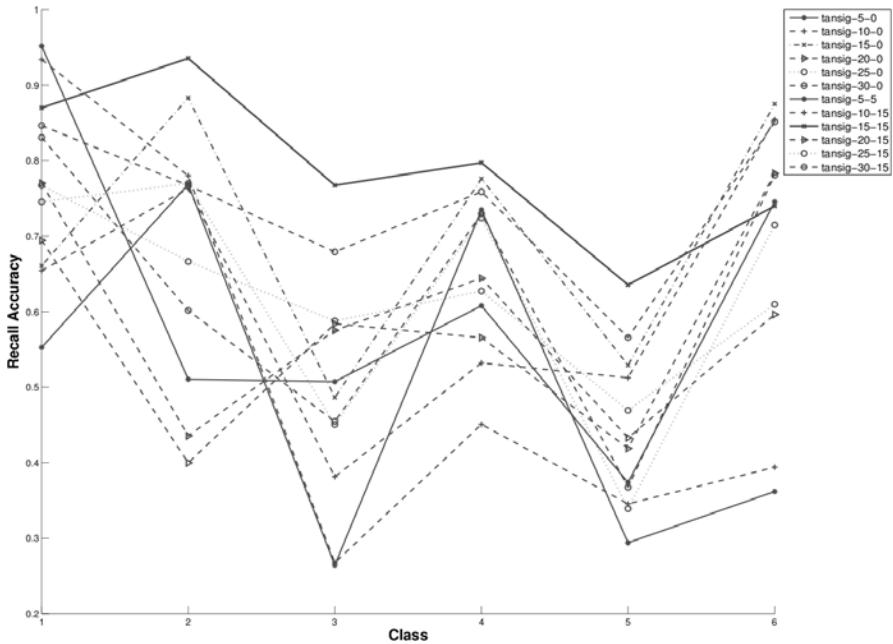


Fig. 6. Estimation of generalisation ability of networks architectures using cross validation experiments: Average accuracy per class of different network architectures in supervised pixel-based classification. 12 different architectures with one and two hidden layers and different number of hidden layer nodes with *tan-sigmoid* transfer functions were tested in 10-fold cross validation experiments sampled from a test dataset comprising extracted image features (see sec. B.3) and expert-created reference segmentations. An optimal prediction of class labels is achieved with a 15, 15 node network and two hidden layers. Networks with smaller number of nodes and one hidden layer deliver lower recall accuracies, while larger networks with more weights seem to over-adapt.

multi-classification, while prototype-based methods necessitate a large number of receptive fields for high accuracy with complex data and multiple classes.

Feed-forward networks for multi-class segmentation have proven to deliver good generalisation and high segmentation accuracy [8]. By training networks with reference ground-truth (*training*) targets and extracted feature data, a system resembles (*recalls*) expert knowledge for segmentation: Vectors of local features associated to pixels are classified to one of the possible classes, composing the segmentation of the image.

Discriminative Features

Regardless of the classifier used, classification (i.e. segmentation) accuracy crucially depends on a discriminative feature space. This holds in particular, as the risk of miss-classification increases with the number of classes.

In order to compile a discriminative feature vector, spatial variations in the gray-value distributions of local pixel neighborhoods, using linear and non-linear operations, are used. Describing contextual data by means of patterns in the local structure of images (i.e. *texture*) is known as powerful features for image classification and segmentation in the literature. Therefore we compiled a high-dimensional feature vector using several approaches for multitexture features and local structure:

- A set of *Gabor*-filters with different scales and angles
- *Gray-Level Cooccurrence Matrices* for different spatial relations and derived statistics
- *Discrete Wavelet Packet Frame* decomposition, employing the best-encoding wavelet tree
- *Range, Entropy, StdDev*, within different local neighborhoods

While it is generally not clear *a priori* what image information can be extracted with filter parameters, optimal parameters were estimated via spectral analysis, guaranteeing high discriminative power for the feature set.

Optimal Network Architectures from Cross Validation Experiments

Multilayer feed-forward networks can implement strong discriminative power for non-linearly separable problems when appropriate models are selected. The question is, how to choose an optimal network architecture for a given classification task, i.e. the number of *hidden layers* and respective number of *nodes, transfer functions* and so forth, beforehand to applying strategies for weight-adaption, i.e. *training* of the network.

While the number of networks delivering good adaption to training data is arbitrarily large, *optimality* of a network architecture certainly refers to how the network generalizes on unlabelled data.

Automated selection of an optimal network architecture has been subject to research [47], employing network pruning and construction techniques. While information based regularization of networks size and structure using upwind statistical or information-theoretic criteria like BIC, MDL or MML [15], relating networks architecture to available data, a straightforward yet powerful method is to directly optimize networks generalization error on test data using cross-validation experiments [4].

3.4 A Hybrid Approach

The integration of variational and supervised segmentation approaches can significantly improve the overall segmentation accuracy, when image data lacks unique features for segmentation. A variational formulation using a global image model allows the natural incorporation of *a priori* knowledge and thereby certain constraints on the topology of segmentations, but has the downside of being little accurate when the (filtered) intensity image is ambiguous or such data is not linearly separable.

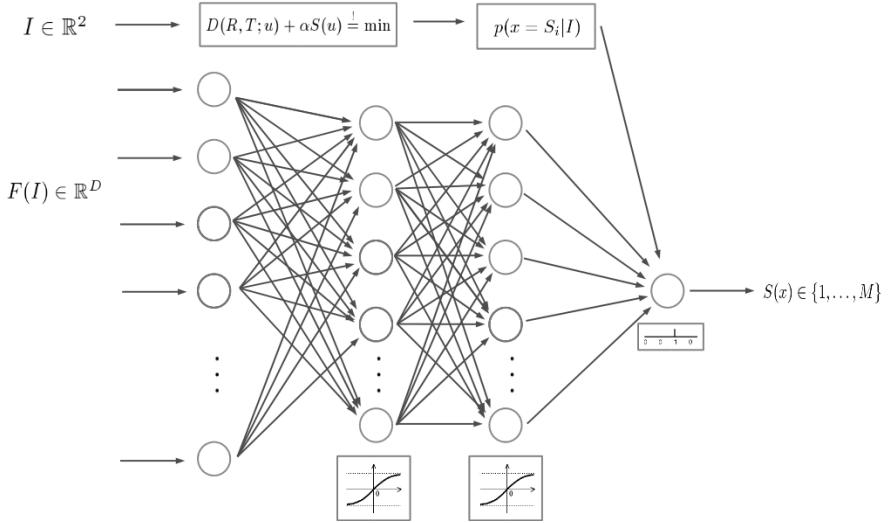


Fig. 7. Hybrid system architecture for joint supervised and variational classification of pixels for image segmentation. A feed-forward network performs the classification based on a higher dimensional feature vector $F(I)$, using two hidden layers. Global image segmentation obtained by the variational approach and computed pixel-wise probabilities are used for voting by a weighted *compete* function.

Segmentation on a pixel basis has the advantage that the classifier, i.e. a neural network, can exploit the information provided in the extracted feature data when there is no linear separation. Implementing much higher discriminative power has made such schemes successful in applications where the segmentation is considered hard due to image properties or multiple classes. Such systems generally have a much higher number of free parameters, ranging from feature extraction methods, possible feature reduction or feature space transforms, to parameters of the classifiers themselves. The problem even with well-tuned systems is that they tend to produce *over segmented* images: Since pixels are individually processed, miss-classified pixels are scattered, where uniform segments are wanted.

While addressing such over-segmentation problems in *postprocessing* using e.g. median filters can solve application-specific problems, it can generally be considered unsatisfactory. Therefore the inclusion of global a priori knowledge is considered one central problem in image segmentation with Artificial Neural Networks [36].

Such a priori knowledge has to be of statistical nature in order to employ it for the computation of class-membership posteriors in combination with a classifier. While the raw net output can be considered to deliver a statistical class voting, it does not implement the constraints of a global statistical model. On the other hand, directly employing fully labelled training-set

images for deriving *a priori* knowledge descriptor is clearly diminishing the generalization ability of the system.

We therefore use the global deformation-based model to obtain statistical descriptions based on *a priori* knowledge *and* the observed data. The system implements the integration of a supervised classifier for local pixel features and global variational segmentation for statistical segmentation by a *model mixing*.

The voting of both algorithms is averaged in a weighted compet function: While the neural network produces a raw output vector where the maximal component determines the class estimation in a usual classification scheme, the *entropy* of the raw output vector can be considered an indicator of how confident the prediction is. A mixing coefficient λ is based on the normalized entropy of the raw net output $H(\mathbf{o}(x))$, being a measure of the certainty of the network voting, and the probability $p(x = S_i|I)$ of the variational formulation

$$\lambda := \frac{1}{2} \left(- \sum_{i=1}^{|o|} o_i \cdot \log_{|o|} o_i + p(x = S_i|I) \right)$$

obtaining an averaged (weighted by λ and $1 - \lambda$ respectively) output by a *compet* transfer function.

Thereby the advantages of both approaches are mutually combined: In large regions were the variational approach delivers high probabilities based on a priori given reference data and global image features, the effect of over-segmentation is avoided. In border or ambiguous regions, were the solely intensity driven global approach is likely to fail, accurate recognition and segmentation in feature space is delivered by the neural network.

4 Results - Segmentation of Serial Section Data

The processing of biomedical image data often produces data amounts, which prohibit interactive processing or reviewing of such. In the case of serial sectioning LM images with spatial resolution of a few micrometers comprise datasets easily exceeding several gigabytes in size.

Enhancement and registration for reconstruction of section data is entirely data driven and do not necessitate any *a priori* knowledge.

Data abstraction by recognition and segmentation into biologically relevant tissues and materials requires expert knowledge: First the introduced segmentation algorithm requires labelled training data, second, the assessment of the method requires expert created segmentations which are considered a *gold standard* reference for the segmentation into prevailing tissues.

While the fraction of labelled data in order to train the system is small (0.05), manual segmentation of a full dataset comprising thousands of images for benchmarking is clearly too time consuming. Architecture selection

(see sec. 3.3) and system evaluation is performed on a substack of images labelled by an expert taken as ground truth reference data, regardless of possible biases in terms of inter- or intraobserver variability, since the segmentation task is intrinsically subjective in terms to what experts identify certain structures as relevant.

The ability of the system to learn and resemble expert knowledge from training data assessed by the segmentation accuracy on reference data gives an initial estimate of the performance on full application data, which comprise several tens of gigabytes of serial section image data in our case.

4.1 Image Segmentation Accuracy

Methods for evaluating segmentation algorithms in biomedical image processing have received attention in the literature [10]. Published approaches include very problem specific indicators, such as the size and shape of a segmented organ in medical imaging, as well as methods directly related to the nature of the segmentation algorithms, i.e. metrics based on parametrized curves representations.

The *Hammoude distance* [19] can be considered a well established generalized metric in biomedical applications by a pixel-by-pixel comparison and aggregation of a contour's inside and outside against a reference. While from a machine learning viewpoint such metric would rather be termed *accuracy*, it can be naturally extended to the multiclass-case in such a context, yielding more specific accuracy indicators such as *confusion matrices* [21].

Such pixel-wise classification accuracy is certainly a fundamental indicator of the performance of a segmentation algorithm. Nevertheless by uniformly aggregating and averaging the number of correctly predicted pixels, the two-dimensional relation of image gridpoints is neglected: A miss-classified pixel spatially close to the correct border is equally reducing accuracy as a pixel far away. Since this observation is universal to image segmentation problems, we suggest an extended accuracy measure for segmentation, weighting miss-classified pixels with the minimal distance to the reference or *true* class border, allowing a quantification of *how wrong* a miss-classification is.

Distance to closest point (DCP) is used as a weighting factor the computation of accuracies: Let $S(x) = i$ be the true class of a pixel $x \in \Omega$ and $\hat{S}(x)$ be the predicted class and $i \neq j$, e.g. a miss-classification. The DCP is given by

$$d(x, S_i) = \min_{\hat{x} \in S_i} ||x - \hat{x}|| \quad (9)$$

delivering a weighting for the aggregation for per-class or overall loss. Apart from the universally motivation of such a metric, it is specifically useful for the performance evaluation of the suggested hybrid algorithm for resolving possibly improved accuracy in border regions.

Table 1. Assessment of the segmentation algorithms on the reference dataset: Examination of the segmentation accuracy for each tissue class, displaying the per-pixel true positive rates w.r.t. to the reference segmentation (see Section 4.2) and values of the distance based error metric (see sec. 4.1). The deformation-based segmentation algorithms performs worse especially on classes with lower density (e.g. *smaller* class segments), while the average distance of miss-classified points is low. The combined approach delivers a significantly higher overall accuracy particularly improving for smaller classes with lower average miss-classification distance.

Method \ Class (class density)	1 (42.18)%	2 (40.64%)	3 (1.21%)	4 (4.37%)	5 (1.22%)	6 (10.42%)
Deform. accuracy	0.82 ± 0.0007	0.83 ± 0.0014	0.40 ± 0.1174	0.82 ± 0.0863	0.53 ± 0.1867	0.53 ± 0.0403
Deform. weighted loss	14.60 ± 0.487	10.70 ± 0.189	48.17 ± 0.478	42.18 ± 0.858	42.23 ± 0.435	34.00 ± 1.062
Hybrid accuracy	0.95 ± 0.0005	0.94 ± 0.0005	0.71 ± 0.1638	.088 ± 0.0319	0.72 ± 0.1884	0.82 ± 0.0140
Hybrid weighted loss	13.21 ± 0.125	6.90 ± 0.039	48.18 ± 0.280	35.81 ± 0.270	37.16 ± 0.109	34.44 ± 0.619

4.2 Experiments

Apart from theoretical properties, especially of classification schemes, performance and robustness of algorithms on real life data is hard to predict. We tested the proposed semi-supervised system and preprocessing pipeline in the context of processing large datasets of serial section images, involving distorted and inhomogeneous data quality, typical for biomedical image data.

To test the algorithm on real life data, we compiled a sub-stack of images of serially sectioned barley grains. The substack comprising 80 images of 1600×1200 pixels were fully manually segmented by a biologist expert.

Preprocessing and Image Stack Registration

Since the interdependency of images within the registration problem, stack reconstruction is highly susceptible to errors in the initial processing, especially with large image stacks. By using the active contours approach in the described multi-scale evolution (see fig. 2) a robust initial foreground segmentation is obtained.

The result of the registration process can instantaneously be reviewed in a three-dimensional display of the full dataset, i.e. by direct volume rendering of intensity data (fig. 3). Correct initial alignment is crucial for deformation-based segmentation, which was obtained by the described multi-strategy approach. By using an exhaustive grid search on larger scales, the optimization process is much more robust against small image distortions and local minima, thereby yielding a correct reconstruction result.

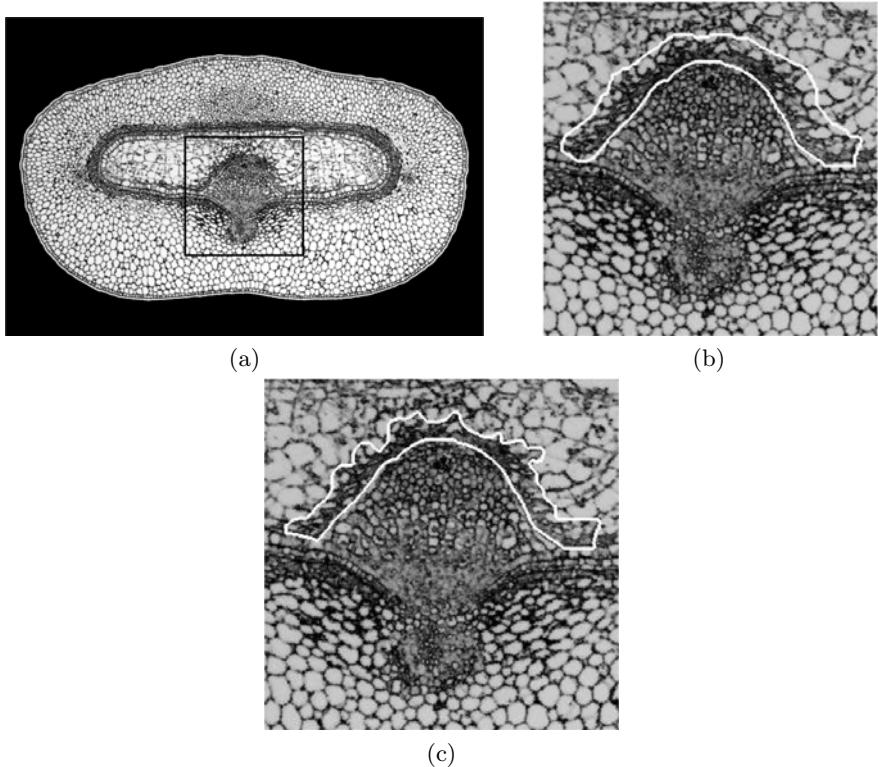


Fig. 8. Segmentation results by means of a cutout of the most challenging image part of the deformation-based and hybrid approach: **SB**: Contour obtained by the global deformation-based segmentation result. The segmented boundary is inaccurate where image data is ambiguous. **SC**: The hybrid approach uses extracted local image features for finding a more exact boundary.

Registration-based Segmentation

In order to assess the accuracy which can be achieved using a global image model for multi-class segmentation of section images, we test the registration-based approach on a substack of expert-labeled images thereby yielding a benchmark for the hybrid approach.

We used 10% of equidistantly sampled reference segmentations as templates for the deformation based on image intensities. The remaining 90% of images were accordingly processed, and the resulting segmentation evaluated against the ground truth data.

While the approach generally delivers good segmentation accuracy in terms of the average true positive rates displayed in table II, smaller classes with lower overall density are segmented less accurate, since there is no particular representation within the global model. The displayed average distance of miss-classified gridpoints to the correct border gives an indication of the

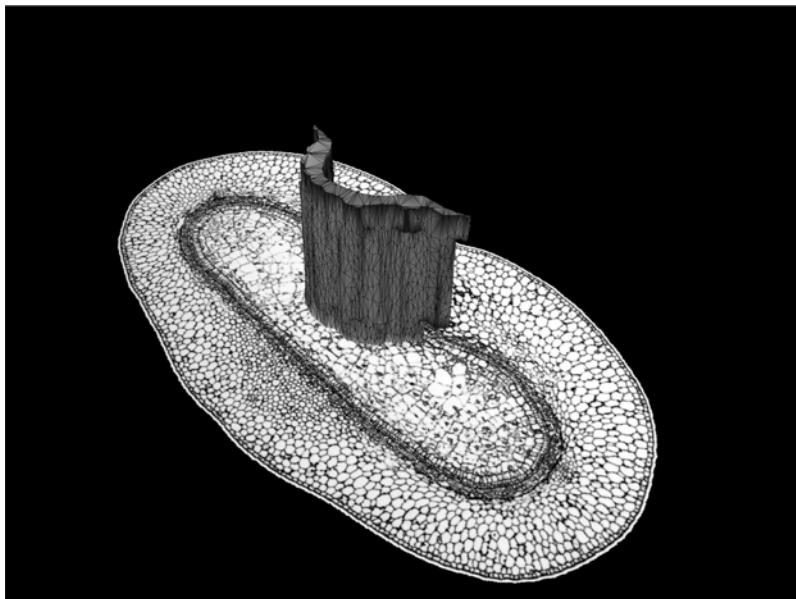


Fig. 9. Segmentation results in three-dimensional data: The iso-surface rendering shows the segmentation result of the hybrid approach (see fig. 8) displaying an underlying slice image. The three-dimensional surface structure of the segmented contour is indicated by displaying surface patches.

spatial weighting of these miss-classification. Here a significantly small error is observed, since the deformation-based method is not susceptible to over-segmentation, meaning that miss-classification is spatially constrained.

Neural Networks Model Selection

While the variational formulation of a deformation-based segmentation described in section 3.2 has a small number of parameters such regarding the PDE solution schemes, the performance tuning of the supervised system includes feature extraction and adaption, model selection and training of the system.

The features described in section 3.3 comprise a 50-dimensional feature space in which classification is performed. Using the described spectral analysis for filter parameter estimation and best representing trees with wavelet packets decomposition only the 10 lowest LDA scored [31] features were rejected.

N-fold cross validation as randomized hold out test-train scheme is well established for model selection thereby being entirely data-driven. *N*, referring to the number of equally sized partitions, is set to 10, being more robust than e.g. *leave-one-out* cross validation for obvious reasons. Based on the assumption that application data is similarly distributed to training data, the

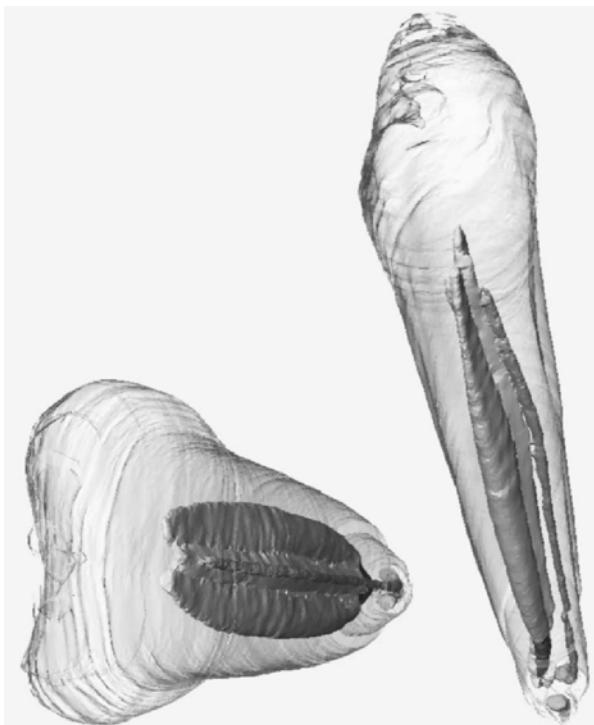


Fig. 10. Two perspectives of a digital barley grain showing an iso-surface rendering of the dataset displayed in figure 3 segmented into multiple tissue types illustrating the context of the application of the segmentation algorithm

model delivering the highest average test score is selected. Cross-validation accuracy is commonly higher than true model accuracy, nevertheless employing cross-validation for model-selection also delivers an estimate of accuracies to expect with the application data.

For the construction of an optimal feed-forward network architecture, feature data and true (expert) class labels were used to iterate over different models with one or two hidden layers, while the neuron transfer functions were universally set to *tangens-sigmoid* for the sake of simplicity. Weight-adaption was performed using momentum gradient descent and adaptive learning rate on the back-propagated *mean-square-error*. The performance of each network architecture was evaluated using a standard 10-fold cross validation, whereby the average recall accuracy on the hold-out data was taken as a performance measure.

Model selection experiments are summarized in figure 6. While the expressive power of one-hidden-layer networks seems to be too low for good generalisation, two-hidden layer architectures delivered high accuracy values. Further

increase of nodes did not improve performance, possibly over-adapting the larger number of parameters to train-data.

Combined Supervised and Variational Segmentation Algorithm

The described hybrid approach superimposes two additional capabilities: The non-linear discrimination of local image feature data by learning such relations from an human expert and an adaptation of a global topology of segments. Therefore we expected the overall accuracy to be significantly higher than the solely global model, which is proven by the accuracies displayed in table 11. Particularly smaller classes are better represented and recognized, now being represented in learning or adaption strategy. The effect of over-segmentation on the other hand is avoided by the described model averaging, utilizing the topological information provided by the global model, which is proven by the overall lower average distance of miss-classified pixels.

Visual inspection of the segmented curve for a class with small overall density (class 4 table 11) over the grayscale image depicted in figure 8 shows that the hybrid approach recognizes local image structures on the segments boundary which are miss-classified otherwise.

Figure 9 depicts the segmentation result in 30 consecutive images for the same class and a single underlying intensity image. The displayed patches on the isosurface indicate that the local image features are incorporated through segmentations of slice images.

5 Conclusion

In this study we show that in the context of biomedical imagery the incorporation of learning schemes can significantly improve the performance of segmentation algorithms. Especially in applications were expert knowledge is complex, yet necessary and can not be formulated explicitly, as often the case in biomedical data, supervised classification is beneficial.

The processing of biological images, characterized by non-uniform image features significantly benefit from combining global physical models and local feature-based supervised classification using neural networks. Apart from the theoretical framework, the usefulness of a combined approach in an application involving the automated generation of high-resolution 3-D models from serial section data is shown.

Insofar, the commonly utilized features of neural networks, such as adaptability and trainability, are extended in this study by their ability to process multimodal data coming in this case from two different segmentation strategies as shown in figure 7. Computational intelligence paradigms have once more proven of their ability to provide an excellent framework to solve image processing tasks necessitating the learning of expert knowledge.

References

1. Ackerman, M.J.: The visible human project. *Proceedings of the IEEE* 86(3), 504–511 (1998)
2. Ahmed, M.N., Farag, A.A.: Two-stage neural network for volume segmentation of medical images. *Pattern Recognition Lett.* 18(11-13), 1143–1151 (1997)
3. Alpert, N.M., Bradshaw, J.F., Kennedy, D., Correia, J.A.: The Principal Axes Transformation - A Method for Image Registration. *The Journal of Nuclear Medicine* 31, 1717–1722 (1990)
4. Andersen, T., Martinez, T.: Cross validation and mlp architecture selection. In: *International Joint Conference on Neural Networks (IJCNN 1999)* (1999)
5. Bresson, X., Vandergheynst, P.: A variational model for object segmentation using boundary information and shape prior driven by the mumford-shah functional. *Int. J. Comput. Vision* 68(2), 145–162 (2006)
6. Broit, C.: Optimal Registration of Deformed Images. PhD thesis (1981)
7. Brüß, C., Bollenbeck, F., Schleif, F.-M., Weschke, W., Villmann, T., Seiffert, U.: Fuzzy Image Segmentation with Fuzzy Labelled Neural Gas. In: *Proc. of the 14th European Symposium on Artificial Neural Networks ESANN* (2006)
8. Brüß, C., Strickert, M., Seiffert, U.: Towards Automatic Segmentation of Serial High-Resolution Images. In: *Proceedings Workshop Bildverarbeitung für die Medizin* (2006)
9. Caselles, V., Catte, F., Coll, T., Dibos, F.: A geometric model for active contours in image processing. *Numer. Math.* 66, 1–31 (1993)
10. Chalana, V., Kim, Y.: A methodology for evaluation of boundary detection algorithms on medical images. *IEEE Trans. Med. Imaging* 16(5), 642–652 (1997)
11. Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Transactions on Image Processing* 10(2), 266–277 (2001)
12. Chapelle, O., Haffner, P., Vapnik, V.: Support Vector Machines for histogram-based Image Classification. In: *IEEE Transactions on Neural Networks*, special issue on Support Vectors (1999)
13. Fischer, B., Modersitzki, J.: Fast Diffusion Registration. In: Nashed, M.Z., Scherzer, O. (eds.) *Inverse Problems, Image Analysis, and Medical Imaging: Contemporary Mathematics*, vol. 313. AMS (2002)
14. Fischer, B., Modersitzki, J.: FLIRT: A Flexible Image Registration Toolbox. In: Gee, J.C., Maintz, J.B.A., Vannier, M.W. (eds.) *WBIR 2003. LNCS*, vol. 2717, pp. 261–270. Springer, Heidelberg (2003)
15. Grünwald, P.D., Myung, I.J., Pitt, M.A. (eds.): *Advances in Minimum Description Length: Theory and Applications*. MIT Press, Cambridge (2005)
16. Gubatz, S., Dercksen, V.J., Brüss, C., Weschke, W., Wobus, U.: Analysis of barley (*Hordeum vulgare*) grain development using three-dimensional digital models. *The Plant Journal* 6, 779–790 (2007)
17. Hajnal, J.V., Saeed, N., Soar, E.J., Oatridge, A., Young, I.R., Bydder, G.M.: A registration and interpolation procedure for subvoxel matching of serially acquired mr images. *Journal of computer assisted tomography* 19(2), 289–296 (1995)
18. Hall, L.O., Bensaid, A.M., Clarke, L.P., Velthuizen, R.P., Silbiger, M.S., Bezdek, J.C.: A comparison of neural network and fuzzy clustering techniques in segmenting magnetic resonance images of the brain. *IEEE Trans. Neural Networks* 3(5), 672–682 (1992)

19. Hammoude, A.: Computer-assisted endocardial border identification from a sequence of two-dimensional echocardiographic images. PhD thesis, Univ. Washington, Seattle, WA (1988)
20. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall PTR, Englewood Cliffs (1998)
21. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *Int'l J. Comp. Vis.* 1, 321–333 (1987)
22. Kaufman, M.H., Brune, R.M., Baldock, R.A., Bard, J.B., Davidson, D.: Computer-aided 3-d reconstruction of serially sectioned mouse embryos: its use in integrating anatomical organization. *Int. J. Dev. Biol.* 41, 223–233 (1997)
23. Kim, K.I., Jung, K., Park, S.H., Kim, H.J.: Support Vector Machines for Texture Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1542–1550 (2002)
24. Kohavi, R., Provost, F.: Glossary. *Machine Learning* 30, 271–274 (1998)
25. Vilarino, D.I., Cabello, D., Pardo, X.M., Brea, V.: Cellular neural networks and active contours: a tool for image segmentation. *Image and Vision Computing* 21(2), 189–204 (2003)
26. Leventon, M.E., Grimson, W.E.L., Faugeras, O.: Statistical shape influence in geodesic active contours. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1 (2000)
27. Li, C., Xu, C., Gui, C., Fox, M.D.: Level set evolution without re-initialization: A new variational formulation. In: *CVPR 2005: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, vol. 1, pp. 430–436. IEEE Computer Society Press, Washington (2005)
28. Lindeberg, T.: Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision* 30(2), 77–116 (1996)
29. Maes, F., Collignon, A., Vandermeulen, D., Marchal, G., Suetens, P.: Multi-modality image registration by maximization of mutualinformation. *Trans. on Med. Imaging* 16(2), 187–198 (1997)
30. Maschino, E., Maurin, Y., Andrey, P.: Joint registration and averaging of multiple 3D anatomical surface models. *Comput. Vis. Image Underst.* 101(1), 16–30 (2006)
31. McLachlan, G.J.: Discriminant Analysis and Statistical Pattern Recognition. Wiley Interscience, Hoboken (2004)
32. Modersitzki, J.: Numerical Methods for Image Registration. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford (2004)
33. Pal, N.R., Pal, S.K.: A review on image segmentation techniques. *Pattern Recognition* 26(9), 1277–1294 (1993)
34. Papenberg, N., Schumacher, H., Heldmann, S., Wirtz, S., Bommeresheim, S., Ens, K., Modersitzki, J., Fischer, B.: A Fast and Flexible Image Registration Toolbox: Design and Implementation of the general approach. In: *Bildverarbeitung für die Medizin*, pp. 106–110. Springer, Heidelberg (2007)
35. Pereanu, W., Hartenstein, V.: Digital three-dimensional models of Drosophila development. *Cur. Opin. Genet. Dev.* 14, 382–391 (2004)
36. Petersen, M.E., de Ridder, D., Handels, H.: Image processing with neural networks — a review. *Pattern Recognition* 35, 2279–2301 (2002)

37. Pielot, R., Seiffert, U., Manz, B., Weier, D., Volke, F., Weschke, W.: 4d warping for analysing morphological changes in seed development of barley grains. In: Ranchordas, A., Araújo, H. (eds.) VISAPP (1), pp. 335–340. INSTICC - Institute for Systems and Technologies of Information, Control and Communication (2008)
38. Ringwald, M., Baldock, R., Bard, J., Kaufman, M.H., Eppig, J., Richardson, J.E., Nadeau, J.H., Davidson, D.: A Database for Mouse Development. *Science* 265, 2033–2034 (1994)
39. Rohlfing, T., Maurer Jr., C.J.: Multi-classifier framework for atlas-based image segmentation. *Pattern Recognition Letters* 26(13), 2070–2079 (2005)
40. Rueckert, D., Sonoda, L.I., Hayes, C., Hill, D.L.G., Leach, M.O., Hawkes, D.J.: Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Transactions on Medical Imaging* 18(8), 712–721 (1999)
41. Schmitt, O., Modersitzki, J., Heldmann, S., Wirtz, S., Fischer, B.: Image Registration of Sectioned Brains. *Int. J. Comput. Vision* 73(1), 5–39 (2006)
42. Schormann, T., Zilles, K.: Three-Dimensional linear and nonlinear transformations: An integration of light microscopical and MRI data. *Human Brain Mapping* 6, 339–347 (1998)
43. Shi, J., Malik, J.: Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 888–905 (2000)
44. Silveira, M., Marques, J.S.: Level set segmentation of dermoscopy images. In: 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008, pp. 173–176 (2008)
45. Styner, M., Gerig, G.: Evaluation of 2d/3d bias correction with 1+1 es-optimization – technical report 179. Technical report, Image Science Lab, ETH Zurich (1997)
46. Suri, J.S., Setarehdan, S.K., Singh, S.: Advanced Algorithmic Approaches to Medical Image Segmentation: State-Of-The-Art Applications in Cardiology, Neurology, Mammography and Pathology. Springer, Heidelberg (2002)
47. Tibshirani, R.: A comparison of some error estimates for neural network models. *Neural Computation* 8, 152–163 (1996)
48. Tsai, A., Yezzi Jr., A., Wells, W., Tempany, C., Tucker, D., Fan, A., Grimson, W.E., Willsky, A.: A shape-based approach to the segmentation of medical imagery using level sets. *IEEE Transactions on Medical Imaging* 22(2), 137–154 (2003)
49. Tu, Z., Zhu, S.-C.: Image Segmentation by Data-Driven Markov Chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 657–673 (2002)
50. Vandenbroucke, N., Macaire, L., Postaire, J.-G.: Color image segmentation by pixel classification in an adapted hybrid color space. Application to soccer image analysis. *Computer Vision and Image Understanding* 90(2), 190–216 (2003)
51. Wismüller, A., Vietze, F., Behrends, J., Meyer-Baese, A., Reiser, M., Ritter, H.: Fully automated biomedical image segmentation by self-organized model adaptation. *Neural Networks* 17(8-9), 1327–1344 (2004)
52. Zhao, H.-K., Chan, T., Merriman, B., Osher, S.: A variational level set approach to multiphase motion. *J. Comp. Phys.* 127, 179–195 (1996)

A Comparative Study of Three Graph Edit Distance Algorithms

Xinbo Gao¹, Bing Xiao¹, Dacheng Tao², and Xuelong Li³

¹ VIPS Lab, School of Electronic Engineering, Xidian University,
Xi'an 710071, P.R. China
{xbgao, bxiao}@mail.xidian.edu.cn

² School of Computer Engineering, Nanyang Technological University,
50 Nanyang Avenue, Blk N4, 639798, Singapore
dacheng.tao@gmail.com

³ School of Computer Science and Information Systems, Birkbeck College,
University of London, London WC1E 7HX, U.K.
xuelong_li@ieee.org

Summary. Abstract-Graph edit distance (GED) is widely applied to similarity measurement of graphs in inexact graph matching. Due to the difficulty of defining cost functions reasonably, we do research on two GED algorithms without cost function definition: the first is combining edge direction histogram (EDH) and earth mover's distance (EMD) to estimate the GED; the second is introducing hidden Markov model (HMM) and Kullback-Leibler distance (KLD) into GED algorithm. These algorithms are evaluated theoretically and experimentally, and are compared with the GED from spectral seriation, one of the leading methods for computing GED with cost functions. Theoretical comparison shows that the proposed two cost function free GED algorithms have less complexity and characterize graph structure more effectively than spectral seriation method. Experimental results on image classification demonstrate that time occupied by the EDH-based method is 4.4% that of the spectral seriation method with the same correct classification rate, and correct classification rate of HMM-based method is 3.4% greater than that of the other two methods with 3.3% the time consumed by spectral seriation method. Clustering rate of these three methods is basically the same, but HMM-based and EDH-based methods only consume 3.17% and 5.43% the time of spectral seriation method.

Keywords: Terms-Graph edit distance (GED), edge direction histogram (EDH), hidden Markov model (HMM).

1 Introduction

Inexact graph matching [3], [4], [6], [21], [24], [26], [27] has been the focus of research in the areas of computer vision and pattern recognition for over two

decades. One of its key issues is the similarity measurement of graphs, for which graph edit distance (GED) has attracted researchers' attention greatly since it is error-tolerant to noise and distortion. The GED between two graphs is the cost of the least expensive edit operation sequence that is needed to transform a graph into another one.

GED for attributed graphs is computed directly according to the attributes of graphs, but straightforward GED algorithms lack formal underpinning of string edit distance, in view of which the relationship between GED and the size of the maximum common subgraph has been studied [1], the uniqueness of the cost-function has been commented [2], a probability distribution for local graph edit distance has been constructed, etc. GED for non-attributed graphs is computed indirectly, in which graphs are converted into strings and then the GED is computed by dint of string edit distance. The underpinning of edit distance [12], [24] is transferred into GED spontaneously and research of edit distance advances GED development.

However, most of the existing algorithms are seriously dependent on cost functions which are difficult to be defined reasonably in a general way. For this purpose, two methods for computing GED without cost function definition are studied and compared in this paper. Because edit sequence consists of edge operations (insertion, deletion and substitution of edges) and node operations (insertion, deletion and substitution of nodes), which are associated with each other, both of them being considered is unnecessary. In the method based on edge direction histogram (EDH) and earth mover's distance (EMD), GED is related to the structure difference of graphs, i.e. edge direction and edge length, which are characterized by EDH, and the distance of EDHs is computed with EMD. In the GED algorithm combining hidden Markov model (HMM) and Kullback-Leibler distance (KLD), GED is related to the node distribution difference of graphs which is modeled by HMM and the distance of HMMs is computed with KLD. This paper focuses on the evaluation of these two cost function free algorithms, with the GED from spectral seriation [19], an important cost function dependent method, as reference.

The remainder of this paper is organized as follows: In section 2, the idea of cost function-dependent GED is introduced briefly; in section 3, two GED algorithms without cost function definition are described in detail respectively; section 4 provides the theoretical comparison of these algorithms. A set of experiments is presented to compare these methods on real data in section 5 and conclusion is given finally.

2 The Cost Function-Dependent GED Algorithm

The GED from spectral seriation proposed by Robles-Kelly [19] is one of the cost function dependent methods with optimal performance for structure graphs. The flow chart of this method is shown in Fig. 1.

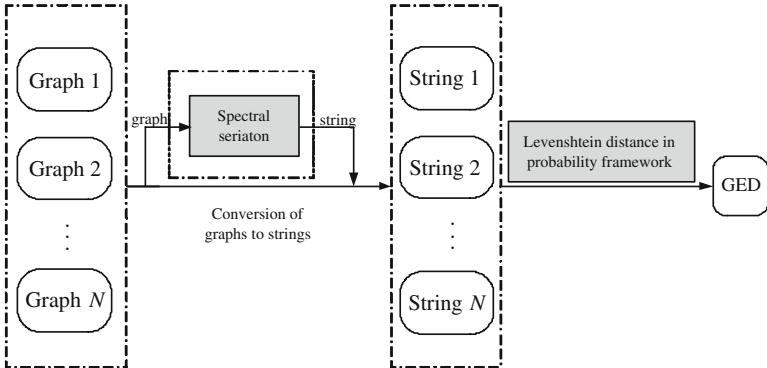


Fig. 1. The GED from spectral seriation. Graph i is converted into String i and GED is computed by the Levenshtein distance of N strings.

Graphs are represented by strings with graph-spectral method, and then the similarity of graphs is measured with the Levenshtein distance in a probabilistic setting.

The Levenshtein distance in probability frame of data graph $G_D = (V_D, E_D)$ and model graph $G_M = (V_M, E_M)$ is computed by the following steps.

Step1: The edit lattice is indexed by two strings $Y = \{y_1, y_2, \dots, y_{|V_D|}\}$ and $X = \{x_1, x_2, \dots, x_{|V_M|}\}$ of data graph and model graph respectively, together with null symbol ε .

Step2: The least expensive path $\Gamma^* = \langle \gamma_1, \gamma_2, \dots, \gamma_k, \dots, \gamma_L \rangle$ is found through the edit lattice based on the idea of Levenshtein distance. Each state $\gamma_k \in (V_D \cup \varepsilon) \times (V_M \cup \varepsilon)$ of the edit path is a Cartesian pair. The diagonal transition on edit lattice $\gamma_k \rightarrow \gamma_{k+1}$, where $\gamma_k = (y_i, x_j)$ and $\gamma_{k+1} = (y_{i+1}, x_{j+1})$, corresponds to the match of an edge (y_i, y_{i+1}) in the data-graph to an edge (x_j, x_{j+1}) in the model graph; A horizontal transition $\gamma_k \rightarrow \gamma_{k+1}$, where $\gamma_k = (y_i, x_j)$ and $\gamma_{k+1} = (\varepsilon, x_{j+1})$, means that the traversed nodes of the model graph are null-matched, i.e. the match of (y_i, ε) and (x_j, x_{j+1}) . Similarly, when a vertical transition $\gamma_k \rightarrow \gamma_{k+1}$ is made, where $\gamma_k = (y_i, x_j)$, and $\gamma_{k+1} = (y_{i+1}, \varepsilon)$, the traversed nodes of the data graph are null-matched, i.e. the match of (y_i, y_{i+1}) and (x_j, ε) .

Step3: Cost functions are defined in probability framework for the above three kinds of elementary matches,

$$\begin{aligned} \eta(\gamma_k \rightarrow \gamma_{k+1}) &= -\ln P(\gamma_k | \phi_X^*(x_j), \phi_Y^*(y_i)) \\ &\quad - \ln P(\gamma_{k+1} | \phi_X^*(x_{j+1}), \phi_Y^*(y_{i+1})) - \ln R_{k,k+1}, \end{aligned} \quad (1)$$

where the edge compatibility coefficient $R_{k,k+1}$ is

$$\begin{aligned}
R_{k,k+1} &= \frac{P(\gamma_k, \gamma_{k+1})}{P(\gamma_k)P(\gamma_{k+1})} \\
&= \begin{cases} \rho_M \rho_D & \text{if } \gamma_k \rightarrow \gamma_{k+1} \text{ is a diagonal transition on the edit lattice, i.e. } (y_i, x_j) \in E_D \text{ and } (y_{i+1}, x_{j+1}) \in E_M \\ \rho_M & \text{if } \gamma_k \rightarrow \gamma_{k+1} \text{ is a vertical transition on the edit lattice, i.e. } (y_i, x_j) \in E_D \text{ and } y_{i+1} = \varepsilon \text{ or } x_{j+1} = \varepsilon \\ \rho_D & \text{if } \gamma_k \rightarrow \gamma_{k+1} \text{ is a horizontal transition on the edit lattice, i.e. } y_i = \varepsilon \text{ or } x_j = \varepsilon \text{ and } (y_{i+1}, x_{j+1}) \in E_M \\ 1 & \text{if } y_i = \varepsilon \text{ or } x_j = \varepsilon \text{ and } y_{i+1} = \varepsilon \text{ or } x_{j+1} = \varepsilon \end{cases} \quad (2)
\end{aligned}$$

and

$$\begin{aligned}
P(\gamma_k | \phi_X^*(x_j), \phi_Y^*(y_i)) &= \\
\begin{cases} \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2\sigma^2}(\phi_X^*(x_j) - \phi_Y^*(y_i))^2\right\} & \text{if } x_j \neq \varepsilon \text{ and } y_i \neq \varepsilon. \\ \alpha & \text{if } x_j = \varepsilon \text{ or } y_i = \varepsilon \end{cases} \quad (3)
\end{aligned}$$

Step4: The cost of edit path is computed with the cost functions for elementary matches:

$$d(X, Y) = C(\Gamma^*) = \sum_{\gamma_k \in \Gamma^*} \eta(\gamma_k \rightarrow \gamma_{k+1}), \quad (4)$$

and it is the GED for G_D and G_M .

3 The Cost Function-Free GED Algorithms

3.1 EDH-Based GED

All edit operations are expressed by edge operations for correlation between edges and nodes and cost of edit operations is related to the graph structure difference which is characterized by EDH. The GED is computed by measuring the EMD of EDHs, and the EDH-based GED is presented in Fig. 2.

The EDH is computed by grouping the edge points falling into edge directions and counting the number of points in each direction. Since edge points are related to shape information closely, the EDH is a very simple and direct way to characterize shape information of an object. It has been applied successfully to image retrieval [9], [14], classification [22] and quality assessment [5], and Kim used EDH to watermark text document images [11]. The EDH is usually normalized to be scaling invariant and Zhang et al [25] further compute the 1-D fast Fourier transform of the normalized EDH to obtain rotation invariance and take it as the final signature of image. With the help of EDH, high level pattern recognition problem is to be solved with relatively simple low-level features. But EDH is seldom used to graph matching. Points in an edge are of the same direction which is the edge direction, and number of points in the direction is the edge length, therefore the EDH for a graph is computed according to the slope and length of edges.

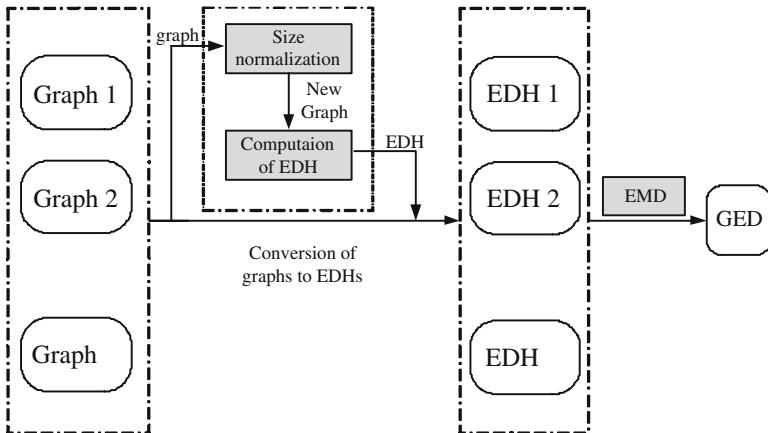


Fig. 2. The GED based on EDH. Graph i is converted into EDH i and GED is computed by the EMD of N EDHs.

EDH for graphs can reflect the structure difference of graphs and reserve information of edit operations directly and sufficiently. For two graphs having the same structure and different size, their EDHs are very different; therefore, as shown in Fig. 2, graphs are normalized before computation of EDH so as to ensure the insensitiveness of EDH to scale. The procedure of computing an EDH for a graph is given as follows.

Step1: The length $l(i)$ and slope $k(i)$ are computed for every edge in the graph, where $i = 1, 2, \dots, N$ and N is the number of edges in the graph. Direction of the i th edge is $D(i) = \arctan(k(i))$. Direction vector D may have some repetitive elements because some edges in the graph may be in the same direction.

Step2: Distinct elements in D are arranged in sort ascending, resulting a new direction vector D' and $|D'| \leq |D|$. Difference of pairwise adjacent elements in D' , (i.e. $D'(i+1) - D'(i)$, $i = 1, \dots, |D'| - 1$) is computed, and

$$\min_{1 \leq i \leq |D'| - 1} \{D'(i+1) - D'(i)\} \quad (5)$$

is the discrete direction interval according to which discrete direction vector P is determined.

Step3: Each element in D is quantified to be a discrete direction in P and different elements in D correspond to distinct discrete directions.

Step4: Number of points falling into every discrete direction is accumulated as below:

$$H(i) = \sum_{j=1}^{|D|} l(j) \delta [D(j) - P(i)] \quad (6)$$

where

$$\delta[D(j) - P(i)] = \begin{cases} 1, & D(j) = P(i) \\ 0, & D(j) \neq P(i) \end{cases} \quad (7)$$

Step 5: Discrete direction vector P is the abscissa vector of EDH and $H(i)$ is the ordinate corresponding to $P(i)$; thereby EDH for the graph is obtained. Number of distinct edge directions may be different in graphs; therefore, the number of bins may be different in the corresponding histograms.

With EDHs on hand, distance of EDHs is computed to obtain GED. Bin-by-bin distances usually overestimate distance of histograms when there is no match between the exact corresponding bins. The cross-bin distances including the weighted L_2 distance, L_1 distance of the cumulative histograms and EMD are studied. They consider neighboring bins when there is no match between the exact corresponding bins in histograms, and accuracy of distance obtained by these methods is improved. But the weighted L_2

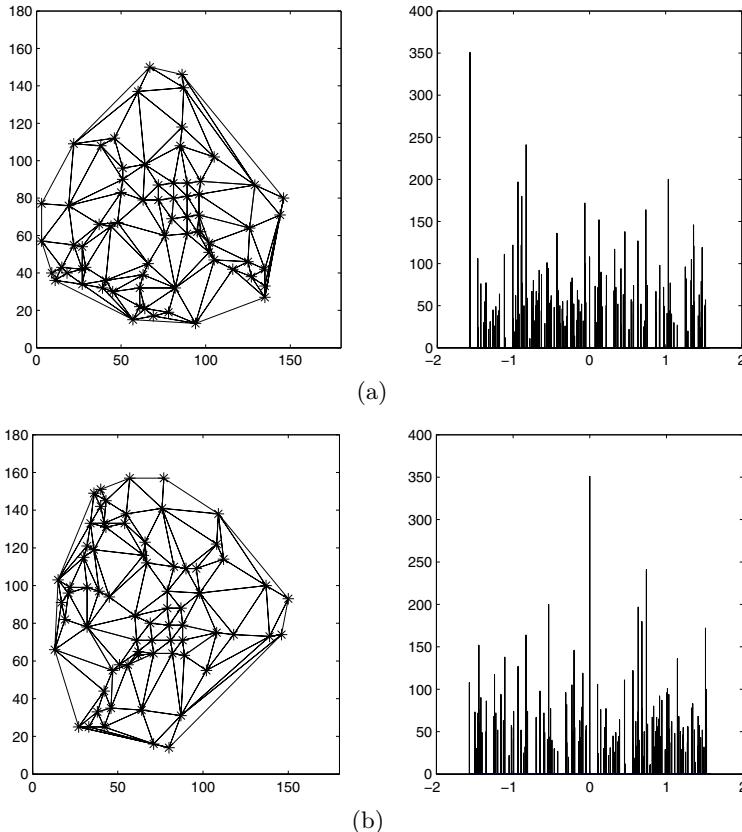


Fig. 3. Rotation of graphs and their corresponding EDHs. Graph in (b) is acquired by the clockwise rotation of graph in (a).

distance overestimates direction similarity without a pronounced mode and L_1 distance of the cumulative histograms needs increased computational complexity when non-empty bins in histogram are few. Besides those, they are sensitive to EDH difference caused by the graph rotation in the same plane. Two graphs shown in Fig. 3 are obtained with the orthogonal viewing angles so one of the graphs is the rotation of another one. Their weighted L_2 distance and L_1 distance of the cumulative histograms are much larger than zero, while their EMD is zero. These two graphs are judged to be the same according to EMD, although EDHs corresponding to these two graphs are very different intuitively. In addition, EMD considers neighboring bins appropriately and need not accumulative histogram so as to decrease the complexity. It is the more appropriate distance of EDHs in this paper.

EMD [20] for two histograms is defined as the minimal amount of work performed to transform a histogram into the other one, which is similar with the idea of minimal cost for transforming a graph into another one. For two histograms $X = \{(x_1, w_1), (x_2, w_2), \dots, (x_n, w_n)\}$ and $Y = \{(y_1, v_1), (y_2, v_2), \dots, (y_m, v_m)\}$, where (x_i, w_i) represents that the number of pixels at position x_i is w_i and the same to (y_j, v_j) , the EMD for transforming from histogram X into Y is the linear programming problem. The objective is searching for optimal F satisfying the function

$$\min_{F=\{f_{ij}\}} \left\{ \sum_{i=1}^n \sum_{j=1}^m d_{ij} f_{ij} \right\} \quad (8)$$

under the following constraints:

$$\begin{aligned} f_{ij} &\geq 0, \text{ where } 1 \leq i \leq n, 1 \leq j \leq m; \\ \sum_{j=1}^m f_{ij} &\leq w_i; \\ \sum_{i=1}^n f_{ij} &\leq v_j; \\ \sum_{i=1}^n \sum_{j=1}^m f_{ij} &= \min \left(\sum_{i=1}^n w_i, \sum_{j=1}^m v_j \right), \end{aligned}$$

where d_{ij} represents the cost of removing a point from i to j , and f_{ij} denotes the number of points removed from i to j . According to the optimal F , EMD is defined as:

$$EMD(X, Y) = \frac{\sum_{i=1}^n \sum_{j=1}^m d_{ij} f_{ij}}{\sum_{i=1}^n \sum_{j=1}^m f_{ij}}. \quad (9)$$

3.2 HMM-Based GED

Different from EDH-based method, all edit operations are characterized by node operations for correlation between edges and nodes; thus the GED is related to the node distribution which is modeled with HMM. The GED is computed by means of the KLD of HMMs whose procedure is shown in Fig. 4.

As the name implies, states of a HMM is non-observable and every state is depicted with hidden state variable s_i . Only a sequence of instances generated by these states can be observed, which is $O = \{o_1, o_2, \dots, o_T\}$ and T is length

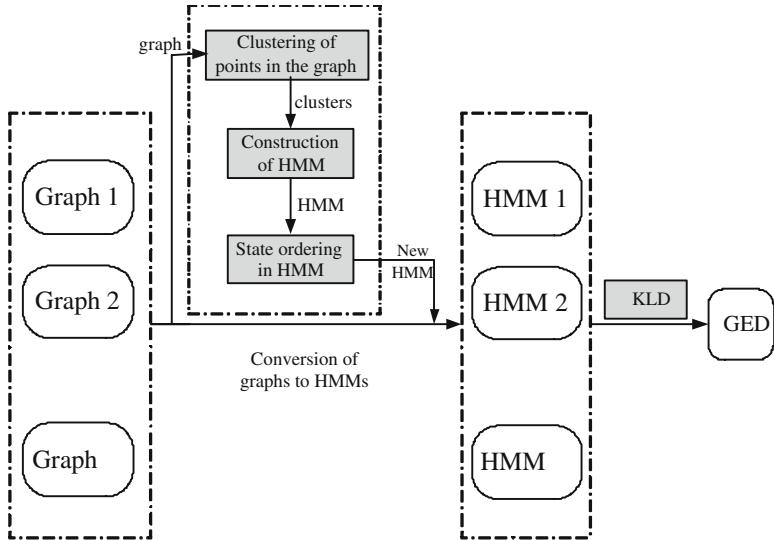


Fig. 4. The GED based on HMM. Graph i is converted into HMM i and GED is computed by the KLD of N HMMs.

of the observation sequence. A HMM is represented by $\lambda = (N, M, \Pi, A, B)$, where N is number of states in HMM; M is the number of distinct elements in observation sequence; $\Pi = \{\pi_i\}$, $i = 1, 2, \dots, N$, is probability distribution of original state space; $A = \{a_{ij}\}$, $i, j = 1, 2, \dots, N$, is the state transition probability matrix; $B = \{b_i(o_t)\}$, $i = 1, 2, \dots, N, t = 1, 2, \dots, T$, is the posterior probability matrix determining the observation when HMM is in some particular state; $V = \{v_1, v_2, \dots, v_M\}$ is the vector of distinct elements in observation sequence. Π, A, B have to be determined for a HMM while others are known quantities.

Position of points in a graph constitutes the observation sequence O , according to which the parameters Π, A, B are determined. As shown in Fig. 4, states of a HMM have to be determined by clustering points in the graph firstly. Points in each graph are grouped into several clusters for the purpose of constraining the affection in local area, which is caused by the tiny variation of some corresponding points in graphs, and each cluster corresponds to a state and a hidden state variable of the HMM. For the aim of clustering here, every point should belong to a unique cluster and distance of points in each cluster ought not be so great, which can be satisfied by K -means clustering algorithm [13]. K clusters of points in a graph are built with K -means algorithms:

Step1: K clustering centers are initialized.

Step2: Each point is assigned to the cluster whose center is the closest to it, compared with other cluster centers.

Step3: After all feature points are assigned into K clusters, K centers are recalculated.

Step4: *Step2* and *Step3* are repeated until the cluster centers remain invariable and K clusters are acquired. If there are some clusters containing only one point in each cluster, go to *Step5*, and otherwise go to *Step6*.

Step5: Some clusters containing only one point in each cluster are deleted and steps 1~4 are performed on the new data set.

Step6: Stop and output the clustering result.

Baum-Welch algorithm [18] is used to estimate parameters of the HMM and construct the HMM. Given an observation sequence $O = \{o_1, o_2, \dots, o_T\}$, we can determine the model λ to maximize $P(O|\lambda)$. The procedure is given as follows:

Step1: Parameters Π, A are initialized randomly. Marginal probability of each hidden state variable is the element in B and it is modeled with Gaussian mixture model (GMM) whose parameters is decided by a variation of expectation-maximization (EM) algorithm introduced in [7]. On the one hand, the number of components in GMM is set to be larger than the true number and it is determined automatically by removing the unnecessary components, which weakens the affection of initialization; on the other hand, components are annihilated when they are not supported by data in order to avoid the problem of parameter space boundary. This new GMM construction method is applied to the states corresponding to clusters of more than 40 points each. Clusters of less than 40 points each are modeled with Gaussian model (GM). So, elements in B are GMMs or GMs.

Step2: With the initialization above, these parameters are modified iteratively according to the observation sequence $O = \{o_1, o_2, \dots, o_T\}$ based on the idea of EM algorithm until $P(O|\lambda)$ is convergent and the HMM is updated. With the help of $\zeta_t(i, j)$ and $\zeta_t(i)$, adjustment of these parameters is explained.

$\zeta_t(i, j) = P(s_t = i, s_{t+1} = j | O, \lambda)$ is the probability of observed instance being in state j and the former observed instance belonging to state i when $O = \{o_1, o_2, \dots, o_T\}$ and λ are determined. By introducing forward variable $\alpha_t(i)$ and backward variable $\beta_{t+1}(j)$, we can deduce the formulation:

$$\zeta_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} \quad (10)$$

where $\alpha_t(i) = P(o_1, o_2, \dots, o_t, s_t = i | \lambda)$, $2 \leq t \leq T$ is the probability of partial observation sequence o_1, o_2, \dots, o_t and the state i corresponding to observation o_t given the model λ ; a_{ij} is the transition probability in λ ; $b_j(o_{t+1})$ is the probability of the observation value o_{t+1} belonging to the state j ; $\beta_{t+1}(j) = P(o_{t+2}, o_{t+3}, \dots, o_T | s_{t+1} = j, \lambda)$ where $1 \leq t \leq T - 1$, is the probability of partial observation sequence $o_{t+2}, o_{t+3}, \dots, o_T$ on the condition that the model λ exists and observation o_{t+1} is produced in the state j ,

$\beta_T(i) = 1$; $P(O|\lambda)$ is the probability of observation sequence given the model λ , that is, $P(O|\lambda) = \sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$.

$\zeta_t(i)$ is the probability of observation o_t belonging to state i and

$$\zeta_t(i) = \sum_{j=1}^N \zeta_t(i, j) = \frac{\alpha_t(i) \beta_t(i)}{P(O|\lambda)}. \quad (11)$$

The updated parameters of HMM are given as follows with two quantities defined in formulas (10) and (11):

$$\hat{\pi}_i = \zeta_1(i) \quad (12)$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \zeta_t(i, j)}{\sum_{t=1}^{T-1} \zeta_t(i)} \quad (13)$$

$$\hat{b}_j(k) = \frac{\sum_{t=1, o_t=v_k}^{T-1} \zeta_t(j)}{\sum_{t=1}^{T-1} \zeta_t(j)} \quad (14)$$

By these parameters determined in (12)-(14), we can acquire the updated HMM $\hat{\lambda} = (\hat{\Pi}, \hat{A}, \hat{B})$, where $\hat{\Pi} = \{\hat{\pi}_i\}$, $\hat{A} = \{\hat{a}_{ij}\}$ and $\hat{B} = \{\hat{b}_j(k)\}$, $i, j = 1, 2, \dots, N$.

With the construction of HMM for points in each graph, GED is converted into the distance of HMMs. Distance of statistical models is usually computed with KLD which is also known as relative entropy [17], and here, the KLD of HMM is computed with an approximate algorithm [10].

Provided two HMMs $\lambda = (\Pi, A, B)$ and $\tilde{\lambda} = (\tilde{\Pi}, \tilde{A}, \tilde{B})$ are derived from two images, where N is the number of states in each model, $\Pi = \{\pi_i\}$ and $\tilde{\Pi} = \{\tilde{\pi}_i\}$, $i = 1, 2, \dots, N$, are probability distributions of original state space, $A = \{a_{ij}\}$ and $\tilde{A} = \{\tilde{a}_{ij}\}$, $i, j = 1, 2, \dots, N$, are the state transition probability matrices, $B = \{b_i\}$ and $\tilde{B} = \{\tilde{b}_i\}$, $i = 1, 2, \dots, N$, are the posterior probability distributions of observation variables, the KLD of these two models is computed with the following formula:

$$D(\lambda \parallel \tilde{\lambda}) \leq D(\Pi \parallel \tilde{\Pi}) + \Pi^T \cdot \left(\sum_{n=1}^{N-1} A^{n-1} d + A^{N-1} D^{(N)} \right). \quad (15)$$

According to (15), we can only acquire the upper limit of distance between λ and $\tilde{\lambda}$ but rather the exact value.

Since Π and $\tilde{\Pi}$ are probability mass functions, their distance $D(\Pi \parallel \tilde{\Pi})$ is computed according to the KLD for probability mass functions, that is to say,

$$D(\Pi \parallel \tilde{\Pi}) = \sum_{i=1}^N \pi_i \times \log \frac{\pi_i}{\tilde{\pi}_i}. \quad (16)$$

d is set to be a vector $(d_1, d_2, \dots, d_N)^T$, where $d_i = D(a_i \parallel \tilde{a}_i) + D(b_i \parallel \tilde{b}_i)$, $a_i = A(i, :)$, $\tilde{a}_i = \tilde{A}(i, :)$. a_i, \tilde{a}_i are probability mass functions too, so $D(a_i \parallel \tilde{a}_i)$ can be computed according to the formula similar to (16). $b_i = \{b_i^j\}$ and $\tilde{b}_i = \{\tilde{b}_i^k\}$ are GMMs; b_i has n_1 components with weight vector $w = \{w_j\}$, $j = 1, 2, \dots, n_1$, and \tilde{b}_i has n_2 components with weight vector $\tilde{w} = \{\tilde{w}_k\}$, $k = 1, 2, \dots, n_2$. KLD for these two probability functions is set to be the mean value of distances from b_i to \tilde{b}_i and from \tilde{b}_i to b_i , and the result is

$$D(b_i \parallel \tilde{b}_i) = \frac{1}{2} \left[\sum_{j=1}^{n_1} w_j \times \min_{k=1}^{n_2} D' \left(b_i^j, \tilde{b}_i^k \right) + \sum_{k=1}^{n_2} \tilde{w}_k \times \min_{j=1}^{n_1} D' \left(\tilde{b}_i^k, b_i^j \right) \right]. \quad (17)$$

For each pair of corresponding states of two HMMs, we compute the distance of their GMMs and construct the vector

$$D^{(N)} = \left(D(b_1 \parallel \tilde{b}_1), D(b_2 \parallel \tilde{b}_2), \dots, D(b_N \parallel \tilde{b}_N) \right)^T.$$

KLD of pairwise HMMs is computed according to the information of their corresponding states; thus it is affected by the correspondence of states in these two models. Considering points in an area of similar graphs should belong to corresponding states, therefore, states in each HMM should be ordered before the KLD is computed. For an image, every state (i.e. GMM) has one or more GMs and it has one or more mean points, among which the mean point with minimal column index is chosen. Then the chosen mean value points of all states are ordered with their column indexes, according to which the states are ordered.

4 Theoretical Comparison of Three Methods

4.1 Complexity Analysis

With the flow charts of these above three GED algorithms, complexity analysis of every algorithm is conducted step by step and is summarized in Table II

In the GED from spectral seriation, a graph with n points is converted into a string with spectral seriation of the computational complexity $O(n \times \log n) - O(n^3)$. When the graph is a complete graph, this step is of the least complexity $O(n \times \log n)$. For two graphs having n points and m points respectively, the complexity of Levenshtein distance in probability is $O(n \times m^2)$, $n \geq m$.

In the GED based on EDH, size of each graph is normalized and each graph is converted into an EDH, which are performed on every edge in the graph. The graph is usually obtained by Delaunay triangulation and it conforms to $e \leq 3 \times n - 6$, where n is number of points and e is the number of edges, so normalization of a graph and conversion of a graph into an EDH are of $O(n)$.

Table 1. Complexity of Three GED Methods

Method	Step	Complexity
GED from spectral seriation method	Conversion of a graph with n points into a string	$O(n \times \log n) - O(n^3)$
	Levenshtein distance in probabilistic framework for two graphs with n points and m points respectively	$O(n \times m^2)$, $n \geq m$
EDH-based GED	Normalization of a graph with n points	$O(n)$
	Conversion of a graph with n points into an EDH	$O(n)$
HMM-based GED	EMD	None theoretical analysis
	Clustering of n points in a graph	$O(n)$
HMM-based GED	Construction of HMM	$O(n)$
	Ordering of states in a HMM	Constant
	KLD	Constant

The EMD is based on the transportation simplex method, whose theoretical analysis of the computational complexity is hard. But EMD can be computed efficiently.

The computational complexity of the HMM-based GED deserves our full attention. n points in a graph are clustered into three classes by K -means clustering algorithm, and the computational complexity is $O(n \times K \times t)$, where $K = 3$ is the number of clusters, t is the number of iterations and is less than 6 in the experiment; therefore, K and t are constants and the complexity of K -mean clustering algorithm is $O(n)$. After clustering corner points in an image, we construct GMMs for clusters and form a HMM. Both of these two steps are based on EM algorithm and their complexity is $O(L \times n \times k)$, where L is the number of iterations before convergence, $1 \leq k \leq 5$ is the number of components in each cluster for constructing a GMM and $k = 3$ is the number of states (i.e. the number of clusters in a graph) for constructing a HMM based on GMMs; therefore L and k can be considered as constants and a HMM can be constructed with the complexity $O(n)$. When the states in a HMM are ordered, complexity of ordering algorithm is $O(N^2) - O(N \times \log n)$, where N is the number of data to be ordered and N (that is $N = 3$) is the number of states in the HMM-based algorithm, so complexity of state ordering can be considered as a constant. KLD for HMMs is evaluated directly on the model parameters and its cost is a constant. In a word, computational complexity of the HMM-based GED algorithm is $O(n)$.

Because the theoretical analysis of EMD is nearly impossible, time-performance for these three kinds of distances: Levenshtein distance in probabilistic framework, EMD and KLD, is measured experimentally. Both of the methods use the same set of data, every graph is converted into a signal which is a string, an EDH or a HMM. The time consumed by these three

Table 2. Time Consumed by Levenshtein Distance in Probability Framework

	1	2	3	4	5
1	0.6482	0.646	0.6453	0.6436	0.6805
2	0.6409	0.6449	0.6541	0.6246	0.653
3	0.6378	0.6329	0.6279	0.6191	0.6602
4	0.6421	0.6218	0.6182	0.6196	0.6626
5	0.6802	0.6536	0.6607	0.6512	0.6743

Table 3. Time Consumed by EMD

	1	2	3	4	5
1	0.0187	0.0189	0.0172	0.0202	0.0181
2	0.0188	0.0181	0.0167	0.0182	0.019
3	0.0174	0.0171	0.0223	0.0197	0.0175
4	0.0178	0.0161	0.0219	0.0273	0.0182
5	0.0203	0.0194	0.0155	0.018	0.0303

Table 4. Time Consumed by KLD

	1	2	3	4	5
1	0.0039	0.0024	0.0025	0.0130	0.0134
2	0.0188	0.0067	0.0170	0.0206	0.0033
3	0.0024	0.0024	0.0025	0.0027	0.0068
4	0.0025	0.0024	0.0025	0.0024	0.0024
5	0.0024	0.0024	0.0024	0.0024	0.0107

distances is compared and shown in Tables 2–4. In these three tables, graphs having the same index are the same. KLD is the most efficient and Levenshtein distance in probability framework takes up the most time. Conclusion in Table II, together with the comparison of Tables 2–4, indicates that GED from spectral seriation method has the highest complexity and HMM-based method has the highest efficiency.

4.2 Sensitivity Analysis to Graph Structure Variety

In this subsection, the aim is to illustrate whether these three methods characterize both the difference of points and the distance between pairwise points (i.e. length of edges) effectively. In GED from spectral seriation, the edge compatibility coefficient $R_{k,k+1} = P(\gamma_k, \gamma_{k+1})/P(\gamma_k)P(\gamma_{k+1})$ is modeled with node attendance and characterizes the difference of points in different graphs. $P(\gamma_k|\phi_X^*(x_j), \phi_Y^*(y_i))$ is used to model the errors distribution in the leading eigenvector of the graph adjacency matrix and a string is derived using the leading eigenvector, so it characterizes the difference of strings. There is

nothing characterizing the distances of pairwise points. In the EDH-based GED, EDH is computed according to the slope and length of edges in a graph. Although the size of graphs is normalized, the relative difference of edge length remains. The difference of feature points and the distance of pairwise points (i.e. length of edges) are both characterized. As regards to HMM-based method, HMM is used to model the distribution of points in graphs and the distribution difference of points leads to different HMMs for these graphs. If number and adjacency of points in graphs are the same and only the length of some edges is distinct, there are two cases:

Case1: the difference of edge length does not affect the clustering of points. The relative-position of points within each cluster remains invariable, but the mean values of some clusters are different, resulting in different HMMs.

Case2: the difference of edge length affects the clustering of points. The distribution of points in some corresponding clusters is dissimilar, resulting in different HMMs.

So, the difference of points and the distance of pairwise points are both characterized in the HMM-based method.

5 Experimental Results

These three algorithms are evaluated on dataset of three sequences [16], [23]: CMU-VASC sequence, INRIA MOVI sequence, Swiss chalet sequence. Instances of the experimental images are shown in Fig. 5. Three panels from the leftmost to the rightmost show the instances of CMU-VASC sequence, INRIA MOVI sequence and Swiss chalet sequence. Ten images in each sequence constitute a class, which are arranged according to the viewing angle. Graphs for these images are derived by extracting corner points with Harris corner detector and determining connectivity of these points with Delaunay triangulation. Experiments are performed in Matlab environment of PC with Intel Core 2 CPU 6300 and Memory 1.99GB.

First, the distance matrices are computed with these three methods in order to measure similarity between pairwise images. Each element of matrix



Fig. 5. Instances of three image sequences

Table 5. Comparison of Distance Matrices

Method	Errors	Execution time (with GED from spectral seriation as reference)
GED from spectral seriation	Two images	100%
EDH-based GED	Two images	9.3%
HMM-based GED	One image	0.64%

Table 6. Mean Value of GED

Method	Mean value of within-class distance	Mean value of between-class distance
GED from spectral seriation	6.77	45.87
EDH-based GED	13.24	40.27
HMM-based GED	4.57	31.18

specifies the color of a rectangular patch so that the matrices are represented by images shown in Fig. 6. The darker the patch, the smaller the value in the matrix. 1–10 correspond to the first class, 11–20 the second class and 21–30 the third class, so blocks along the diagonal present within-class distance and other blocks present between-class distance.

Image 26 and image 27 are closer to the first class images in Fig. 6(a) and there are errors of two images in the third class, i.e. image 28 and image 30, in Fig. 6(b), but 712.26 seconds are consumed by the method from spectral seriaton while 65.91 seconds by EDH-based method. In Fig. 6(c), only image 21 is closer to the first class images and it only needs 4.55 seconds. The conclusion is listed in Table 5. Distance matrix of HMM-based method is of the least-error and the least-time.

The mean values of within-class distance and between-class distance are computed for these three methods in order to measure the ability of distinguishing images in different sets quantitatively. As shown in Table 6, mean value of within-class distance is much smaller than that of between-class distance in these three GED matrices, which coincides with Fig. 6. Although the difference for sepectral seriation method is the most, differences between within-class distance and between-class distance for other two methods are great enough to distinguish three classes effectively. When the efficiency is considered, EDH-based and HMM-based methods are better.

K -nearest neighbor (K -NN) classification [15] is an approach of nonparametric classification and can be implemented more easily and efficiently. High classification rate can be achieved for the samples whose distribution is unknown or nonnormal. The distribution of experimental data is unknown; therefore, K -NN is available for distinguishing the performance of GED methods here.

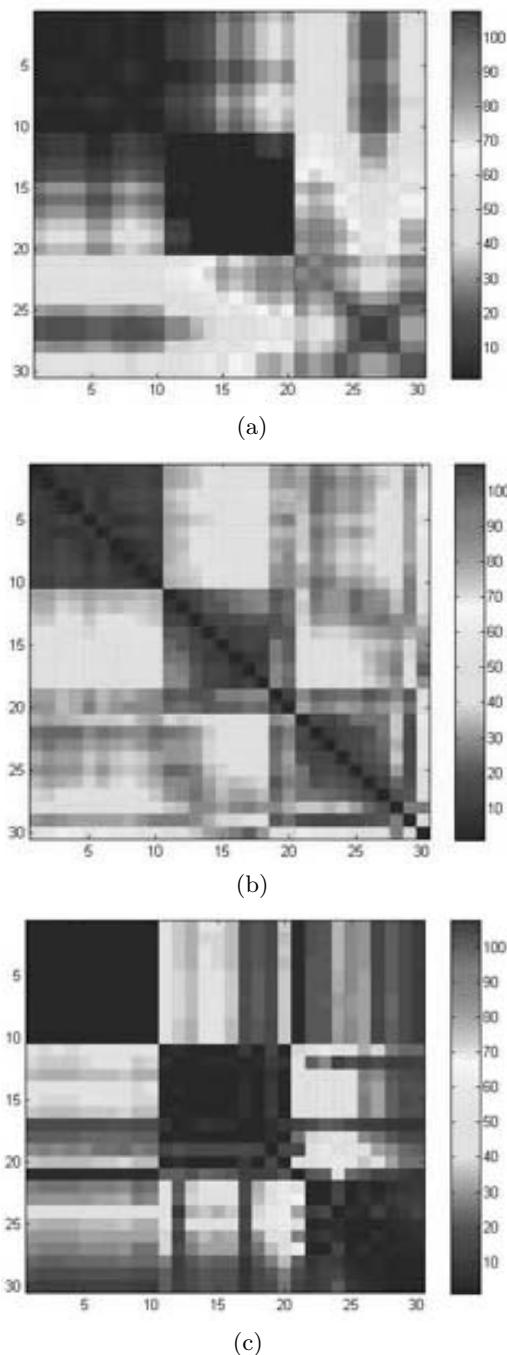


Fig. 6. Distance matrices: (a) GED from spectral seriation; (b) GED based on EDH; (c) GED based on HMM

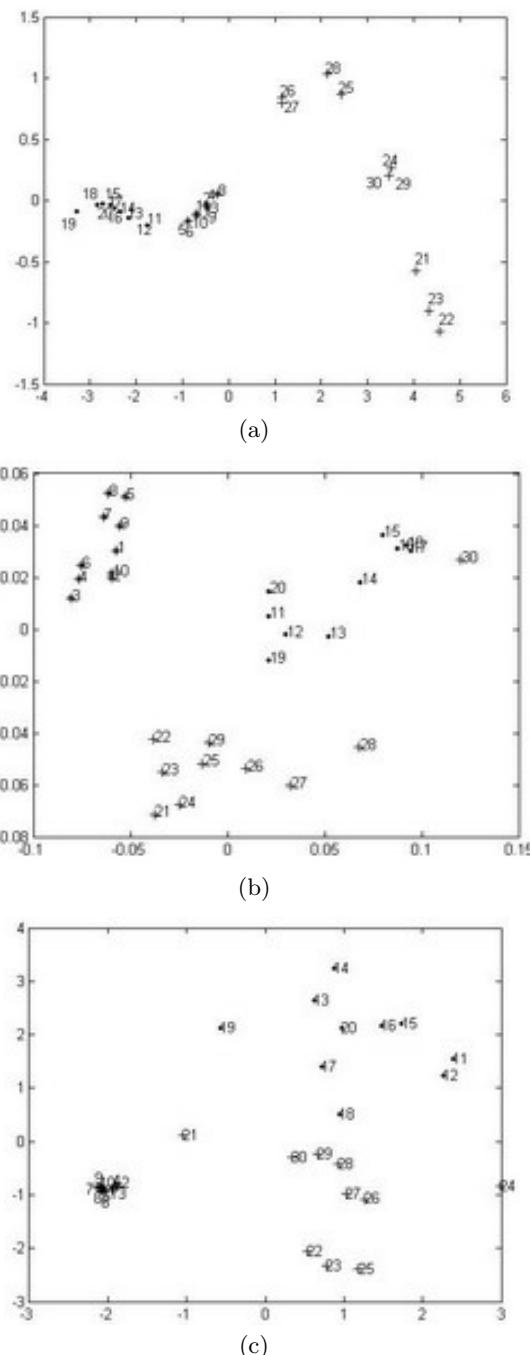


Fig. 7. MDS for each of distance matrices: (a) MDS of GED from spectral seriation; (b) MDS of GED based on EDH; (c) MDS of GED based on HMM

Table 7. Confusion Matrices of the GED

Class of images	GED from spectral seriation			EDH based GED			HMM based GED		
	C1	C2	C3	C1	C2	C3	C1	C2	C3
1	10	0	10	10	0	0	10	0	0
2	0	10	0	0	10	0	0	10	0
3	2	0	8	0	2	8	1	0	9
Classification rate	93.3%			93.3%			96.7%		
Execution time(s)	683.54s			30.1s			22.6s		

K is the number of the nearest neighbors and is set to be 9. The classification results of these three GED algorithms are represented by confusion matrices shown in Table 7. The rows are indexed by the true class index of images (i.e. 1, 2, 3) and columns are indexed by the predicted class index obtained with K -NN classification (i.e. C1, C2, C3). Element (i, j) in the matrix represents the number of images in i th class being classified into the j th class, and the diagonal elements represents the number of correctly classified images and others represent the number of the misclassified images.

For the GED from spectral seriation and EDH-based GED, there are two images being misclassified and correct classification rate of these two methods is the same. But execution time of the former method is 683.54 seconds, which is in contrast to 30.1 seconds for the later one. EDH-based method consumes 4.4% the time consumed by the spectral seriation method. GED based on HMM misclassifies only one image with 22.6 seconds which is 3.3% that of the spectral seriation method, leading to the highest classification rate most efficiently.

The multidimensional scaling (MDS) [8] is a kind of factor analysis virtually. It is usually used to detect meaningful underlying dimensions so that we can explain observed similarities or dissimilarities between images and can get a visual representation of high dimensional images. MDS is implemented with three GED matrices obtained above as distance measures in order to examine whether they can capture the potential dissimilarity of different image sets. We show the MDS results in Fig. 7 and index of each image is given near its corresponding node in each graph.

Three classes of images can be clustered clearly by three distance measurements, but the distribution of the third class is too sparse in comparison with the other two classes in Fig. 7(a), and image 30 is clustered wrongly into the second class in Fig. 7(b), and results of the first two methods have their own disadvantages respectively. When the efficiency is taken into account, 747.64 seconds is occupied by the spectral seriation method and 40.61 seconds for the EDH-based method. In Fig. 7(c), the distribution of the each class is compact appropriately and HMM-based method only consumed 23.73 seconds, the least execution time. So, HMM-based method corresponds to the best performance.

6 Conclusions

The work in this paper proposes and studies two cost function-free GED algorithms theoretically and experimentally. In the EDH-based method, edit operations are involved in graph structure difference characterized by EDH and GED is converted into EMD of EDHs, while edit operations are involved in node distribution difference characterized by HMM and GED is converted into KLD of HMMs in the HMM-based method. These two methods are completely independent on cost function definition. By theoretical and experimental comparison between them and a cost function-dependent GED algorithm, that is GED from spectral seriation, these two cost function-free algorithms have higher correct rate of clustering and classification for images, not lower at least, and are far superior to spectral seriation method, especially HMM-based method, with regard to efficiency. With respect to two cost function free algorithms, HMM-based method excels EDH-based method in classification and clustering rate, and efficiency.

Cost function-free algorithms is a novel kind of approach to compute GED, future research may extend to multi-scale EDH and overcoming the randomicity of the HMM's generation.

Acknowledgments

This work was supported in part by the Program for Changjiang Scholars and Innovative Research Team in University of China (IRT0645), the National Natural Science Foundation of China (No. 60771068, No.60702061), the Open-End Fund of National Laboratory of Pattern Recognition in China, the Open-End Fund of National Laboratory of Automatic Target Recognition (ATR), Shenzhen University, China.

References

1. Bunke, H.: Pattern Recognition Letters 18(8), 689–694 (1997)
2. Bunke, H.: IEEE Trans. Pattern Anal. Mach. Intell. 21(9), 917–922 (1999)
3. Bunke, H.: Proc. IEEE Int'l Conf. on Pattern Recognition, 117–124 (2000)
4. Caelli, T., Kosinov, S.: IEEE Trans. Pattern Anal. Mach. Intell. 26(4), 515–519 (2004)
5. Chen, G.H., Yang, C.L., Po, L.M., Xie, S.L.: Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing, pp. 933–936 (2006)
6. Cross, A.D.J., Wilson, R.C., Hancock, E.R.: Pattern Recognition 30(7), 953–970 (1997)
7. Figueiredo, M.A.T., Jain, A.K.: IEEE Trans. Pattern Anal. Mach. Intell. 24(3), 381–396 (2002)
8. Hofmann, T., Buhmann, J.M.: Proc. Conf. Neural information processing systems, pp. 459–466 (1994)
9. Jain, A., Dubes, R.: Algorithms for clustering data. Prentice-Hall, Englewood Cliffs (1988)

10. Kim, D.H., Yun, I.D., Lee, S.U.: Proc. the 17th Int'l Conf. Pattern Recognition, pp. 48–51 (2004)
11. Kim, Y.W., Oh, I.S.: Pattern Recognition Letters 25(11), 1243–1251 (2004)
12. Levenshtein, V.: Soviet Physics-Doklady 10(8), 707–710 (1966)
13. MacQueen, J.B.: Proc. 5-th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
14. Mahmoudi, F., Shanbehzadeh, J.: Pattern Recognition 36(8), 1725–1736 (2003)
15. Mitchell, T.M.: Machine Learning. UK McGraw Hill, Maidenhead (1997)
16. INRIA-MOVI houses,
<http://www.inria.fr/recherche/equipes/movi.en.html>
17. Qian, H.: Physical Review E 63, 042103/1–042103/4 (2001)
18. Rabiner, L.R.: Proc. IEEE 77(2), 257–286 (1989)
19. Robles-Kelly, A., Hancock, E.R.: IEEE Trans. Pattern Anal. Mach. Intell. 27(3), 365–378 (2005)
20. Rubner, Y., Tomasi, C., Guibas, L.J.: Proc. IEEE Int'l Conf. Computer Vision, 59–66 (1998)
21. Umeyama, S.: IEEE Trans. Pattern Anal. Mach. Intell. 10(5), 695–703 (1988)
22. Vailaya, A., Jain, A., Zhang, H.J.: Proc. Workshop in Content-Based Access to Image and Video Libraries, pp. 3–8 (1998)
23. Vision and Autonomous Systems Center's Image Database,
<http://vasc.ri.cmu.edu//idb/html/motion/house/index.html>
24. Wagner, R.A., Fischer, M.J.: Journal of the ACM 21(1), 168–173 (1974)
25. Zhang, L., Allinson, N.M.: Proc. 5th Annual UK Workshop on Comp. Intelligence, pp. 137–142 (2005)
26. Xiao, B., Gao, X.B., Tao, D.C., Li, X.L.: International Journal of Imaging Systems and Technology 18(2-3), 209–218 (2008)
27. Gao, X.B., Xiao, B., Tao, D.C., Li, X.L.: Pattern Recognition 47(10), 3179–3191 (2008)

Classification of Complex Molecules

Francisco Torrens^{1,*} and Gloria Castellano²

¹ Institut Universitari de Ciència Molecular, Universitat de València,
Edifici d'Instituts de Paterna, P.O. Box 22085, E-46071 València, Spain

² Instituto Universitario de Medio Ambiente y Ciencias Marinas,
Universidad Católica de Valencia *San Vicente Mártir*,
Guillem de Castro-94, E-46003 València, Spain

Abstract. Algorithms for classification and taxonomy based on criteria, *e.g.*, *information entropy* and its production are proposed. In molecular classification, the feasibility of replacing a given molecule (*e.g.*, anaesthetic) by similar ones in the composition of a complex drug is studied. Some local anaesthetics currently in use are classified using characteristic chemical properties of different portions of their molecules. In taxonomy, the detailed comparison of the sequences (primary structures) of biomolecules, proteins or nucleic acids, allows the reconstruction of a molecular phylogenetic tree for some species, *e.g.* the 1918 influenza virus. The method is applied to the classifications of: (1) indazolols (action against *Trichomonas vaginalis*), (2) fullerenes, fullerite and single-wall carbon nanotubes, (3) living and heat-inactivated lactic acid bacteria against cytokines, (4) phylogenesis of avian birds and of the 1918 influenza virus, (5) local anaesthetics (analogues of procaine), (6) transdermal-delivery percutaneous enhancers, (7) quantitative structure–activity relationship modelling of anti-human immunodeficiency virus type 1 (HIV-1) compounds and (8) HIV-1 inhibitors. If, in the calculation of entropy associated with the phylogenetic tree, a species is systematically omitted, the difference between the entropy with and without this species can be considered as a measure of the species entropy. Such contributions may be studied with the *equipartition conjecture*. Obviously, it is not within the scope of our simulation method to replace biological tests of drugs or field data in palaeontology, but such simulation methods can be useful to assert priorities in detailed experimental research. Available experimental and field data should be examined by different classification algorithms to reveal possible features of real biological significance.

1 Introduction

Ab initio theoretical calculations, molecular dynamics simulations and docking studies are useful tools for investigating important biological complexes [1–3]. At least three anti-human immunodeficiency virus type 1 (HIV-1) drugs, for combination therapy, became the standard treatment of acquired immunodeficiency

* Corresponding author.

syndrome (AIDS) drugs that have been licensed for clinical use, or are subjected to advanced clinical trials, belong to one of three classes: (1) nucleoside/nucleotide reverse transcriptase inhibitors (NRTIs/NtRTIs) {abacavir (ABC), emtricitabine [(-)FTC], zidovudine (AZT), didanosine (ddI), zalcitabine (ddC), stavudine (d4T), lamivudine (3TC) and tenofovir disoproxil fumarate}, (2) non-nucleoside reverse transcriptase inhibitors (NNRTIs) (emivirine, efavirenz, nevirapine and delavirdine) and (3) protease inhibitors (PIs) (lopinavir, nelfinavir, ritonavir, amprenavir, saquinavir and indinavir) [4–6]. Various other events in the HIV replicative cycle can be considered as potential targets for chemotherapeutic intervention: (1) viral entry *via* blockade of viral coreceptors CXCR4 [bicyclam (AMD3100) derivatives] and CCR5 (TAK-799 derivatives), (2) viral adsorption *via* binding to viral envelope glycoprotein gp120 (polysulphates, polysulphonates, polycarboxylates, polyoxometalates, polynucleotides and negatively charged albumins), (3) viral assembly and disassembly *via* NCp7 Zn finger-targeted agents [2,2'-dithiobisbenzamides (DIBAs), azodicarbonamide (ADA)], (4) virus-cell fusion *via* binding to viral envelope glycoprotein gp41 (T-1249), (5) proviral deoxyribonucleic acid (DNA) integration *via* integrase inhibitors (4-aryl-2,4-dioxobutanoic acid derivatives) and (6) viral messenger ribonucleic acid (mRNA) transcription *via* inhibitors of transcription (transactivation) process (flavopiridol, fluoroquinolones) [7–9]. New NRTIs, NNRTIs and PIs were developed that possess, respectively: (1) improved metabolic characteristics (phosphoramidate and cyclosaligenyl pronucleotides bypassing the first phosphorylation step of NRTIs), (2) increased activity [*second* or *third* generation NNRTIs (TMC-125, DPC-093)] and (3) different, non-peptidic scaffold [cyclic urea (mozenvir), 4-hydroxy-2-pyrone (tripanavir)] [10–12].

The advent of so many new compounds, other than those that have been formally approved for the treatment of HIV infections, will undoubtedly improve the prognosis of patients with AIDS and AIDS-associated diseases. Nucleoside analogues constitute a family of biological molecules (ddI, d4T, ddC and 3TC), which play an important role in the transcription process of HIV. The normal nucleoside substrates, used by reverse transcriptase (RT) to synthesize DNA, are mimicked by these nucleoside analogues, which lacked a 3'-OH group and, consequently, act as chain terminators when incorporated into DNA by RT. Although these nucleoside analogues show good activity as inhibitors of HIV, their long-term usefulness is limited by toxicities. Resistance and mutation are also problems. The development of better drugs requires a better understanding of how the drugs work, the mechanism of drug resistance and interaction with receptor, and stability of the drugs inside active site. A HIV RT inhibitor ligand was proposed, which indicated highest docking scores and more hydrogen-bond interactions with the residues of RT active site [13].

A simple computerized algorithm, useful for establishing a relationship between chemical structures and their biological activities or significance, is proposed and exemplified [14, 15]. The starting point is to use an informational or configurational entropy for pattern recognition purposes. As entropy is weakly discriminating for classification purposes, the more powerful concept of *entropy production* and its *equipartition conjecture* are introduced [16]. In earlier publications, the periodic classifications of local anaesthetics [17–19] and HIV

inhibitors [20] were analyzed. The aim of the present report is to develop the learning potentialities of the code and, since molecules are more naturally described *via* a varying size structured representation, the study of general approaches to the processing of structured information. Section 2 presents computational method. Section 3 describes classification algorithm. Section 4 exposes the equipartition conjecture of entropy production. Section 5 analyzes learning procedure. Section 6 classifies indazolols against *Trichomonas vaginalis*. Section 7 classifies fullerenes, fullerite and single-wall carbon nanotubes. Section 8 classifies lactic acid bacteria by cytokine immunomodulation. Section 9 analyzes the phylogenesis of avian birds and 1918 influenza virus. Section 10 reviews the classification of local anaesthetics. Section 11 classifies transdermal-delivery percutaneous enhancers. Section 12 reviews QSAR modelling of anti-HIV compounds. Section 13 analyzes the phylogenesis of vertebrates, mammals and monkeys. Section 14 examines the phylogenesis of apes, hominids and man. Section 15 studies the phylogenesis of extinct species. Section 16 analyzes the classification of HIV inhibitors. Section 17 reports perspectives.

2 Computational Method

The key problem in classification studies is to define *similarity indices*, when several criteria of comparison are involved. The first step in quantifying the concept of similarity, for molecules of HIV-1 inhibitors, is to list the most important portions of such molecules. Furthermore, the *vector of properties* $i = \langle i_1, i_2, \dots, i_k, \dots \rangle$ should be associated with each inhibitor i , whose components correspond to different characteristic groups in the inhibitor molecule, in a hierarchical order according to the expected importance of their pharmacological potency. If the m -th portion of the molecule is pharmacologically more significant for the inhibitory effect than the k -th portion, then $m < k$. The components i_k are "1" or "0", according to whether a similar (or identical) portion of rank k is present or absent in inhibitor i , compared with the reference inhibitor. It is assumed that the *structural elements* of an inhibitor molecule can be *ranked*, according to their contribution to inhibitory activity, in the following order of decreasing importance: number of N atoms > number of O atoms > number of S atoms > number of P atoms > number of halogen atoms. The ddI molecule contains four N, three O, no S, no P and no halogen (X = F, Cl, Br) heteroatoms ($\text{N}_4\text{O}_3\text{S}_0\text{P}_0\text{X}_0$). Most inhibitors contain no S heteroatom (ddI, ddC, d4T, novel proposed ligand, $\text{N}_{3-4}\text{O}_3\text{S}_0\text{P}_0\text{X}_0$), while 3TC includes one S heteroatom ($\text{N}_3\text{O}_3\text{S}_1\text{P}_0\text{X}_0$). In the NRTI inhibitor ddI the molecule contains four N, three O, no S, no P and no halogen ($\text{N}_4\text{O}_3\text{S}_0\text{P}_0\text{X}_0$, cf. Fig. 1). Obviously its associated vector is $\langle 11111 \rangle$. In this study, ddI was selected as a *reference* HIV-1 inhibitor, because of the good docking scores with receptor RT. This improves the quality of classification for those inhibitors similar to ddI.

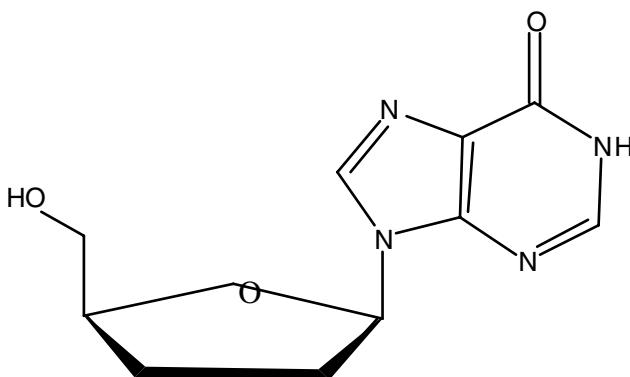


Fig. 1. Molecular structure of a HIV-1 NRTI inhibitor didanosine (ddI) molecule

Table 1. Vector properties of human immunodeficiency virus type 1 inhibitors

Non-nucleoside reverse transcriptase inhibitors (NNRTIs)			
1. efavirenz	<00110>	7. R165335 (TMC125)	<00110>
2. nevirapine	<10111>	8. SJ-3366	<01111>
3. delavirdine	<00111>	9. capravirine (AG1549)	<10010>
4. emivirine (MKC-442)	<01111>	10. PNU-142721	<10010>
5. thiocarboxanilide UC-781	<00010>	11. (+)-calanolide A	<00111>
6. DPC 083	<00110>		
Nucleoside reverse transcriptase inhibitors (NRTIs)			
12. didanosine (ddI)	<11111>	17. zidovudine (AZT)	<00111>
13. zalcitabine (ddC)	<01111>	18. abacavir (ABC)	<00111>
14. stavudine (d4T)	<01111>	19. emtricitabine [(-)-FTC]	<01010>
15. lamivudine (3TC)	<01011>	20. amdoxovir (DAPD)	<01111>
16. novel proposed ligand	<11111>	21. (±)-2'-deoxy-3'-oxa-4'-thiocytidine (dOTC)	<01011>
Nucleotide reverse transcriptase inhibitors (NtRTIs)			
22. adefovir dipivoxyl	<00101>	25. d4T aryloxyphosphoramidate	<00101>
23. tenofovir disoproxil	<00101>	26. cyclosaligenyl d4TMP	<00101>
24. bis(S-acetyl-2-thioethyl)phosphotriester of 2',3'-dideoxyadenosine [bis(SATE)ddAMP]	<00001>		
Protease inhibitors (PIs)			
27. amprenavir	<00011>	30. mozenavir (DMP-450)	<11111>
28. lopinavir	<10111>	31. tipranavir (PNU-140690)	<00010>
29. atazanavir (BMS-232632)	<00111>		

Table 1 contains the vectors associated with 31 HIV-1 inhibitors of various types: NNRTIs, NRTIs, NtRTIs and PIs. Vector <00110> is associated with efavirenz since the molecule contains one N, two O, no S, no P and four halogens. Vector <10111> is associated with nevirapine since there are four N, one O, no S,

no P and no halogen. Let us denote by r_{ij} ($0 \leq r_{ij} \leq 1$) the similarity index of two inhibitors associated with the i and j vectors, respectively. The similitude relation is characterized by a *similarity matrix* $\mathbf{R} = [r_{ij}]$. The similarity index between two inhibitors $i = < i_1, i_2, \dots, i_k \dots >$ and $j = < j_1, j_2, \dots, j_k \dots >$ is defined as:

$$r_{ij} = \sum_k t_k (a_k)^k \quad (k=1,2,\dots) \quad (1)$$

where $0 \leq a_k \leq 1$ and $t_k = 1$ if $i_k = j_k$, but $t_k = 0$ if $i_k \neq j_k$. The definition assigns a weight $(a_k)^k$ to any property involved in the description of molecule i or j .

3 Classification Algorithm

The *grouping algorithm* uses the *stabilized* matrix of similarity, obtained by applying the *max-min composition rule* o defined by:

$$(\mathbf{RoS})_{ij} = \max_k \left[\min_k (r_{ik}, s_{kj}) \right] \quad (2)$$

where $\mathbf{R} = [r_{ij}]$ and $\mathbf{S} = [s_{ij}]$ are matrices of the same type, and $(\mathbf{RoS})_{ij}$ is the (i,j) -th element of the matrix \mathbf{RoS} [21]. It can be shown that when applying this rule iteratively so that $\mathbf{R}(n+1) = \mathbf{R}(n) o \mathbf{R}$, there exists an integer n such that: $\mathbf{R}(n) = \mathbf{R}(n+1) = \dots$. The resulting matrix $\mathbf{R}(n)$ is called the *stabilized similarity matrix*. The importance of stabilization lies in the fact that in the classification process, it will generate a partition into disjoint classes. From now on it is understood that the stabilized matrix is used and designated by $\mathbf{R}(n) = [r_{ij}(n)]$. The *grouping rule* is the following: i and j are assigned to the same class if $r_{ij}(n) \geq b$. The class of i noted \hat{i} is the set of species j that satisfies the rule $r_{ij}(n) \geq b$. The matrix of classes is:

$$\hat{\mathbf{R}}(n) = \left[\hat{r}_{ij} \right] = \max_{s,t} (r_{st}) \quad (s \in \hat{i}, t \in \hat{j}) \quad (3)$$

where s stands for any index of a species belonging to the class \hat{i} (similarly for t and \hat{j}). Rule (3) means finding the largest similarity index between species of two different classes. In information theory, the *information entropy* h measures the surprise that the source emitting the sequences can give [22, 23]. For a single event occurring with probability p , the degree of surprise is proportional to $-\ln p$. Generalizing the result to a random variable X (which can take N possible values x_1, \dots, x_N with probabilities p_1, \dots, p_N), the average surprise received on learning the value of X is $-\sum p_i \ln p_i$. The information entropy associated with the matrix of similarity \mathbf{R} is:

$$h(\mathbf{R}) = - \sum_{i,j} r_{ij} \ln r_{ij} - \sum_{i,j} (1 - r_{ij}) \ln (1 - r_{ij}) \quad (4)$$

Denote also by C_b the set of classes and by $\hat{\mathbf{R}}_b$ the matrix of similarity at the grouping level b . The information entropy satisfies the following properties. (1) $h(\mathbf{R}) = 0$ if $r_{ij} = 0$ or $r_{ij} = 1$. (2) $h(\mathbf{R})$ is maximum if $r_{ij} = 0.5$. (3) $h(\hat{\mathbf{R}}_b) \leq h(\mathbf{R})$ for any b . (4) $h(\hat{\mathbf{R}}_{b_1}) \leq h(\hat{\mathbf{R}}_{b_2})$ if $b_1 < b_2$.

4 The Equipartition Conjecture of Entropy Production

In the classification algorithm, each *hierarchical tree* corresponds to a dependence of entropy on the grouping level, and thus an h - b diagram can be obtained. The Tondeur and Kvaalen *equipartition conjecture of entropy production* is proposed, as a selection criterion among different variants resulting from classification among hierarchical trees. According to this conjecture for a given charge or duty, the best configuration of a flowsheet is the one in which entropy production is most uniformly distributed. One proceeds here by analogy using *information entropy* instead of thermodynamic entropy. Equipartition implies a linear dependence so that the *equipartition line* is described by:

$$h_{\text{eqp}} = h_{\max} b \quad (5)$$

Since the classification is discrete, a way of expressing equipartition would be a regular staircase function. The best variant is chosen to be that minimizing the sum of squares of the deviations:

$$SS = \sum_{b_i} (h - h_{\text{eqp}})^2 \quad (6)$$

5 Learning Procedure

Learning procedures similar to those encountered in *stochastic methods* are implemented as follows [24]. Consider a given partition into classes as *good* or ideal from practical or empirical observations, which corresponds to a *reference* similarity matrix $\mathbf{S} = [s_{ij}]$ obtained for equal weights $a_1 = a_2 = \dots = a$ and for an arbitrary number of fictitious properties. Next consider the same set of species as in the good classification and the actual properties. The similarity degree r_{ij} is then computed with Eq. (1) giving the matrix \mathbf{R} . The number of properties for \mathbf{R} and \mathbf{S} may differ. The learning procedure consists in trying to find classification results for \mathbf{R} , as close as possible to the *good* classification. The first weight a_1 is taken constant, and only the following weights a_2, a_3, \dots are subjected to random variations. A new similarity matrix is obtained using Eq. (1) and the new weights. The distance between the partitions into classes characterized by \mathbf{R} and \mathbf{S} is given by:

$$D = - \sum_{ij} (1 - r_{ij}) \ln \frac{1 - r_{ij}}{1 - s_{ij}} - \sum_{ij} r_{ij} \ln \frac{r_{ij}}{s_{ij}} \quad \forall 0 \leq r_{ij}, s_{ij} \leq 1$$

The result of the algorithm is a set of weights allowing adequate classification. The procedure was applied to the synthesis of complex flowsheets using information entropy [25].

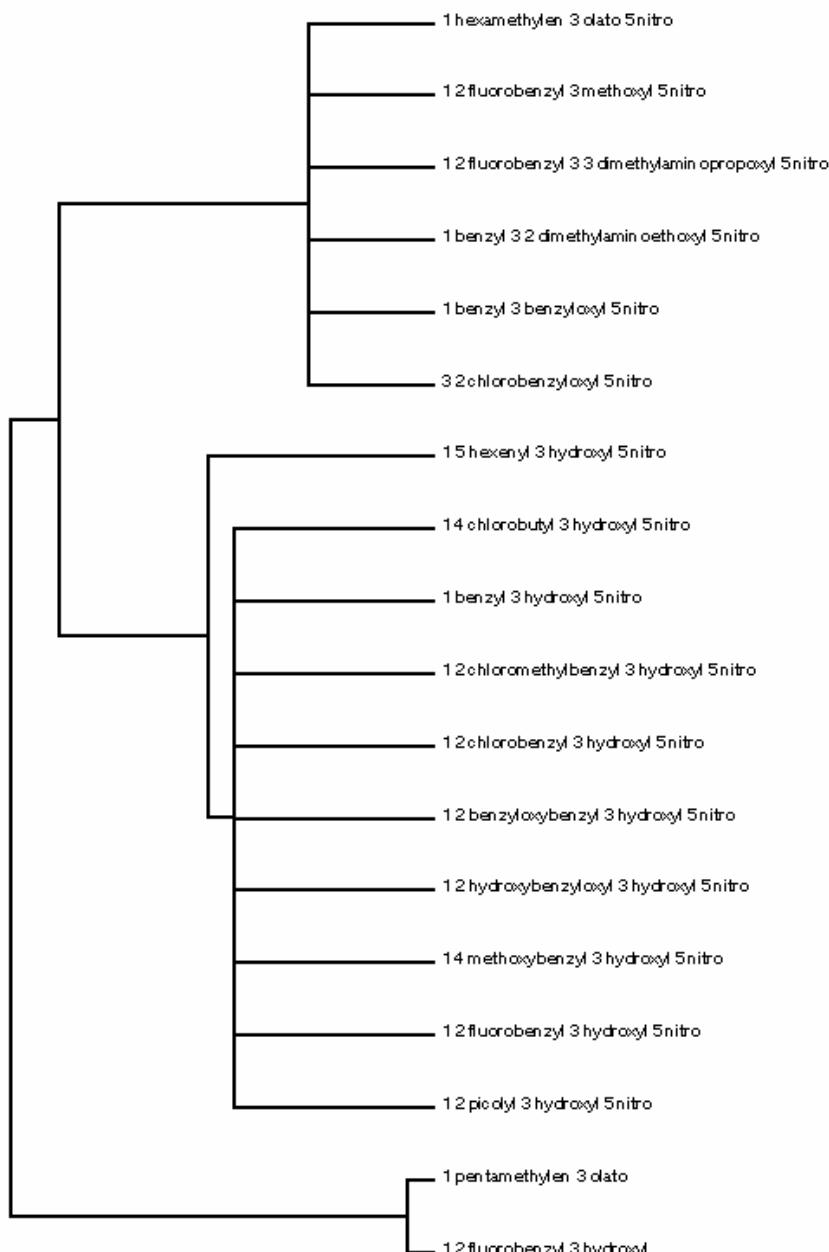


Fig. 2. Dendrogram for indazolols

6 Classification of Indazolols: Action against *T. vaginalis*

A set of 18 indazolol derivatives were assayed against *Trichomonas vaginalis* [26–28]. The *structural substitutions* of an indazolol molecule can be *ranked*, according to their contribution to inhibitory activity, in order: 5-nitro > 3-hydroxyl > 1,5'-hexenyl [29–31]. Reference molecule contains these substitutions; its associated vector is $\langle 111 \rangle$ [32–34]. All the indazolols that showed activity have a nitro group ($-\text{NO}_2$) at position 5, so it can be hypothetically considered the correspondence of the trichomonacide activity with the presence of this $-\text{NO}_2$ in these indazolols [35, 36]. The dendrogram for indazolols is illustrated in Fig. 2.

7 Classification of Fullerenes, Fullerite and C Nanotubes

The calculation of the Kekulé structure count K and permanent of adjacency matrix $\text{per}(\mathbf{A})$ of fullerenes allows the principal components analysis (PCA) of structural parameters and cluster analysis (CA) of fullerenes [37, 38]. Fullerene K and $\text{per}(\mathbf{A})$ are related to structural parameters involving the presence of contiguous pentagons p (number of edges common to two pentagons), q (number of vertices common to three pentagons), r (number of pairs of non-adjacent pentagon edges shared between two other pentagons) and, correspondingly, u , v and w for hexagons [39]. Structural parameter PCA and fullerene CA allow classifying them [40]. Linear and nonlinear correlations permit modelling K and $\text{per}(\mathbf{A})$ of fullerenes [41]. Structural parameter PCA agrees with CA of the fullerenes [42]. A simple linear correlation is a good model for $\text{per}(\mathbf{A})$ of fullerenes; $\{q,r,v,w\}$ is redundant information; $\{p,u\}$ contains the essential characters of $\text{per}(\mathbf{A})$ [43]. Fullerene dendrogram relating to $\{p,q,r,u,v,w,q/p,r/p,v/u,w/u\}$ separates first the 8 units in class 1 ($\text{C}_{20}\text{-I}_h\text{-C}_{30}\text{-C}_{2v}$ II, cf. Fig. 3), class 2 (8 units, $\text{C}_{32}\text{-D}_2\text{-C}_{34}\text{-C}_s$), class 3 (9 units, $\text{C}_{36}\text{-D}_{6h}\text{-C}_{40}\text{-D}_{5d}$ II), class 4 (3 units, $\text{C}_{40}\text{-T}_d\text{-C}_{44}\text{-D}_{3h}$) and class 5 (6 units, $\text{C}_{60}\text{-I}_h\text{-C}_{240}\text{-I}_h$). The classes correspond to PCA and radial tree. For classifying some fullerenes the dendrogram was repeated for smaller sets, resulting their enclosure in new branches attached to C_{28} (D_2) and C_{40} (T_d), respectively.

The Prout–Mendeleev–Meyer–Moseley periodic table (PT) of the elements suggested that hydrogen could be the origin of each element. The construction principle is an evolutionary process, which is formally similar to those of Darwin and Oparin. Inspired by PT of the elements a PT of fullerenes and its periodic law (PL) are proposed. Fullerene PT is built based on $\{p,q,r,u,v,w\}$, PCA and CA. The recommended fullerene PT format (cf. Table 2) shows that small fullerenes are classified first by p , then by q and, finally, by r ; larger fullerenes are furtherly grouped by u and v , and, at last, by w . Periods of 10 p units are assumed. Group $p0$ stands for $p = 0, 10, \dots$, subgroup $q0$, for $q = 0, 10, \dots$, and subsubgroup $r0$, for $r = 0, 10, \dots$, etc. Fullerenes in the same column of appear close in PCA, dendrogram (Fig. 3) and radial tree. Fullerene PT allows PL analysis. Fullerene PT built on $\{p,q,r,u,v,w\}$, PCA and CA shows that PL has not the rank of the laws of

physics: (1) fullerene properties are not repeated; (2) order relationships are repeated with exceptions. Proposed statement is: The relationships that any fullerene p has with its neighbour $p + 1$ are approximately repeated for each period.

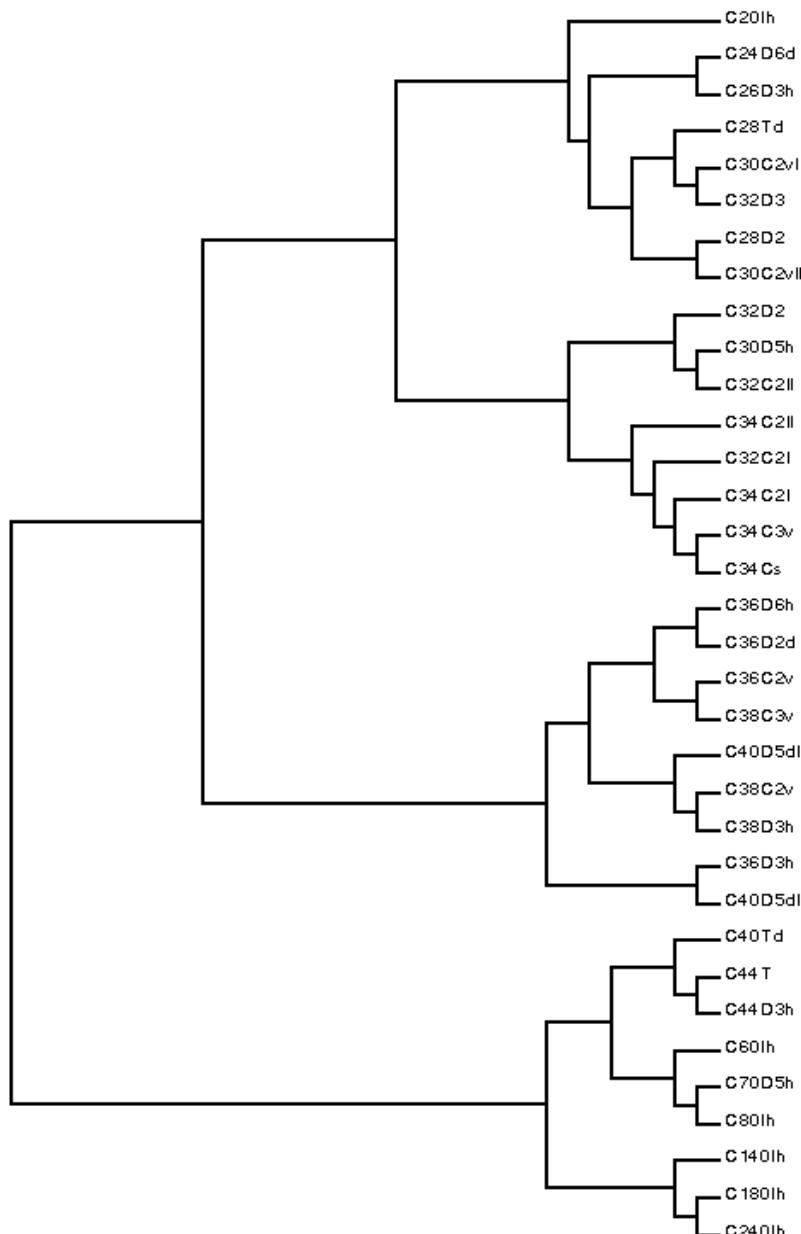


Fig. 3. Dendrogram for the fullerenes

Table 2. Table of periodic properties of fullerenes based on structural parameters

p0	p1	p2			p3
q0	q8	q8	q0	q1	q4
r0	r4	r0	r2	r9	r0
$60I_h^a$					
40D _{5d} I			36D _{2d}	38C _{3v}	40T _d
			36D _{6h}		44T
30D _{5h}	28D ₂	26D _{3h}			
40D _{5d} II					
20I _h					
p4		p5			
q2		q2	q3		q6
r4	r6	r8	r5	r6	r6
34C ₂ I		32D ₃	34C _{3v}	34C _s	36D _{3h}
38C _{2v}					
24D _{6d}					
p6	p7		p8		p9
q4	q4	q6	q4	q6	q2
r6	r0	r6	r4	r0	r0
				q8	
				r5	
				r8	
					44D _{3h}
32C ₂ I	30C _{2v} I	32C ₂ II	28T _d	30C _{2v} II	32D ₂
		34C ₂ II			38D _{3h}
u0		u5			
v0		v0			
w0		w8		w0	
60I _h					
					70D _{5h}
		80I _h			
140I _h					
180I _h					
240I _h					

^a 60I_h, 70D_{5h}, 72D_{6h}, 74D_{3h}, 76D₂, 78C_{2v}, 78D₃, 80D₂, 80I_h, 82C₂, 82C_{2v}, 84D_{2d}, 84D₂, 86C₂, 88D₂, 90C_{2v}, 92C₂, 94C_s, 96C_{3h}, 98, 100, 120T_d, 140I, 140I_h, 180I_h, 240I_h, 260I, 320I_h, 380I, 420I, 500I_h, 540I_h, 560I, 620I, 720I_h, 740I, 780I, 860I, 960I_h, 980I_h, 980I, 1500I_h.

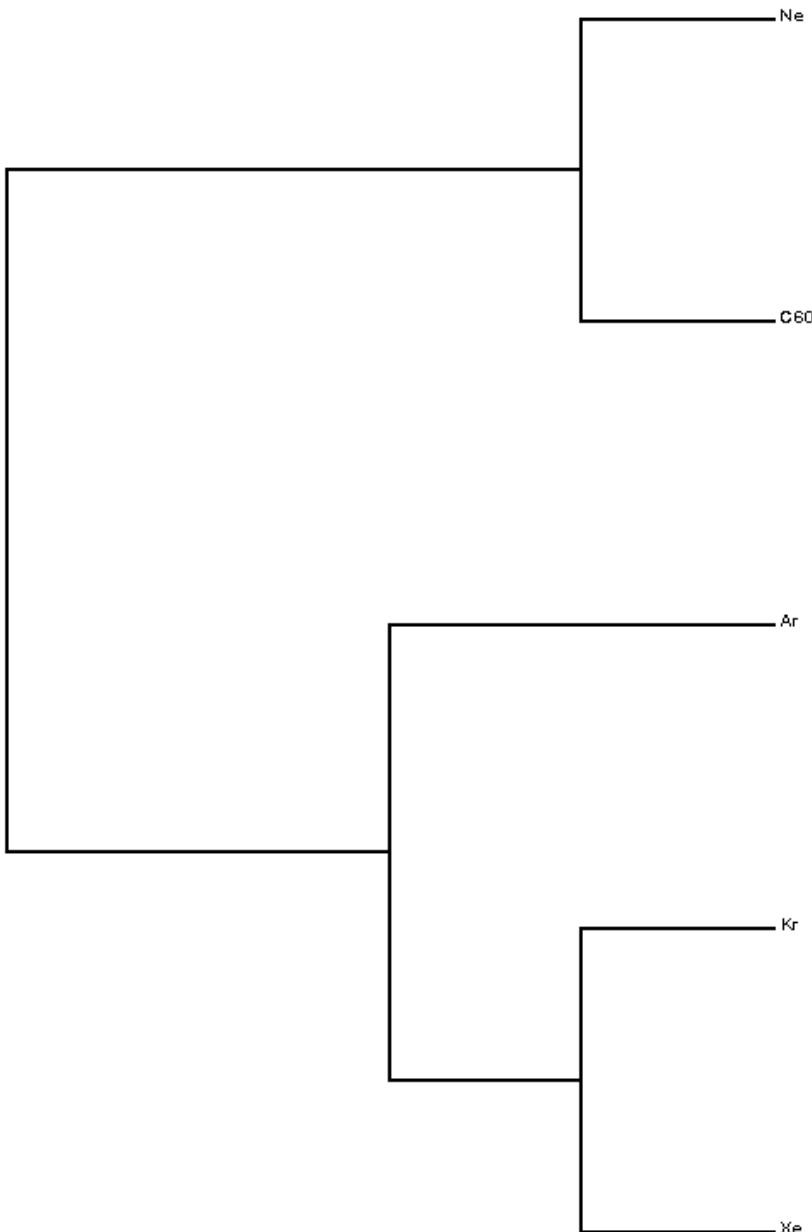


Fig. 4. Hierarchical CA dendrogram for properties of inert-gas/C₆₀ crystals

Growth mechanisms of fractal clusters in fullerene solutions are analyzed along with similarity laws determining the thermodynamic characteristics of fullerite crystals [44–47]. A simplified examination of thermodynamic properties in CA

can be obtained from hierarchical cluster analysis (HCA) [48–51]. The most important result is that in dendrogram Ne (class 1) is separated from Ar, which joins {Kr,Xe} in class 2 (*cf.* Fig. 4). Separation of Ne is attributed to well-known quantum effects at low temperatures. Dendrogram is in agreement with partial correlation diagram and radial tree. In particular Ne is closer to C₆₀ than to remaining inert gases. However, result should be taken with care because binary-tree structure is forced by HCA.

It was discussed the existence of single-wall carbon nanotubes (SWNTs) in organic solvents in the form of clusters [52, 53]. A theory was developed based on a *bundle* model [54]. Comparison of calculated solubilities with experiments would permit obtaining energetic parameters characterizing the interaction of an SWNT with its surrounding [55]. Fullerenes and SWNTs are objects whose behaviour in many physical situations is characterized by peculiarities, which show up in that these systems represent the only soluble forms of carbon [56]. Fullerene molecule is a virtually uniform closed spherical–spheroidal surface, and an SWNT is a smooth cylindrical unit [57]. Both give rise to weak interactions between neighbouring units in a crystal and promote their interaction with solvent molecules [58]. Phenomena had a unified explanation in bundle model, in which the free energy of an SWNT in a cluster is combined from two components: a volume one proportional to the number of molecules n in a cluster and a surface one proportional to $n^{1/2}$ [59]. Some properties are calculated with the aim of discussing SWNT PT format [60]. Elementary polarizability $\langle\alpha\rangle$ and geometric, topological and solvation properties are computed because experimental properties depend on the sample, since some samples: (1) contain fullerenes, (2) consist of diameters d_t of metallic–semimetallic or semiconductor SWNTs, (3) show polydispersity between short and large SWNTs, (4) solubility of SWNTs differ for different d_t and (5) SWNTs thinner than (5,5) are scarce [61]. The SWNT properties are related to the indices (n,m) designating the *chiral vector* [62]. The SWNTs are classified in *zigzag* $(n,0)$, *armchair* (n,n) and *chiral* (n,m) [63]. The properties allow classifying SWNTs according to (n,m) [64]. The $\langle\alpha\rangle$ relationship of any SWNT (n,m) is similar to that of its neighbour $(n-1,m+1)$ [65]. The trend is repeated for each period [66]. Correlations between $(n^2+nm+m^2)^{1/2}$ and other properties show that (n,m) are adequate indices [67]. The (10,10) is the favourite, presenting great kinetic stability and small $\langle\alpha\rangle$, pyramidalization angle, fractal index D_{cavity} and solubility, and great d_t , linear density, D , density and 1-octanol/cyclohexane/chloroform–water partition coefficients [68]. SWNTs in some organic solvents are positively charged, while in water–Triton X are negative, which is explained on the basis of permittivity–electron affinity [69]. The $\langle\alpha\rangle$ allows the format presented (*cf.* Table 3) for (n,m) SWNT PT. SWNTs are classified by n and m [group m0 ($m = 0$) includes the $(n,0)$ SWNTs]. Periods of

$(n+1)/2$ (n odd) and $(n+2)/2$ (n even) units are assumed because $n \geq m$. SWNTs in the same row (period) show close values of $\langle \alpha \rangle$. The $\langle \alpha \rangle$ increases from top to bottom and from left to right. The $\langle \alpha \rangle_{(10,10)}$ is almost the greatest for all SWNTs. The $\langle \alpha \rangle_{(n,0)-(n,n)}$ correlate with $(n^2 + nm + m^2)^{1/2}$.

Table 3. Periodic table of nanotubes. Differential elementary polarizability (\AA^3)^a.

m0	m1	m2	m3	m4	m5
(9,0)	(8,1)	(7,2)	(6,3)	(5,4)	–
1.130	1.076	1.089	1.119	1.096	
-0.054	0.013	0.030	-0.023	0.049	
(10,0)	(9,1)	(8,2)	(7,3)	(6,4)	(5,5)
1.145	1.122	1.090	1.096	1.094	1.136
-0.023	-0.032	0.006	-0.002	0.042	0.024
(11,0)	(10,1)	(9,2)	(8,3)	(7,4)	(6,5)
1.160	1.139	1.099	1.098	1.109	1.104
-0.021	-0.040	-0.001	0.011	-0.005	0.071
(12,0)	(11,1)	(10,2)	(9,3)	(8,4)	(7,5)
1.175	1.127	1.099	1.098	1.089	1.114
-0.048	-0.028	-0.001	-0.009	0.025	0.044
(13,0)	(12,1)	(11,2)	(10,3)	(9,4)	(8,5)
1.189	1.112	1.110	1.120	1.113	1.134
-0.077	-0.002	0.010	-0.007	0.021	-0.011
(14,0)	(13,1)	(12,2)	(11,3)	(10,4)	(9,5)
1.202	1.116	1.105	1.128	1.127	1.143
-0.086	-0.011	0.023	-0.001	0.016	-0.004
(15,0)	(14,1)	(13,2)	(12,3)	(11,4)	(10,5)
1.215	1.120	1.161	1.133	1.144	1.113
-0.095	0.041	-0.028	0.011	-0.031	0.016
(16,0)	(15,1)	(14,2)	(13,3)	(12,4)	(11,5)
1.227	1.133	1.156	1.153	1.142	1.152
-0.094	0.023	-0.003	-0.011	0.010	0.004
(17,0)	(16,1)	(15,2)	(14,3)	(13,4)	(12,5)
1.239	1.147	1.166	1.156	1.161	1.165
-0.092	0.019	-0.010	0.005	0.004	0.014
(18,0)	(17,1)	(16,2)	(15,3)	(14,4)	(13,5)
1.249	1.166	1.170	1.176	1.161	1.165
-0.083	0.004	0.006	-0.015	0.004	-0.024
(19,0)	(18,1)	(17,2)	(16,3)	(15,4)	(14,5)
1.137	1.192	1.174	1.187	1.180	1.183
0.055	-0.018	0.013	-0.007	0.003	0.009
(20,0)	(19,1)	(18,2)	(17,3)	(16,4)	(15,5)
1.151	1.182	1.189	1.182	1.188	1.179
0.031	0.007	-0.007	0.006	-0.009	0.018

Table 3. (continued)

m6	m7	m8	m9	m10
—	—	—	—	—
—	—	—	—	—
—	—	—	—	—
(6,6)	—	—	—	—
1.158				
0.031				
(7,6)	—	—	—	—
1.123				
0.079				
(8,6)	(7,7)	—	—	—
1.139	1.179			
0.040	0.036			
(9,6)	(8,7)	—	—	—
1.129	1.154			
0.025	0.073			
(10,6)	(9,7)	(8,8)	—	—
1.156	1.156	1.200		
0.000	0.044	0.039		
(11,6)	(10,7)	(9,8)	—	—
1.179	1.169	1.171		
-0.010	0.002	0.078		
(12,6)	(11,7)	(10,8)	(9,9)	—
1.141	1.192	1.166	1.219	
0.051	-0.026	0.053	-0.082	
(13,6)	(12,7)	(11,8)	(10,9)	—
1.192	1.195	1.193	1.197	
0.003	-0.002	0.004	-0.046	
(14,6)	(13,7)	(12,8)	(11,9)	(10,10)
1.197	1.199	1.180	1.211	1.237
0.002	-0.019	0.031	0.026	—

^a Values whose sign is opposite to that in its group are boldfaced; exceptions reaching 2% are italicized.

8 Classification of Lactic Acid Bacteria

The immune system evolved in its fight against different pathogens [70, 71]. The system has different levels of action of increasing complexity; it could be modulated by different environmental factors, being diet one of the most important [72, 73]. There is interest in the impact of different foods or nutrients over the immune system (immunonutritional studies), including the functional food called *probiotics* [74, 75]. Probiotics are live microbial food supplements, which benefit host improving its intestinal microbial balance and which upon ingestion in certain

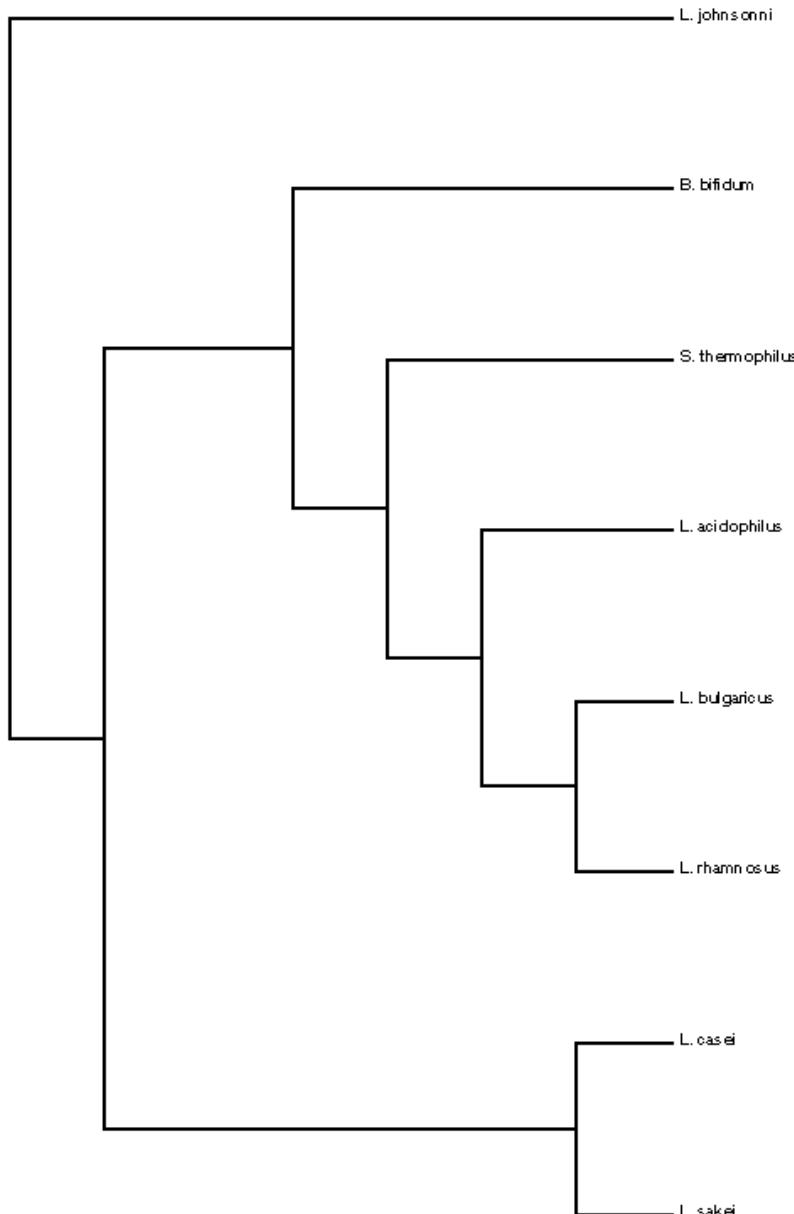


Fig. 5. Dendrogram for effects of lactic acid bacteria on production of cytokines

numbers induce health benefits beyond inherent basic nutrition [76, 77]. Yoghurt is a coagulated milk obtained by lactic acid fermentation in the presence of *L. bulgaricus* and *S. thermophilus* [78, 79]. Belief that yoghurt may be beneficial to health is centuries old [80, 81]. An immunostimulatory effect of yoghurt was

proposed based on its preventive effect on diseases (cancer, infections, gastrointestinal disorders or asthma) [82, 83]. However, results were controversial [84, 85]. The number of volunteers on study, strain of bacteria, quantity of intake, analyzed parameters, time of intervention and studies on animal vs. humans were different variables conduced finally to a puzzle that might be resolved [86, 87]. Although some functions of lactic acid bacteria (LAB) can reside in either cytoplasm or cell-wall components (being the heat-inactivated product more effective in the latter), more effects over immune system were demonstrated for the fresh yoghurt than for the heat-inactivated one [88, 89]. The classification of living and heat-inactivated LABs against cytokines was studied [90, 91]. Eight LABs were classified using the effects (induction, inhibition or no effect) of living and heat-inactivated LABs on the production of nine cytokines (*cf.* Fig. 5). The obtained classification (((((1,4),2),7),6),(3,5)),8) is in agreement with principal component analysis.

9 Phylogeny of Avian Birds and 1918 Influenza Virus

Lysozyme is an enzyme with 129 residues. The amino-acid compositions of certain avian lysozymes was determined. The amino-acid sequence of hen egg-white lysozyme was annotated [92]. Certain discrepancies exist between this sequence and that reported in [93] (at residues 40, 41, 42, 46, 48, 58, 65, 66, 92 and 93). Crystallographic analysis [94] gave results for residues 40, 41, 42, 58, 59, 92 and 93 that are in agreement with the former. Discrepancies at residues 46, 48, 65 and 66 are a difference between Asp or Asn; from the electron density maps it could not be determined whether these residues are amide or free acid. The amino-acid sequences of duck, Japanese quail and turkey egg-white lysozymes were determined [95–97]. Amino-acid sequences for human urine and milk lysozymes were determined. Comparative studies of sequences for lysozymes of different origins are interesting from the viewpoint of structure–function relationships (the Asp-101 of hen lysozyme, which is known to be implicated at the substrate binding site, is replaced by Gly in turkey lysozyme). The Trp-62 of hen lysozyme, which also plays an important role in substrate binding, is replaced by Tyr in human lysozyme. Differences between avian species sequences that are compared are expressed as percentage of different amino acids in lysozyme. The greater the differences, the farther in time must be separation between species. *Grouping level b* can be identified with *biological time*. Obtained *phylogenetic tree* is represented by scheme: (1,...,5) → (1,4,5)(2,3) → (1,5)(2,3)(4) → (1)(2,3)(4)(5) → (1)(2)(3)(4)(5). The scheme is in agreement with data obtained in morphological studies. Optimality criterion SS associated with different proposals for phylogenetic trees allows *equipartition conjecture* to be validated or invalidated in phylogenesis. If, in the calculation of *entropy* associated with the phylogenetic tree, a species is systematically omitted, difference between entropy with and without this species can be considered as a measure of *species entropy*. Such contributions may be studied with equipartiton conjecture. Available experimental and field data should be examined by different classification algorithms to reveal possible features of real biological significance. Scheme (*cf.* Fig. 6) is in agreement with data obtained in morphological studies and with the method based on entropy

production and the conjecture of equipartition of production of entropy. Each band of parallel edges indicates a split. The distance between any two taxa x and y corresponds to the sum of weights of all splits that separate x and y .

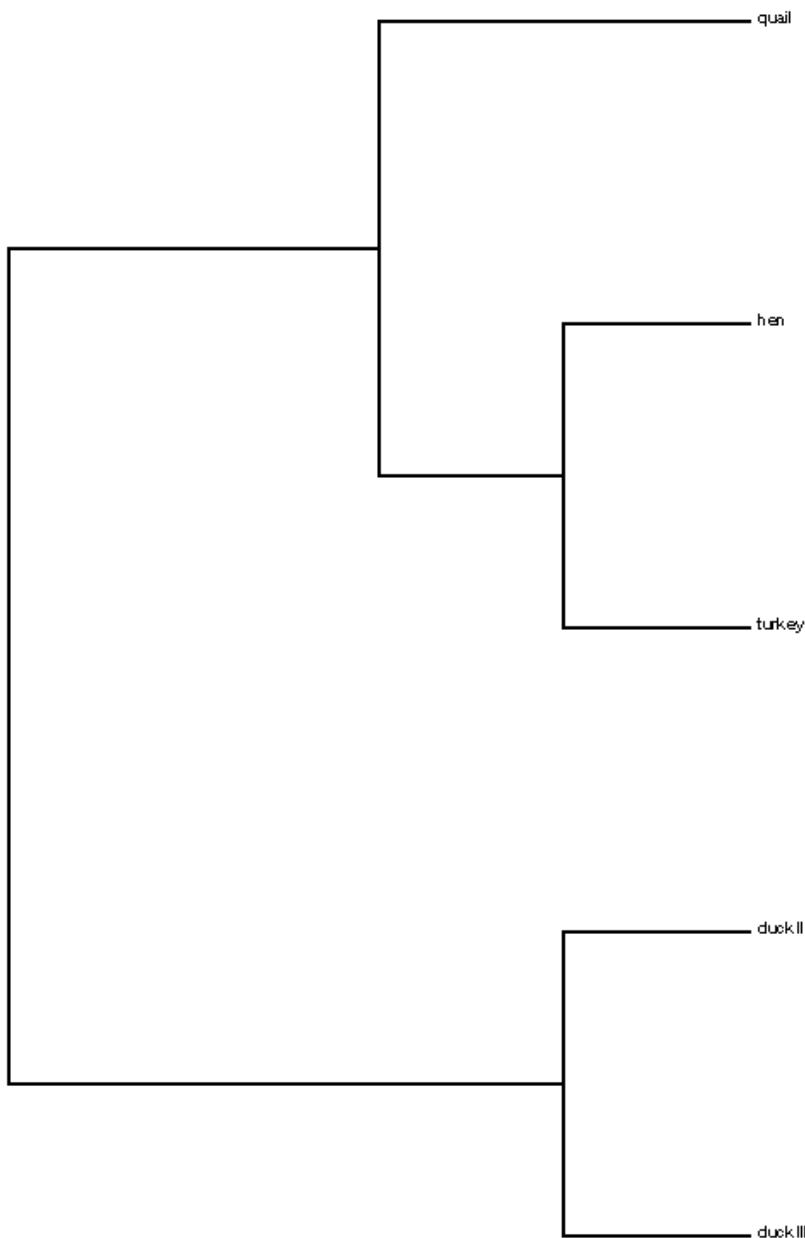


Fig. 6. Dendrogram for distances as different amino acids in lysozyme

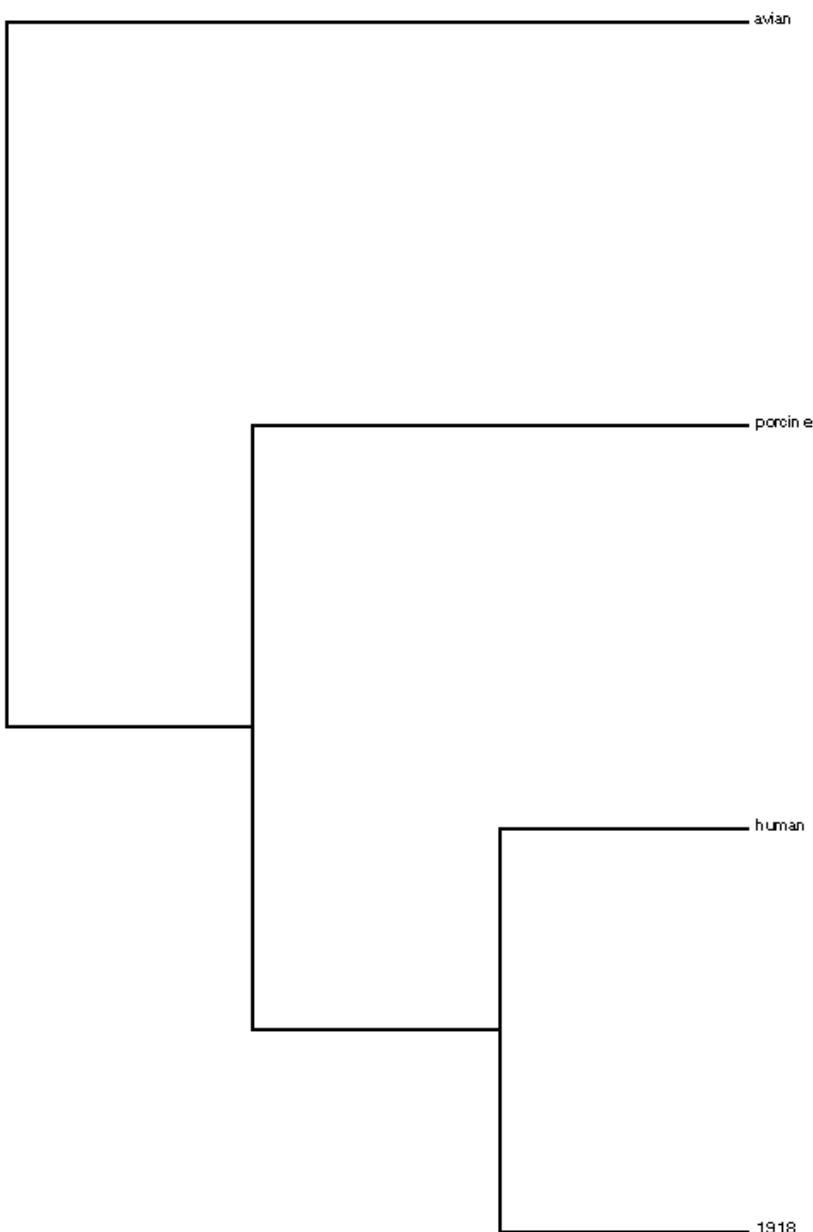


Fig. 7. Family dendrogram of influenza

An arsenal of effective medicines and others in developing phase is available. Research in viral genomes expedites progress [98–101]. In the search of the keys of the origin of 1918 virus hemagglutinin (HA), the gene sequences of HA subtype H1 of several strands of influence virus were analyzed [101–104]. Its phylogenetic tree

was built [105]. The samples of 1918 strand are inscribed in that family of influenza virus adapted to man (*cf.* Fig. 7) [106]. Distance between 1918 gene H1 and known avian family reflects that it was originated in a strand of avian influenza virus, although it evolved in an unidentified host before emerging in 1918.

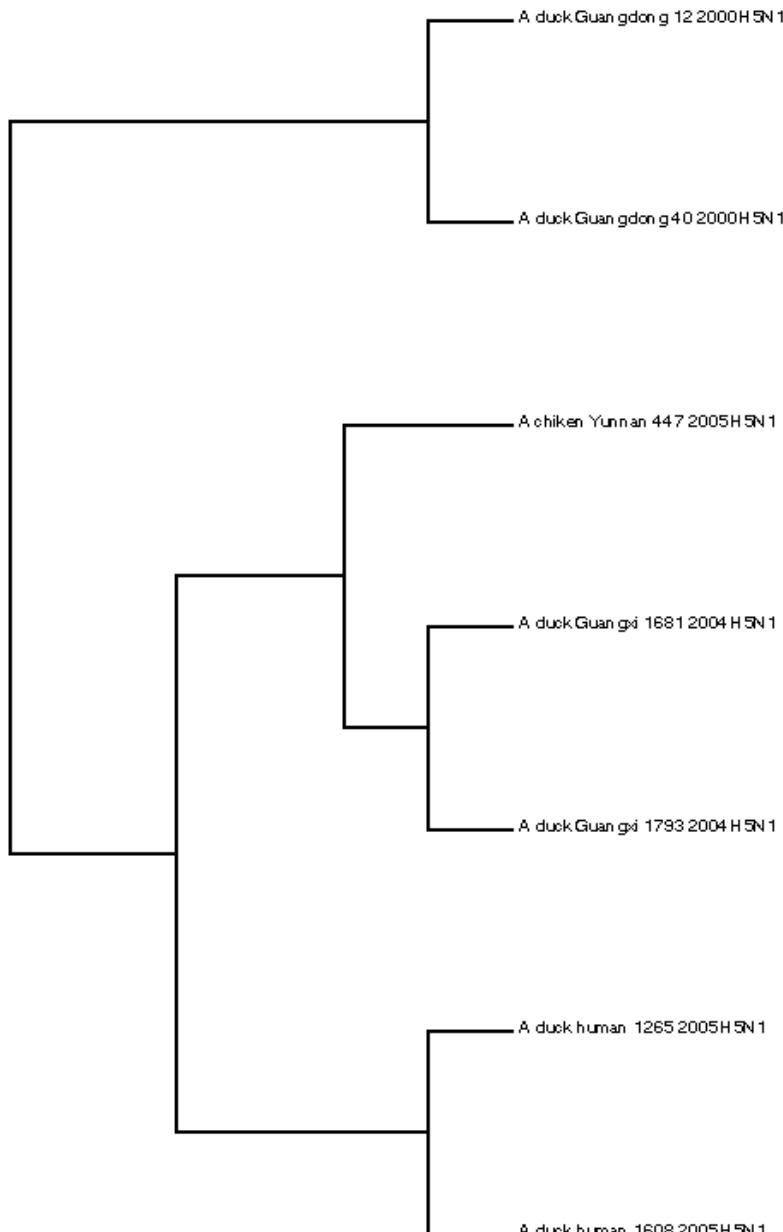


Fig. 8. Phylogenetic tree for the seven HA (H5N1) sequences

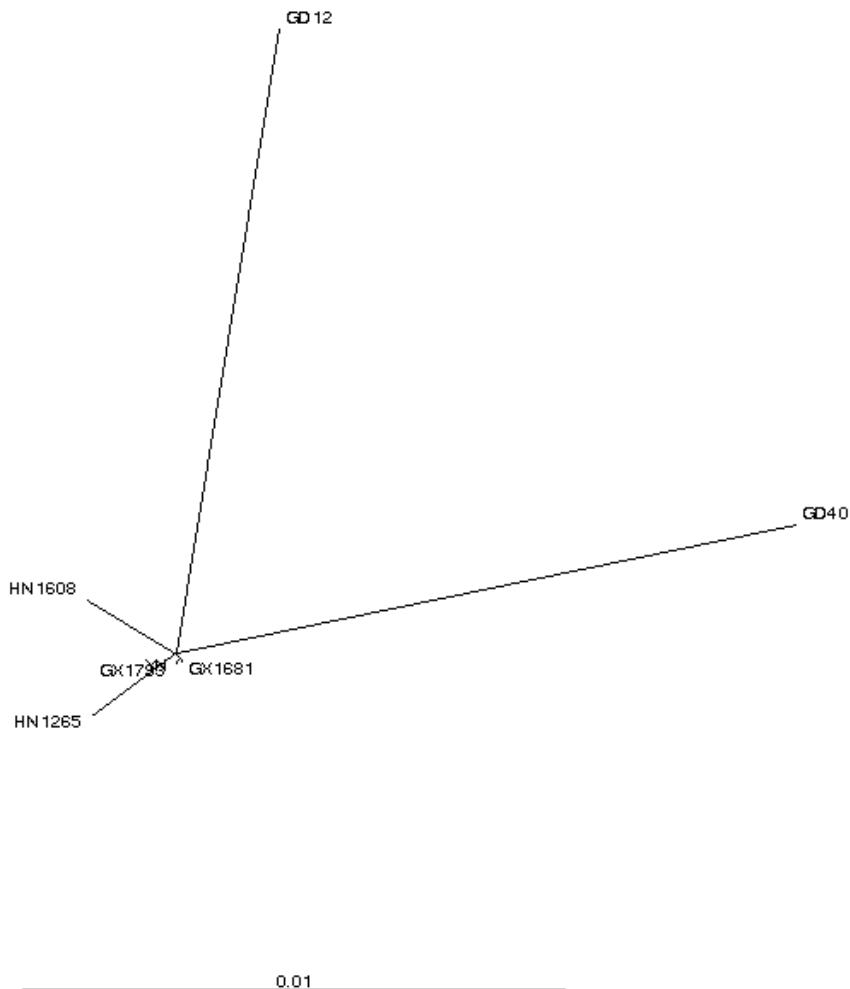


Fig. 9. Radial tree for the seven HA (H5N1) sequences

The Z_{inv} , a new invariant based on 3DD-curves of DNA sequence, which is simple for calculation and it approximates to the leading eigenvalues of the matrix associated with DNA sequence [107]. The utility of the invariant is illustrated on the DNA sequence of 11 species. The Z_{inv} is used to analyze the phylogenetic relationships for the seven HA (H5N1) sequences of avian influenza virus (*cf.* Figs. 8–10). Biomacromolecular structural data are maintained by Protein Data Bank (PDB) [108]. Classroom applications were described [109–112]. Three-dimensional (3D) structures can be displayed by program RasMol [113]. Program WPDB compresses PDB structure files into a set of indexed files [114, 115]. Program BABEL converts molecular modelling file formats [116]. Database system RELIBASE+

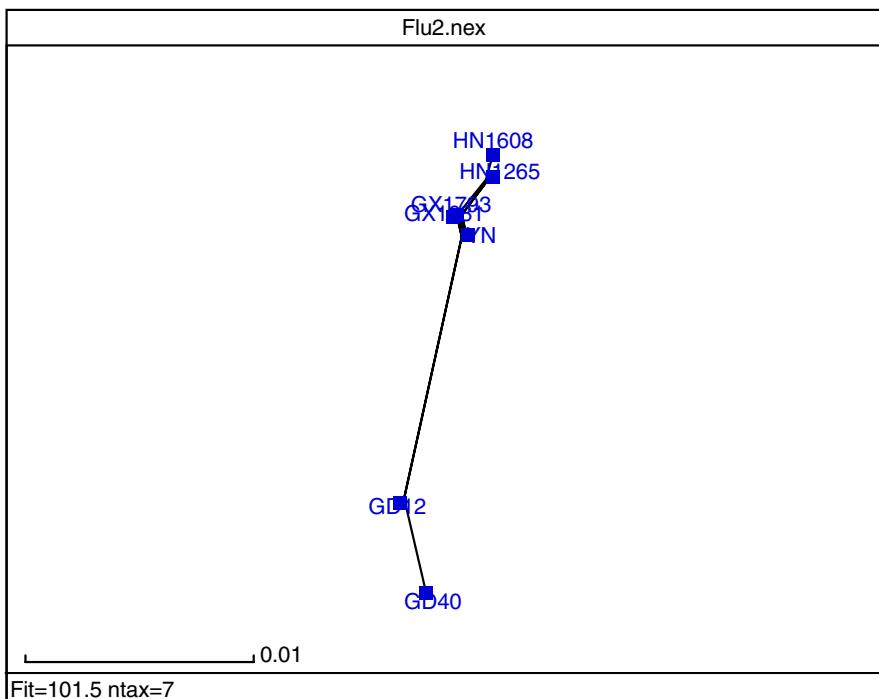
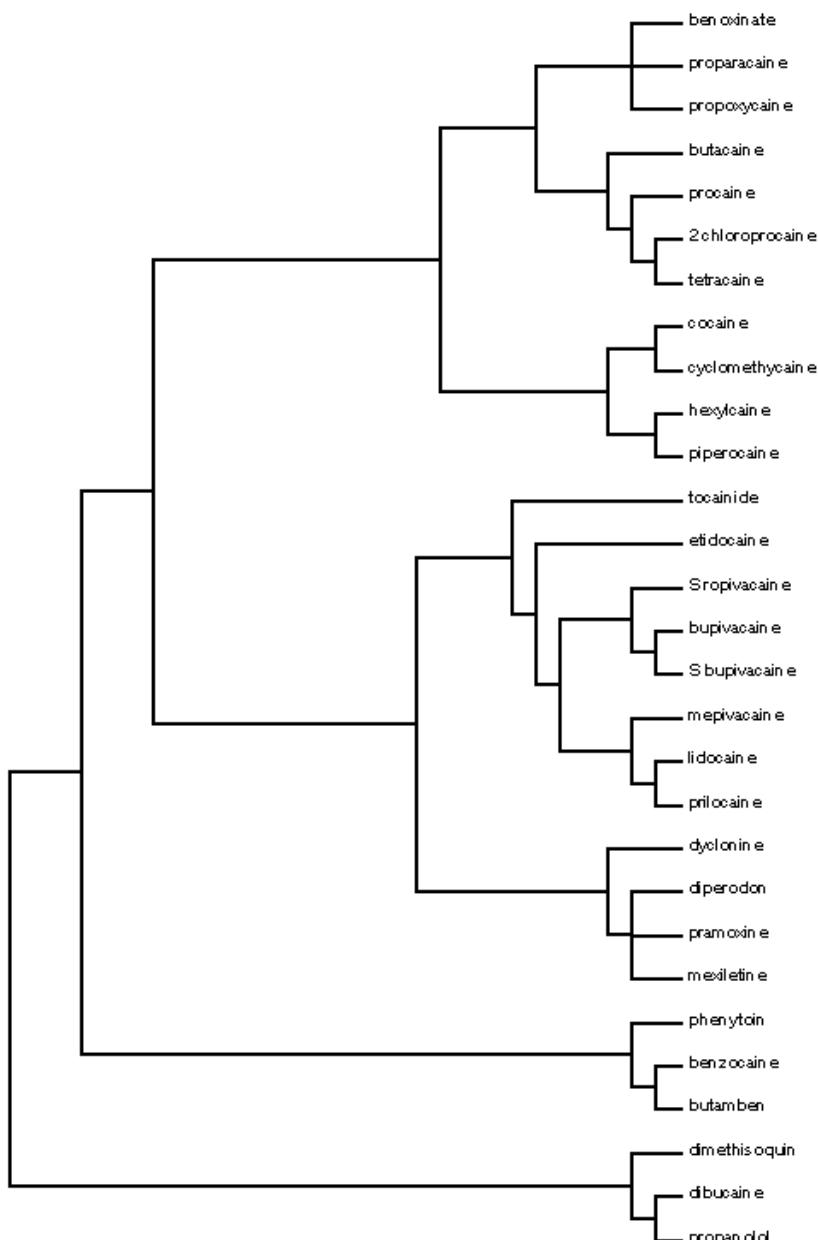


Fig. 10. Splits graph for the seven HA (H5N1) sequences

analyzes protein-ligand structures in PDB [117]. Classroom applications of WPDB were described [118]. Successor of RasMol and Chime [119] is Jmol [120–127]. Fractals for hybrid orbitals in protein models were proposed [128–134].

10 Periodic Classification of Local Anaesthetics

The classification of 28 local anaesthetics (procaine analogues, *cf.* Fig. 11) is in agreement with partial correlation diagrams, *dendograms* (binary trees), previous results with 27 anaesthetics and earlier publications [135–138]. Classification scheme from 1–11 levels is conserved after the addition of S-ropivacaine. S-bupivacaine was compared with racemic bupivacaine [139]. S-ropivacaine is structurally close to bupivacaine; the main difference is that the former is a pure S-(–) enantiomer where the latter is a racemate. Ester and amide local anaesthetics are grouped into different classes; the agents of low potency and short duration are separated from the agents of high–medium potency and long–medium duration. The classification presents lower bias and greater precision, resulting in lower divergence with respect to original distribution. A natural trend is to interchange similar anaesthetics in composition of complex drugs [140]. However, mixtures of dissimilar anaesthetics are also used [141–143].

**Fig. 11.** Dendrogram for local anaesthetics

The recommended format for the PT of the local anaesthetics (*cf.* Table 4) shows that they are classified first by i_5 , then by i_4 , i_3 , i_2 and, finally, by i_1 . Periods of five units are assumed. Group g010 stands for $\langle i_1, i_2, i_3 \rangle = \langle 010 \rangle$ [$\langle 01001 \rangle$

Table 4. Periodic properties for local anaesthetics (procaine analogues) and ice

g000 ice	g010 dibucaine, propanolol	g100 dimethisoquin
		phenytoin
g101 benzocaine, butamben	g110 diperodon, pramoxine, mexiletine	g111 cocaine, cyclomethycaine
	bupivacaine, etidocaine, lidocaine, mepivacaine, prilocaine, tocainide, S-ropivacaine	benoxinate, proparacaine, propoxycaine
		butacaine, 2-chloroprocaine, procaine, tetracaine

(dibucaine, propanolol) and <01010> (dimethisoquin)], etc. The local anaesthetics in the same column appear close in partial correlation diagrams, dendograms, radial trees, splits graph and principal component analysis. The variation of vector property $P = \langle i_1, i_2, i_3, i_4, i_5 \rangle$ as a function of structural parameters $\{i_1, i_2, i_3, i_4, i_5\}$ shows that the lines for the structural parameters i_4 and i_5 appear superposed, what agrees with a PT of properties with vertical groups defined by $\{i_1, i_2, i_3\}$ and horizontal periods described by $\{i_4, i_5\}$. The variation of $P = \langle i_1, i_2, i_3, i_4, i_5 \rangle$, as a function of the number of group in PT, reveals that the minima correspond to $\langle i_1, i_2, i_3 \rangle = <010>$ (group g010). Corresponding function $P(i_1, i_2, i_3, i_4, i_5)$ reveals a series of waves clearly limited by maxima or minima, which suggest a periodic behaviour. For $\langle i_1, i_2, i_3, i_4, i_5 \rangle$ two minima are clearly shown.

11 Classification of Transdermal-Delivery Enhancers

Skin offers an excellent barrier to molecular transport [144]. There was interest in developing systems for controlled delivery of drugs [145]. A suggested technique was to join a specialized patch on skin [146]. Simplest scheme utilized uncoated

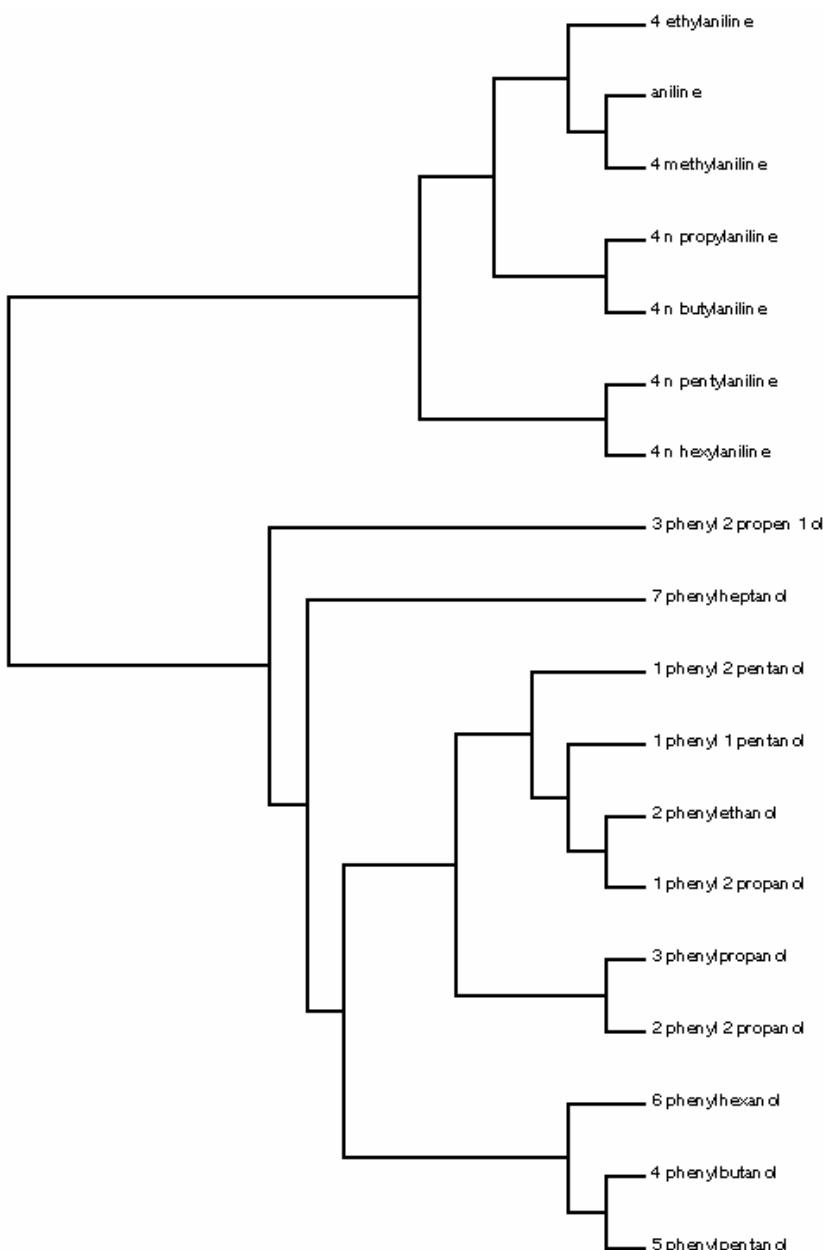


Fig. 12. Dendrogram of percutaneous transdermal-delivery drug models

polymer matrices containing embedded drug [147]. Pathways form percolating path [148]. Pathways are fractal structures [1489]. Several groups developed computational methods for predicting fluxes, including multiple regression methods

[150]. It was reviewed some selective ways for circumventing *stratum corneum* barrier [151]. There was a significant effort directed to finding new drug release systems in which bioactives contained in a reservoir can be supplied to a host system while controlling the rate and period of delivery [152]. Redox chemistry approach was performed to conducting electroactive polymers for drug delivery and sensing of bioactives. A drug-delivery system consisting of a specialized patch joined to the skin was proposed. Fractal dimensions of transdermal-delivery drug models have been proposed [153, 154]. It is shown (Fig. 12) the molecular classification of transdermal-delivery drug models, percutaneous enhancers of 5-fluorouracil penetration.

12 QSAR Modelling of Anti-HIV-1 Compounds

Investigation was performed *via* theoretical approaches, in the field of quantitative structure–activity relationship (QSAR) studies, with a goal of explaining anti-HIV-1 activity, in terms of structural, physicochemical, topological and quantum chemical parameters [155]. Classical approaches, receptor surface analysis (RSA), molecular shape analysis (MSA) and molecular field analysis (MFA) were performed as 3D-QSARs [156]. Overall aim was to explore important interaction sites and optimum physicochemical requirements of selected anti-HIV compounds, and to develop suitable models of statistical quality with prediction potential, which may help to develop newer compounds with desired activity [157]. The main aim of a medicinal chemist is to discover novel drugs with greater potency and reduced toxicity, which may be achieved by molecular modification of existing drugs, optimization of various lead compounds, isolation of active constituents from natural sources or syntheses of new series of compounds [158]. Conventional drug discovery process is costly and time consuming [159]. Researchers try to highlight novel methods, which reduce time and cost involvement in drug discovery [160]. The goal of medicinal chemistry is to design and synthesize novel pharmacologically active molecules with reduced toxicity [161]. The QSARs increase the probability of success and reduce time and cost in drug discovery [162]. A rational explanation of drug action is often limited by one's ability to correlate observed physiological effects with a reasonable hypothesis or concept [163]. Various structural, physicochemical and biological parameters are used to correlate these with biological activity; observed relations are used to predict the activity of a new compound, and this information is exploited to develop newer molecules of optimum activity [164]. A QSAR of selected classes of anti-HIV ligands was attempt and it was tried to find out the impact of various structural, fragmental, physicochemical and substitutional requirements on binding affinity. Used descriptors were charge parameters, physicochemical substituent parameters and topological descriptors. Chemometric tools: (1) stepwise regression, (2) multiple linear regression with factor analysis as data preprocessing step for variable selection (FA-MLR), (3) partial least squares with factor analysis as

preprocessing step (FA-PLS), (4) multiple linear regression with genetic function approximation (GFA-MLR), (5) genetic partial least squares (G/PLS) and (6) principal component regression analysis (PCRA) were used to identify relation



Fig. 13. Dendrogram for CCR5 antagonists

between various descriptors for biological activity. It was performed QSAR of 3-(4-benzylpiperidin-1-yl)-*N*-phenylpropylamine derivatives as potent CCR5 antagonists (*cf.* Fig. 13).

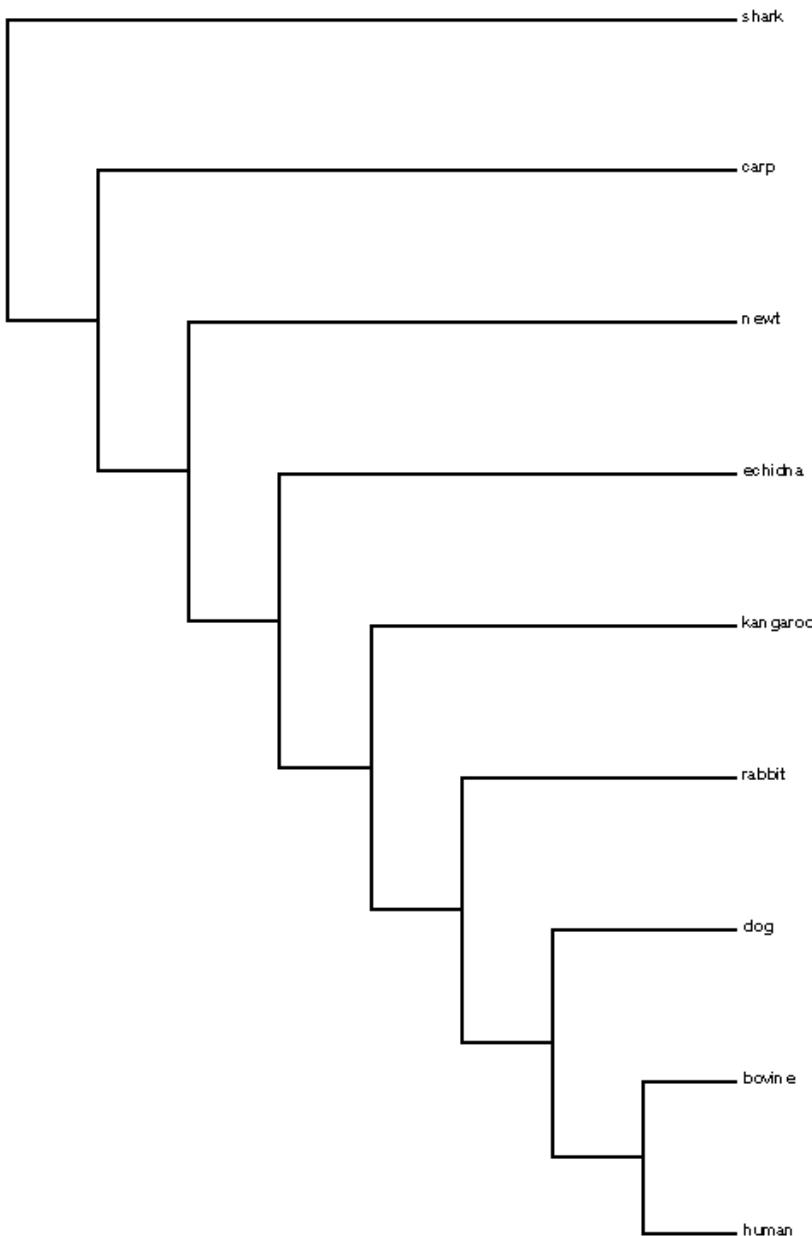


Fig. 14. Phylogenic tree of the vertebrates

13 Phylogenesis of Vertebrates, Mammals and Monkeys

It was carried out a study, toward the automatic reconstruction of a highly resolved tree of life, by Dopazo's group [165–277] and others [278–291]. A detailed

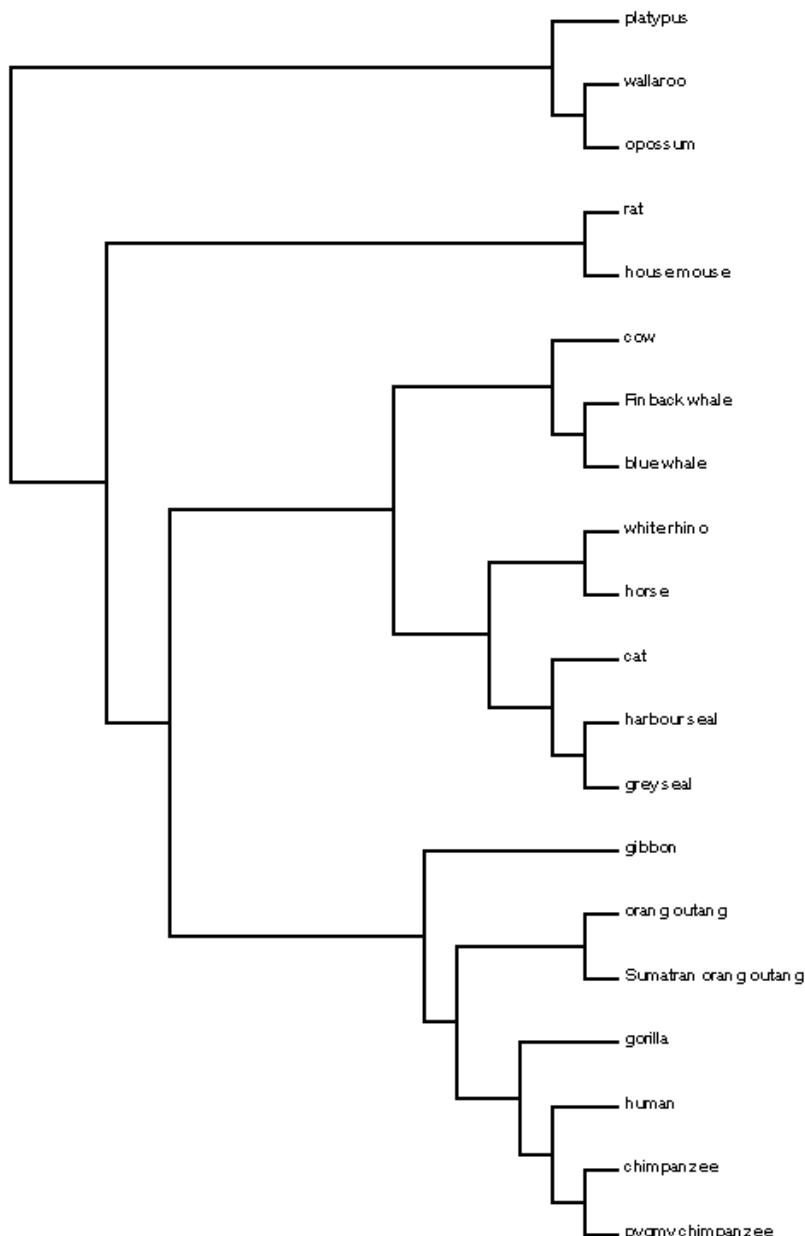


Fig. 15. Phylogenetic tree I built from the complete mammalian mtDNA

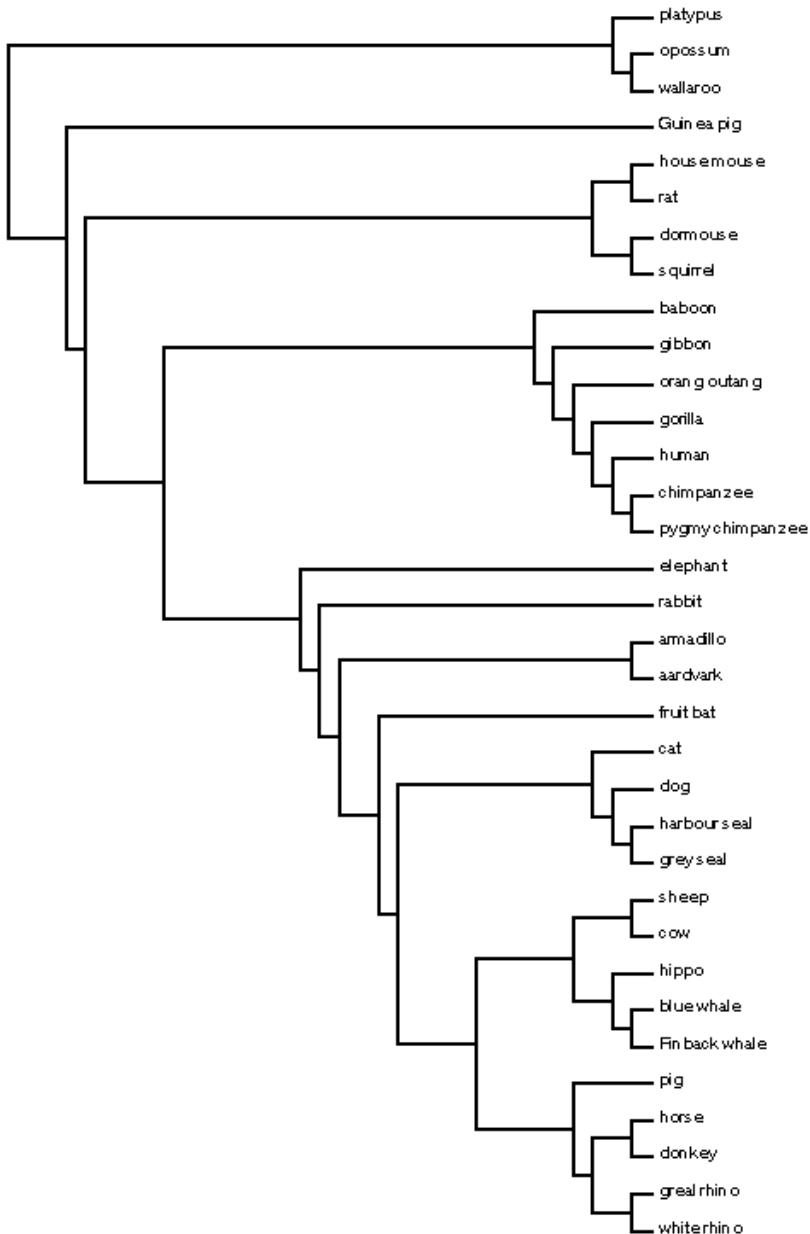


Fig. 16. Phylogenetic tree II built from the complete mammalian mtDNA

comparison of the sequences (primary structures) of biomolecules, proteins or nucleic acids, allows for molecular reconstruction. The key argument in molecular phylogeny is the existence of molecular clocks, *i.e.*, a constant rate of change of a given molecule. Various similarity-distance indices can be constructed from such

primary data; *e.g.*, one can define the distance as the number of amino-acid differences between the haemoglobin α -chain of each couple (i,j) of some vertebrates. The phylogenetic tree of nine vertebrates is shown in Fig. 14.

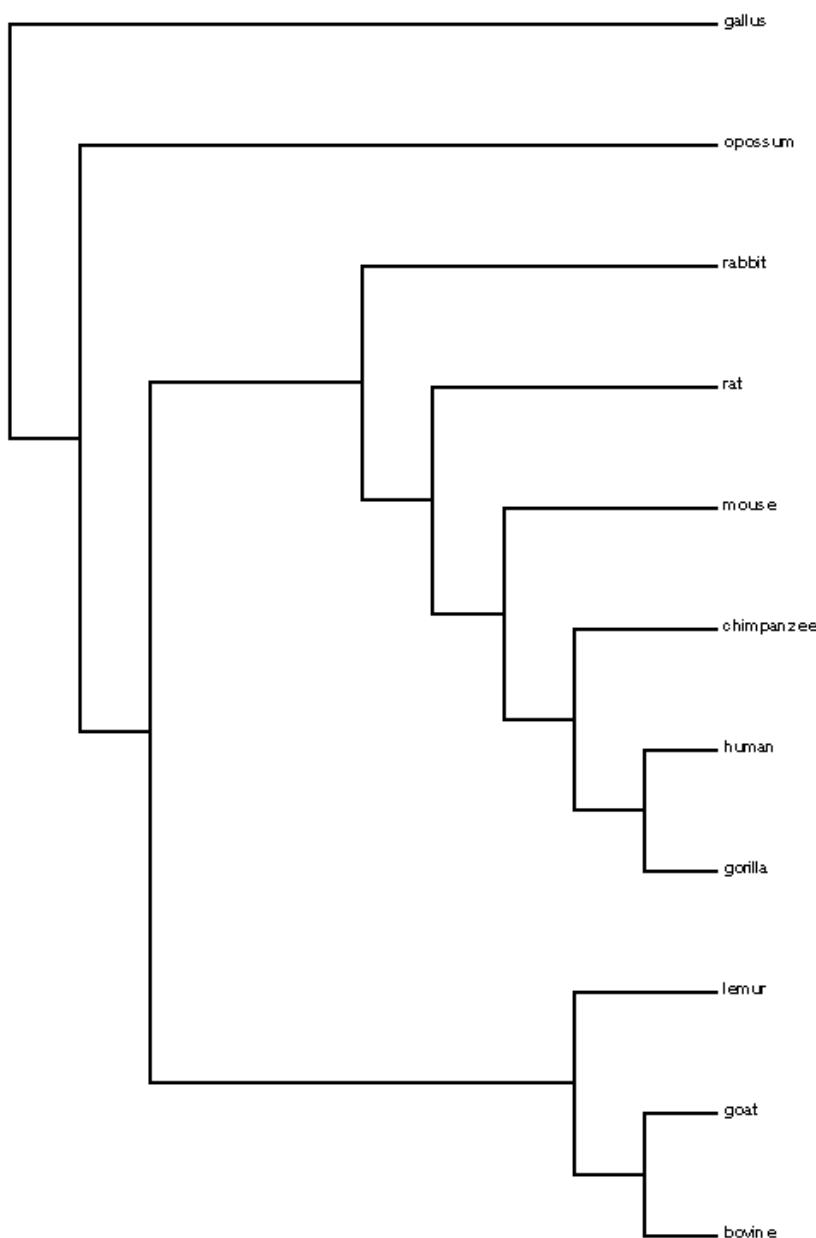


Fig. 17. Topological UPGMA species tree based on the nucleotides

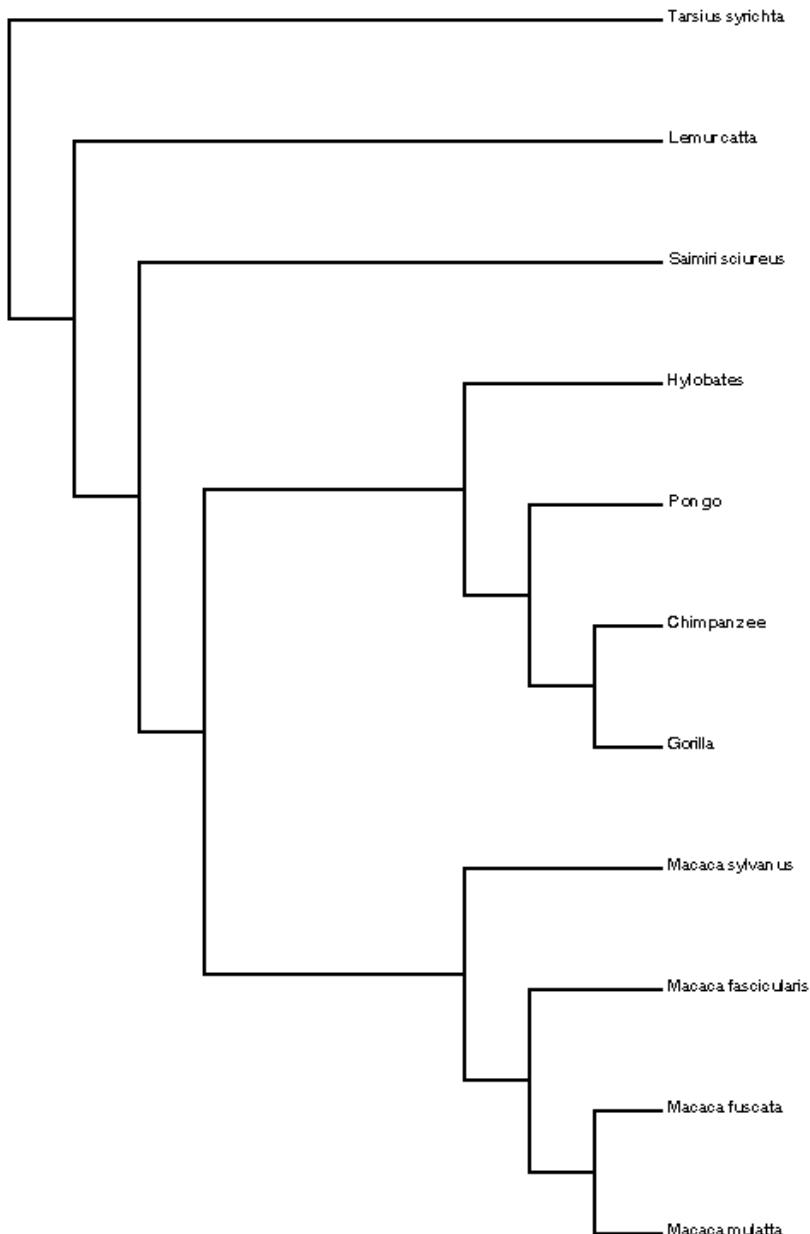


Fig. 18. Phylogenetic tree of the monkeys

The problem of phylogenetic inference from datasets including incomplete or uncertain entries is among the most relevant issues in systematic biology [292]. A method for reconstructing phylogenetic trees from partial distance matrices is

proposed. The new method combines the usage of the four-point condition and the ultrametric inequality with a weighted least-squares approximation to solve the problem of missing entries. It can be applied to infer phylogenies from evolutionary data, including some missing or uncertain information, *e.g.*, when observed nucleotide or protein sequences contain gaps or missing entries. In a number of simulations involving incomplete datasets, the proposed method outperformed the well-known Ultrametric and Additive procedures. Generally, the method also outperformed all the other competing approaches including Triangle and Fitch, which is the most popular least-squares method for reconstructing phylogenies. The usefulness of the introduced method is illustrated by analyzing two well-known phylogenies derived from complete mammalian mitochondrial DNA (mtDNA) sequences (*cf.* Figs. 15, 16). Some interesting theoretical results concerning the NP-hardness of the ordinary and weighted least-squares fitting of a phylogenetic tree to a partial distance matrix are also established [293–303].

Six new models to analyze the DNA sequences were constructed [304]. First, a DNA primary sequence was regarded as a random process in t and gave three ways to define nucleotides' random distribution functions. Some parameters were extracted from the linear model, and the changes of the nucleotides' distributions were analyzed. In order to facilitate the comparison of DNA sequences, two ways to measure their similarities were proposed. Finally, the six models were compared by analyzing the similarities of the presented DNA primary sequences, and the optimal one was selected (*cf.* Fig. 17).

Cartesian coordinates has been derived for mathematical denotation of DNA sequence [305]. The 3D graphical representation also avoids loss of information accompanying alternative two-dimensional (2D) and 3D representations in which the curve standing for DNA sequence overlaps and intersects itself, and resolves sequences' degeneracy. The examination of similarities–dissimilarities between the DNA sequences belonging to 11 species illustrates the utility of the approach. The elements of the similarity matrix are used to construct phylogenetic tree (*cf.* Fig. 18).

14 Phylogeny of Apes, Hominids and Man

Several workers have observed that there is an extremely close immunological resemblance between the serum albumins of apes and man [306]. The studies with the quantitative microcomplement fixation method confirm the observation. To explain the closeness of the resemblance, previous workers suggested that there has been a slowing down of albumin evolution since the time of divergence of apes and man. Recent evidence, however, indicates that the albumin molecule has evolved at a steady rate. Hence, it was suggested that apes and man have a more recent common ancestry than is usually supposed. The calculations lead to the suggestion that, if man and Old World monkeys last shared a common ancestor 30 million years ago (MYA), then man and African apes shared a common ancestor 5 MYA, *i.e.*, in the Pliocene era. The living hominoid primates are Man, the chimpanzees, the Gorilla, the Orangutan and the gibbons [307, 308]. Nucleotide

sequences of homologous 0.9-kb fragments of mtDNAs, derived from four species of old-world monkeys, one species of new-world monkeys and two species of prosimians, were determined [309]. With these nucleotide sequences and

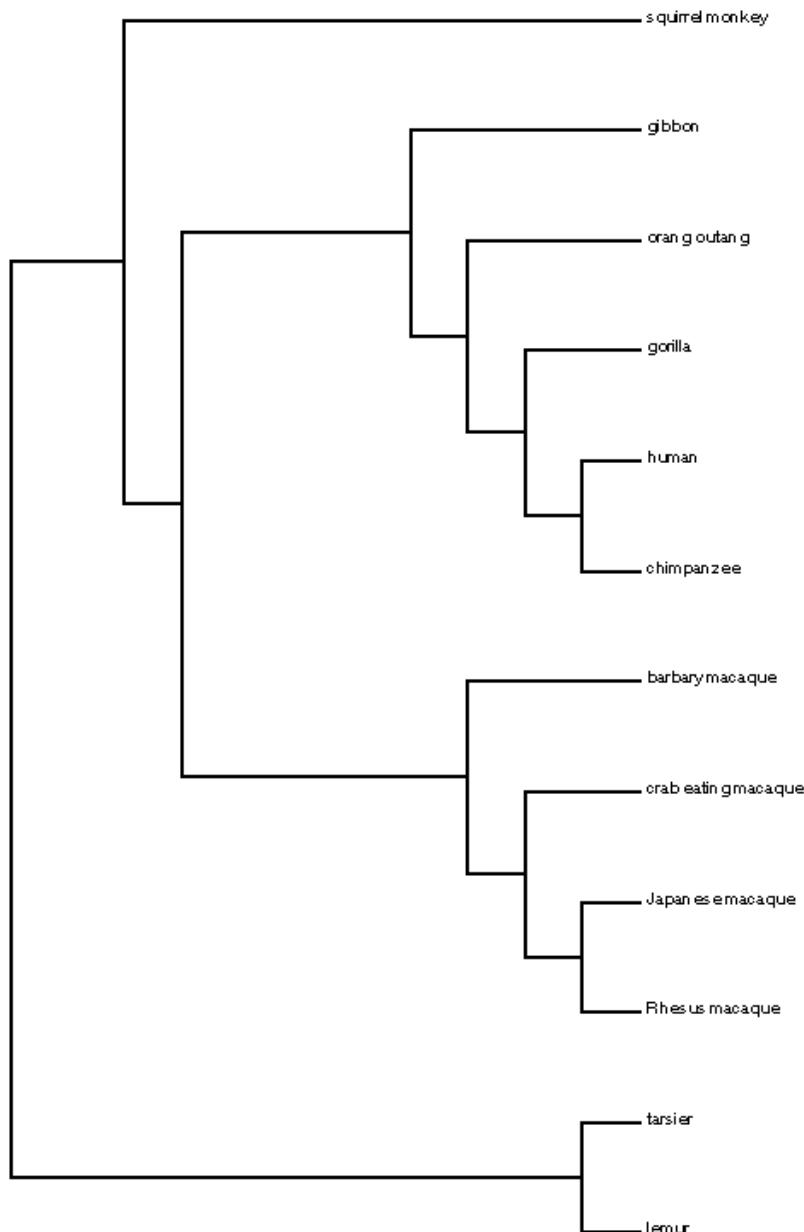


Fig. 19. Phylogenetic tree for 12 species of primates constructed by method NJ

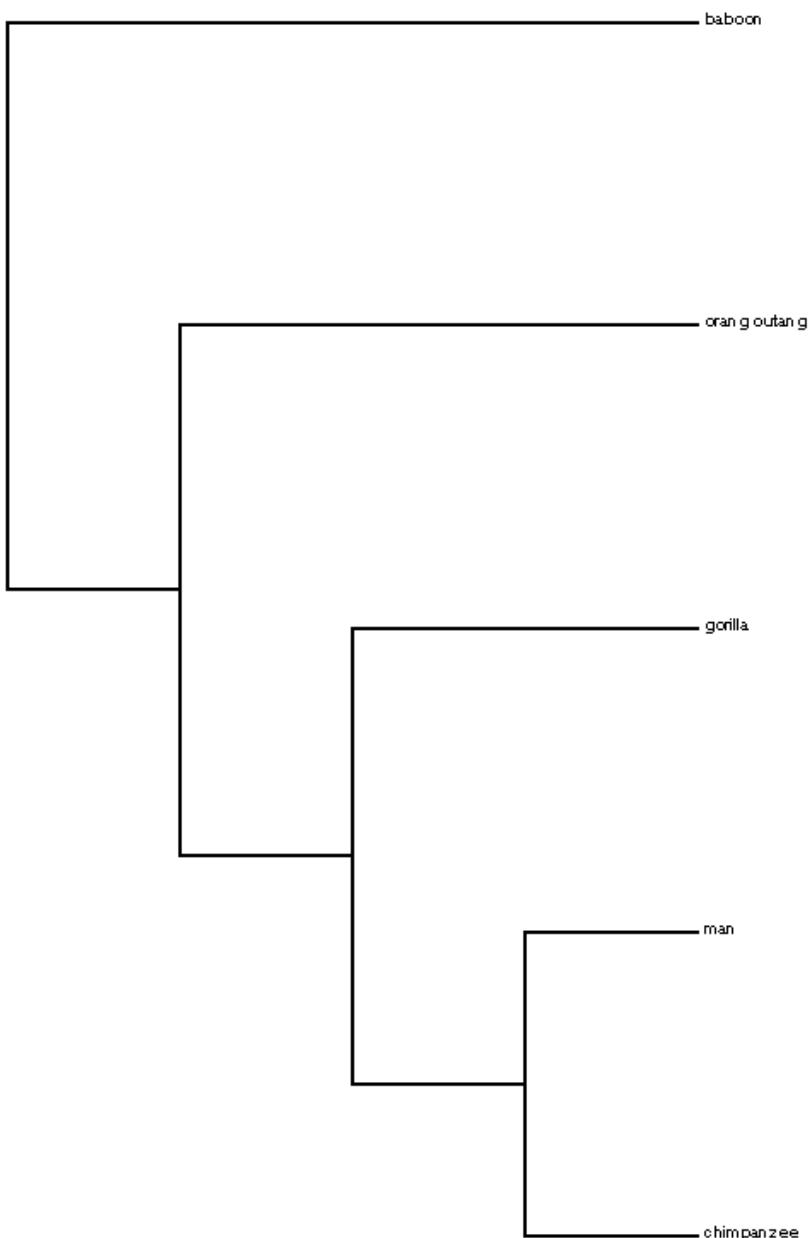


Fig. 20. Phylogenetic tree of the apes

homologous sequences for five species of hominoids, a phylogenetic tree for the four groups of primates was constructed. The obtained phylogeny is generally consistent with evolutionary trees constructed in previous studies (*cf.* Fig. 19). The results also suggest that the rate of nucleotide substitution for mtDNAs in

hominines (human, chimpanzee and gorilla) may have slowed down compared with that for old-world monkeys. This evolutionary feature of mitochondrial genes is similar to one found in nuclear genes.

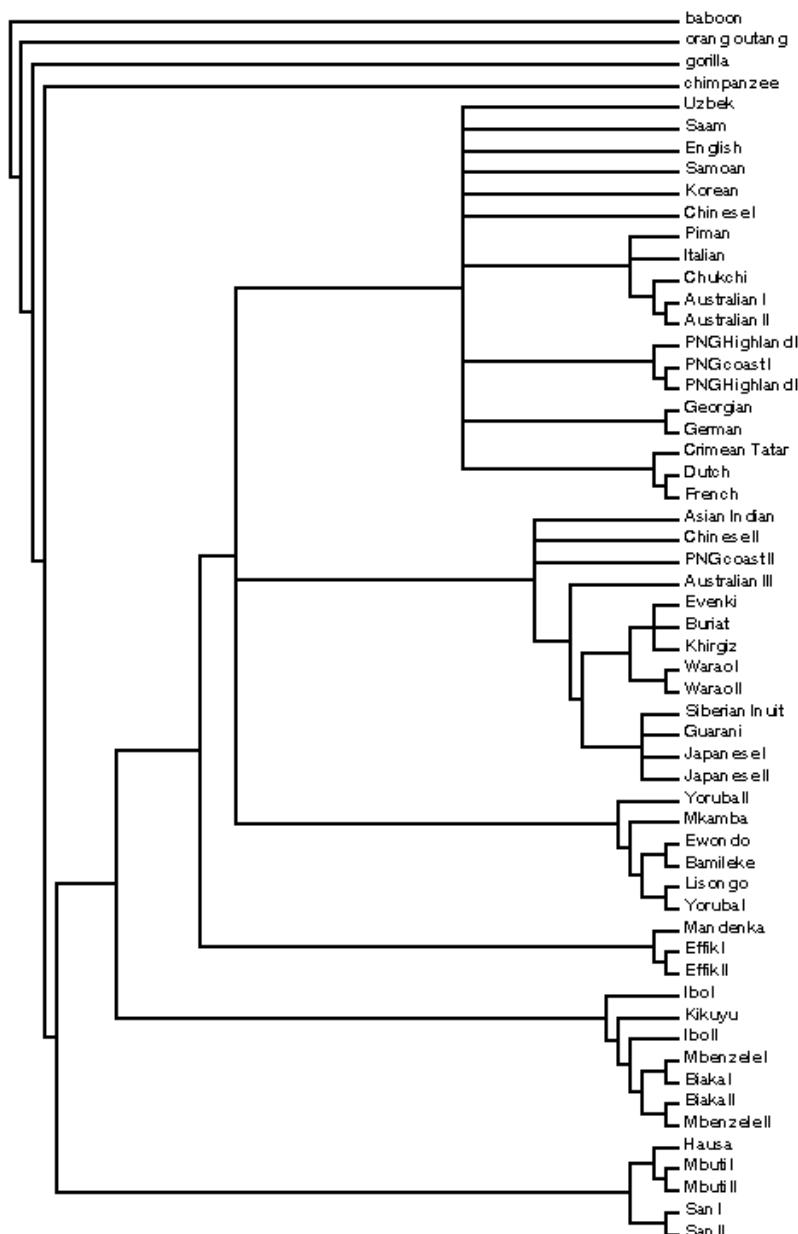


Fig. 21. Neighbour-joining phylogram based on complete mtDNA genome

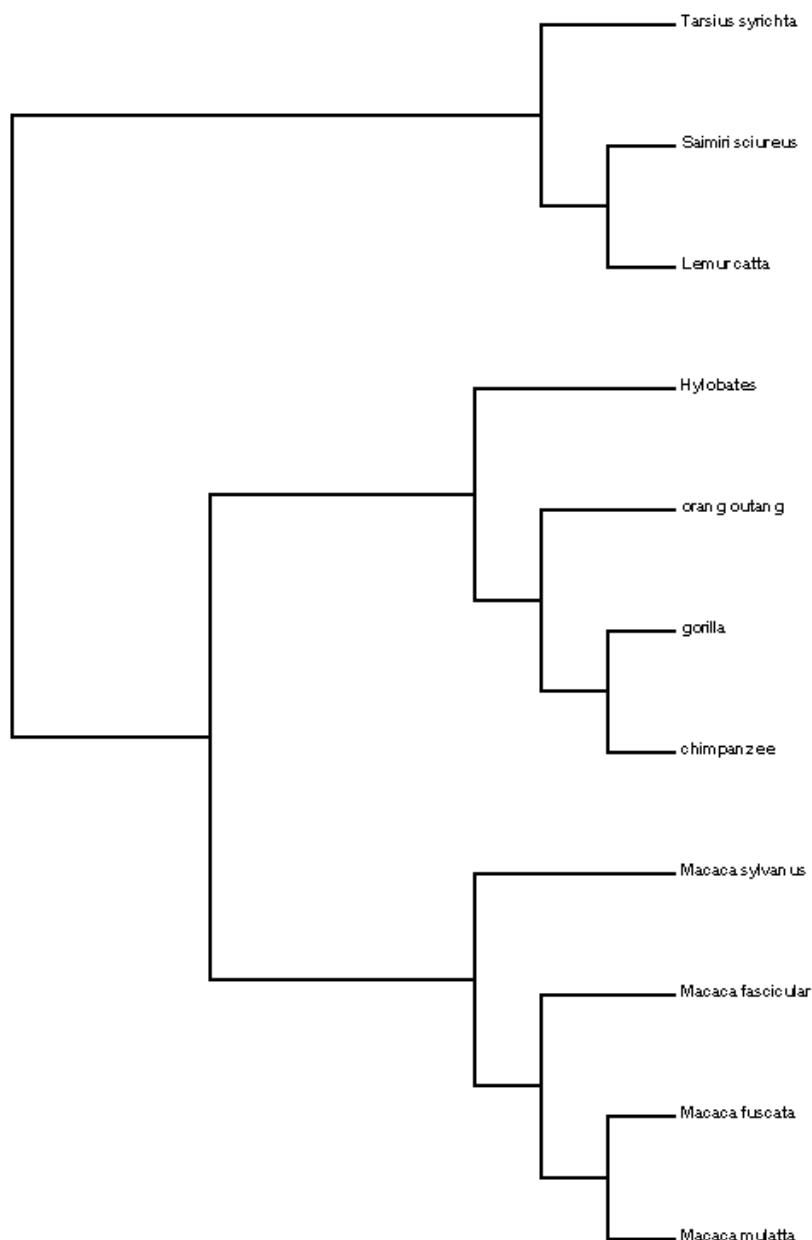


Fig. 22. Phylogenetic tree based on 3DD-Curve

The cercopithecoids (Old World monkeys) are the sister group of the hominoids [310–315]. The composition of the Hominoidea is not in dispute, but a consensus has not yet been reached concerning the phylogenetic branching pattern and the

dating of divergence nodes (*cf.* Fig. 20) [316]. It has been compared the single-copy nuclear DNA sequences of the hominoid genera using DNA–DNA hybridization to produce a complete matrix of delta T_{50H} values [317]. The data show that the branching sequence of the lineages, from oldest to most recent, was: Old World monkeys, gibbons, Orangutan, Gorilla, chimpanzees and Man. The calibration of the delta T_{50H} scale in absolute time needs further refinement, but the ranges of the estimates of the datings of the divergence nodes are: Cercopithecoidea, 27–33 MYA; gibbons, 18–22 MYA; Orangutan, 13–16 MYA; Gorilla, 8–10 MYA and chimpanzees–man, 6.3–7.7 MYA.

The analysis of mtDNA has been a potent tool in our understanding of human evolution, owing to characteristics, *e.g.*, high copy number, apparent lack of recombination, high substitution rate and maternal mode of inheritance [318]. However, almost all studies of human evolution based on mtDNA sequencing have been confined to the control region, which constitutes less than 7% of the mitochondrial genome. These studies are complicated by the extreme variation in substitution rate between sites, and the consequence of parallel mutations causing difficulties in the estimation of genetic distance and making phylogenetic inferences questionable. Most comprehensive studies of the human mitochondrial molecule have been carried out *via* restriction-fragment length polymorphism analysis, providing data that are ill suited to estimations of mutation rate and, therefore, the timing of evolutionary events. To improve the information obtained from the mitochondrial molecule for studies of human evolution, the global mtDNA diversity in humans is described based on analyses of the complete mtDNA sequence of 53 humans of diverse origins (*cf.* Fig. 21). The mtDNA data, in comparison with those of a parallel study of the Xq13.3 region in the same individuals, provide a concurrent view on human evolution with respect to the age of modern humans.

A novel method for phylogenetic analysis of DNA sequence data has been proposed [319]. At first, a new distance matrix of DNA sequence based on the 3DD-Curves is provided and new similarities/dissimilarities matrix is constructed by using this distance matrix of DNA sequence. As application, a phylogenetic tree for 11 species of primates was constructed (*cf.* Fig. 22). The obtained phylogeny is generally consistent with evolutionary trees constructed in previous studies

15 Phylogenesis of Extinct Species

Two groups of flightless ratite birds existed in New Zealand during the Pleistocene: the kiwis and the moas [320]. The latter are now extinct but formerly included 11 species. It has been enzymatically amplified and sequenced *ca.* 400 base pairs of the mitochondrial 12S rRNA gene from bones and soft tissue remains of four species of moas as well as eight other species of ratite birds and a tinamou. Contrary to expectation, the phylogenetic analysis shows that the kiwis are more closely related to Australian and African ratites than to the moas (*cf.* Fig. 23). Thus, New Zealand probably was colonized twice by ancestors of ratite birds.

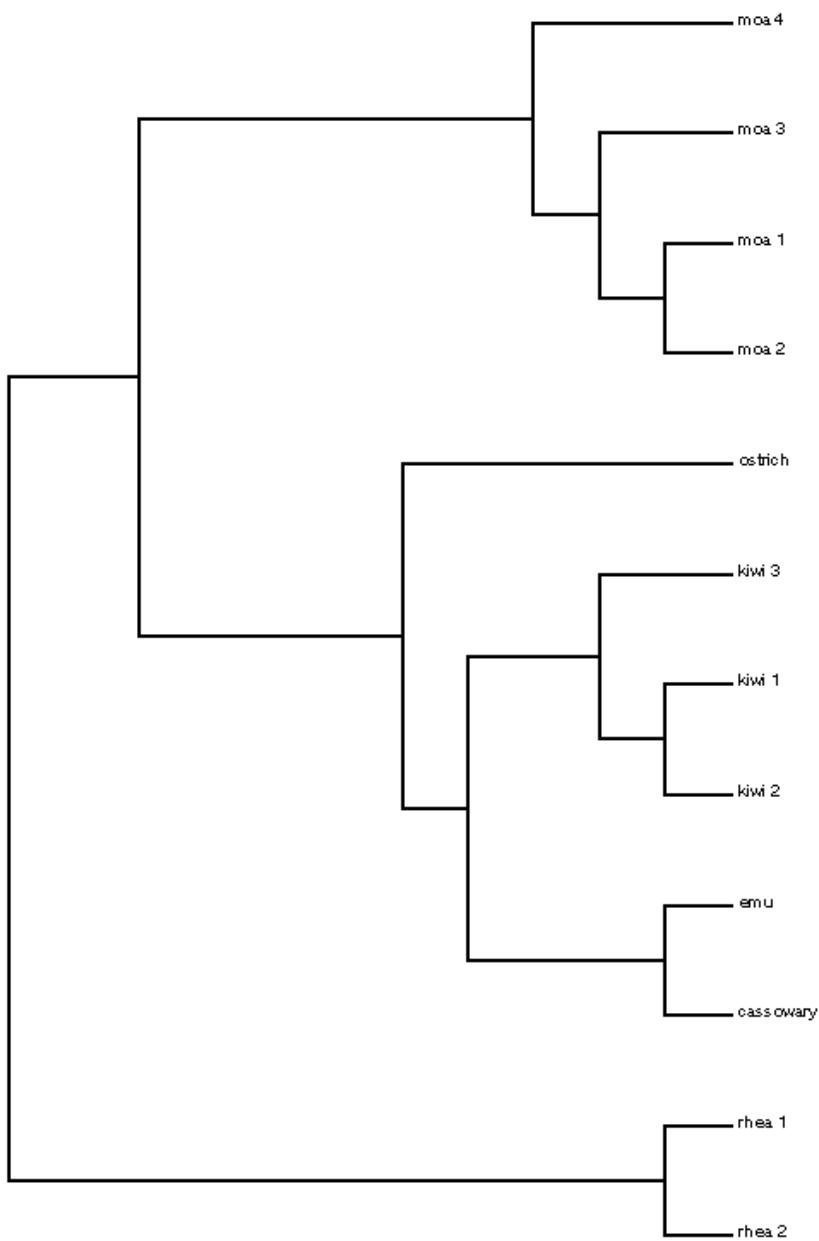


Fig. 23. Phylogenetic tree of the flightless ratite birds

The origin and evolution of the genus *Homo* was studied [321, 322]. DNA was extracted from the Neanderthal-type specimen found in 1856 in western Germany [323]. By sequencing clones from short overlapping PCR products, a

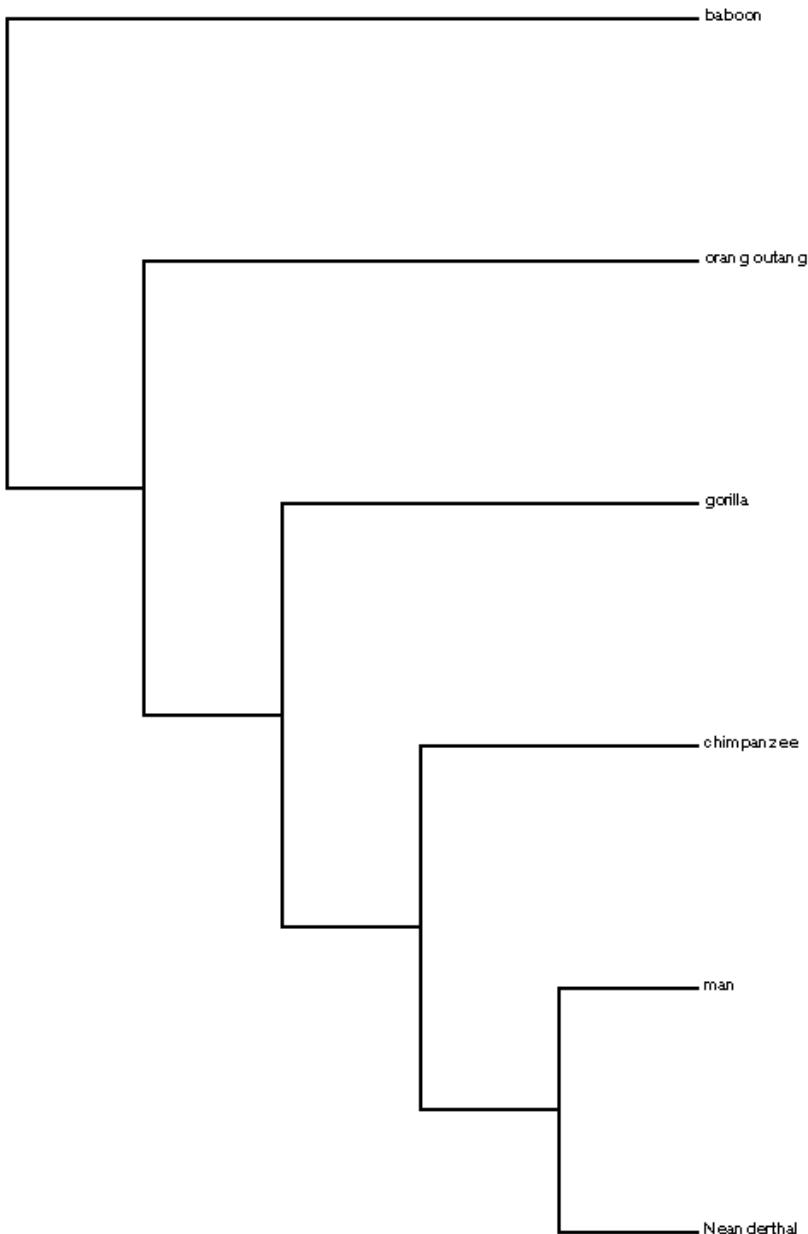


Fig. 24. Phylogenetic tree of the apes showing Neanderthal

hitherto unknown mitochondrial (mt) DNA sequence was determined. Multiple controls indicate that the sequence is endogenous to the fossil. Sequence comparisons with human mtDNA sequences, as well as phylogenetic analyses, show that the Neanderthal sequence falls outside the variation of modern humans (*cf.* Fig. 24).

Furthermore, the age of the common ancestor of the Neanderthal and modern human mtDNAs is estimated to be four times greater than that of the common ancestor of human mtDNAs. This suggests that Neanderthals went extinct without contributing mtDNA to modern humans.

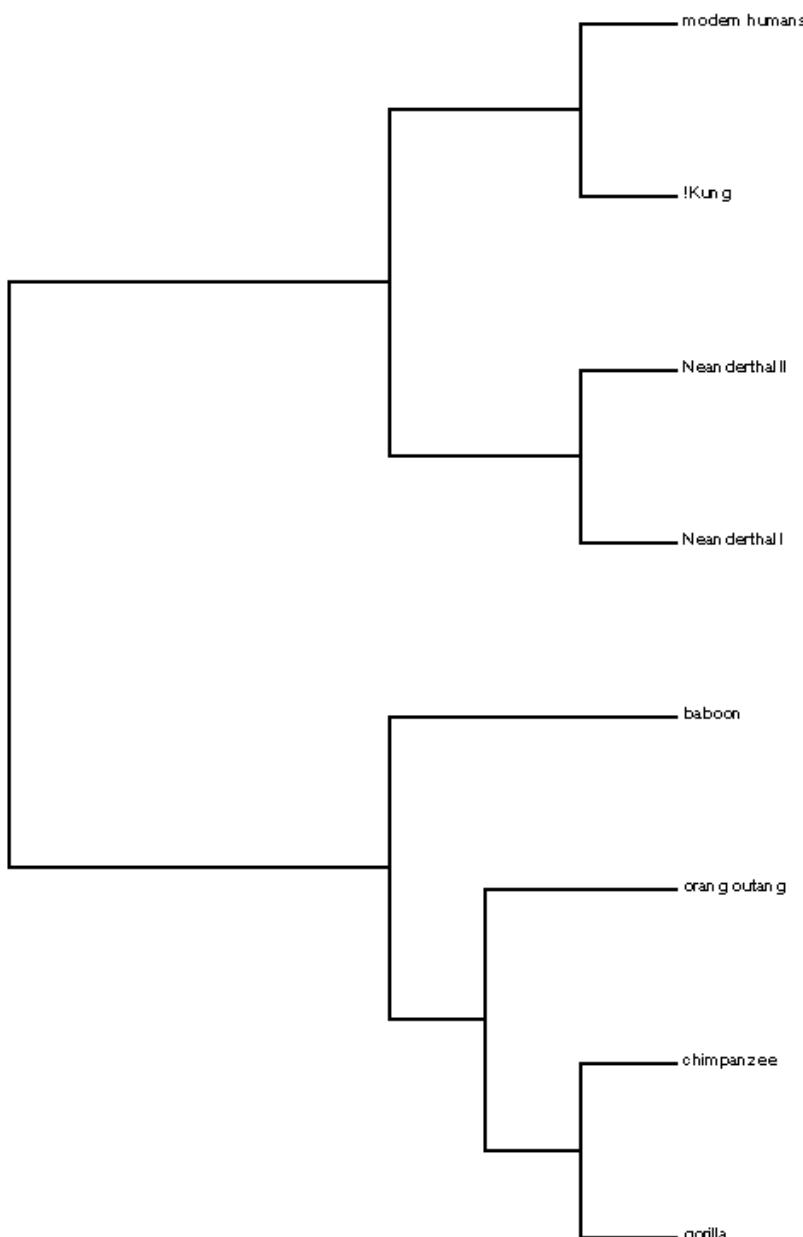


Fig. 25. Phylogenetic relationship of the two Neanderthals and modern humans

The expansion of premodern humans into western and eastern Europe *ca.* 40 000 years before the present led to the eventual replacement of the Neanderthals by modern humans *ca.* 28 000 years ago [324]. The second mtDNA analysis of a Neanderthal, and the first such analysis on clearly dated Neanderthal remains are reported. The specimen is from one of the eastern-most Neanderthal populations, recovered from Mezmaiskaya Cave in the northern Caucasus. Radiocarbon dating estimated the specimen to be *ca.* 29 000 years old and therefore from one of the latest living Neanderthals. The sequence shows 3.48% divergence from the Feldhofer Neanderthal. Phylogenetic analysis places the two Neanderthals from the Caucasus and western Germany together in a clade that is distinct from modern humans, suggesting that their mtDNA types have not contributed to the modern human mtDNA pool (*cf.* Fig. 25). Comparison with modern populations provides no evidence for the multiregional hypothesis of modern human evolution [325].

16 Calculation Results and Discussion

In the present report 31 HIV-1 inhibitors (Table 1) have been studied. Many of them fit the following general scheme: (base derivative)–(furan ring) since, among the species used in practice of HIV inhibition, these are the most numerous and have the widest range of uses. The base portion is often a guanine (Gua) or cytosine (Cys) derivative; the furan ring normally contains one O heteroatom. It has been calculated the Pearson correlation coefficient matrix, between the pairs of vector properties $\langle i_1, i_2, i_3, i_4, i_5 \rangle$ of the 31 inhibitors. The Pearson intercorrelations are illustrated in the partial correlation diagram, which contains high ($r \geq 0.75$), medium ($0.50 \leq r < 0.75$) and low ($0.25 \leq r < 0.50$) partial correlations. Pairs of inhibitors with high partial correlations show a similar vector property. However, the results should be taken with care, because the three compounds with constant $\langle 11111 \rangle$ vector (Entries 12, 16 and 30) show null standard deviation, causing high partial correlations ($r = 1$) with any inhibitor, which is an artifact. With the equipartition conjecture the intercorrelations are illustrated in the partial correlation diagram, which contains 157 high (*cf.* Fig. 26, red), 140 medium (orange) and 88 low (yellow) partial correlations. Notice that 84 out of the 90 ($3 \times 28/30$) high partial correlations of Entries 12–16–30 have been corrected; *e.g.*, for Entry 12 the correlations with Entries 2, 9, 10 and 28 are medium, and its correlations with Entries 4, 8, 13, 14, 15, 19, 20 and 21 are low partial correlations. The grouping rule in the case with equal weights $a_k = 0.5$ for $0.82 \leq b_1 \leq 0.87$ allows the classes: $C_{b_1} = (1, 3, 6, 7, 11, 17, 18, 22, 23, 25, 26, 29)(2, 28)(4, 8, 13, 14, 20)(5, 24, 27, 31)(9, 10)(12, 16, 30)(15, 19, 21)$

The seven classes are obtained with associated entropy $h(\bar{\mathbf{R}}_{b_1}) = 24.18$. Dendrogram and radial tree [326–329] matching to $\langle i_1, i_2, i_3, i_4, i_5 \rangle$ and C_{b_1} are calculated [330]; they provide a binary taxonomy of Table 1, which separates the same seven classes. In particular ddI, novel proposed ligand and mozenavir are grouped into the same class, and emivirine, SJ-3366, ddC, d4T and DAPD. Those inhibitors be

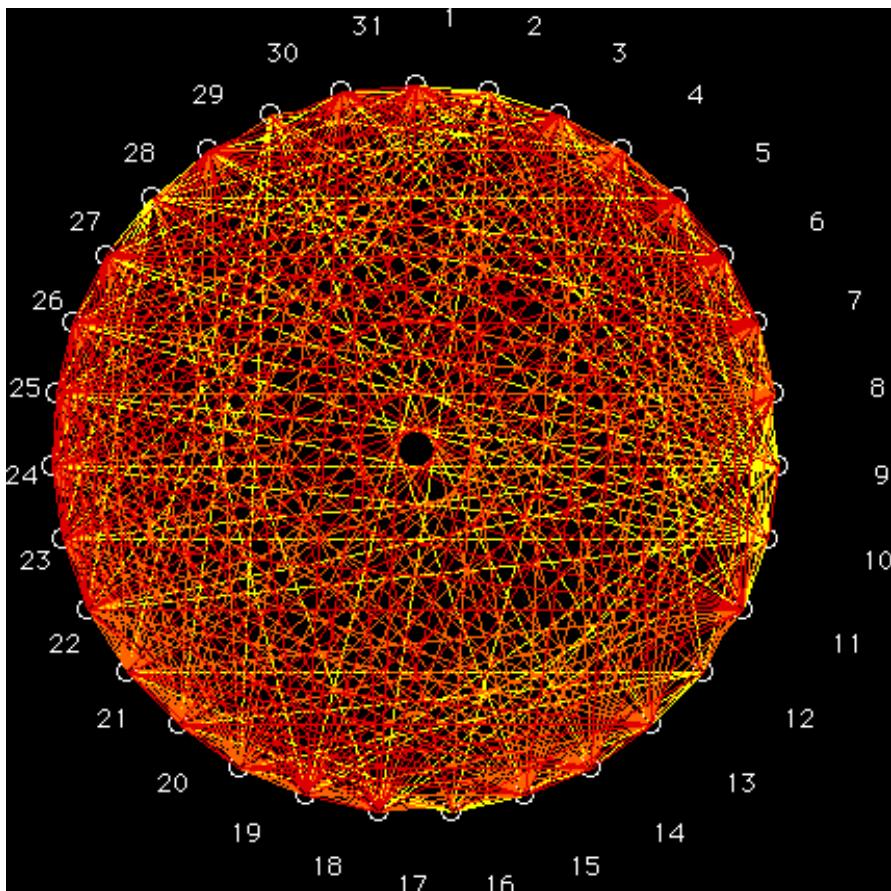


Fig. 26. Partial correlation diagram: High (red), medium (orange), low (yellow)

longing to the same class appear highly correlated in both partial correlation diagrams, in agreement with previous results obtained for Entries 1–3, 12–18, 23, 27 and 28.

At level b_2 with $0.76 \leq b_2 \leq 0.81$ the set of classes turns out to be:

$$C_{b_2} = (1, 3, 5, 6, 7, 11, 17, 18, 22, 23, 25, 26, 27, 29, 31)(2, 9, 10, 28)(4, 8, 13, 14, 15, 19, 20, 21) \\ (12, 16, 30)(24)$$

Five classes result in this case; the entropy decreases to $h(\bar{\mathbf{R}}_{b_2}) = 11.45$.

Both dendograms (*cf.* Fig. 27) matching to $\langle i_1, i_2, i_3, i_4, i_5 \rangle$ and C_{b_2} separate the same five classes, in agreement with both partial correlation diagrams (Fig. 26), binary trees and previous results (Entries 1–3, 12–18, 23, 27, 28). High similarity is found for Entries 2–28 (nevirapine and lopinavir), 3–11–17–18–29 (delavirdine, (+)-calanolide A, AZT, ABC and atazanavir), 12–16–30 (ddI, novel proposed ligand and mozenavir), and 4–8–13–14–20 (emivirine, SJ-3366, ddC, d4T and DAPD). The ddI, novel proposed ligand and mozenavir are grouped into the same

class, and emivirine, SJ-3366, ddC, d4T and DAPD. Entries 12–16–30 (ddI, novel proposed ligand and mozenavir) belong to the same class at any grouping level *b*, except at highest level in which each class contains one species.

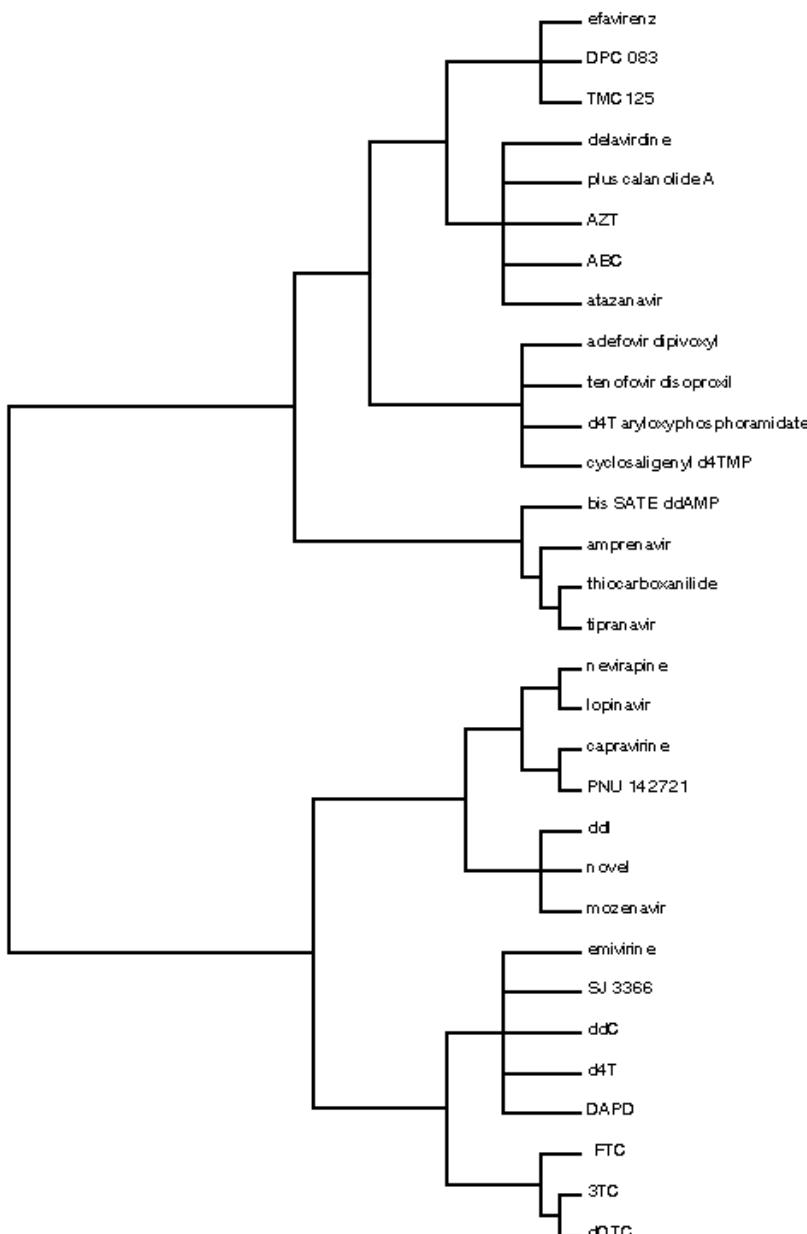


Fig. 27. Dendrogram for human immunodeficiency virus type 1 inhibitors

The analysis in 1–31 classes is in agreement with previous results obtained for Entries 1–3, 12–18, 23, 27 and 28 (Table 1). For 1–31 classes, the radial tree matching to $\langle i_1, i_2, i_3, i_4, i_5 \rangle$ and $C_{b_{1-31}}$ separates the same five and seven classes, in agreement with both partial correlation diagrams (Fig. 26), dendograms, radial trees and previous results. The inclusion in the radial tree (Fig. 27) is in agreement with both partial correlation diagrams (Fig. 26), etc. Once more ddI, novel proposed ligand and mozenavir are grouped into the same class, and emivirine, SJ-3366, ddC, d4T and DAPD. SplitsTree is a program for analyzing CA data [331]. Based on the method of *split decomposition*, it takes as input a *distance matrix* or a set of CA data and produces as output a graph, which represents the relationships between the taxa. For ideal data this graph is a tree whereas less ideal data will give rise to a tree-like network, which can be interpreted as possible evidence for different and conflicting data. Furthermore, as split decomposition does not attempt to force data onto a tree, it can provide a good indication of how tree-like given data are. The splits graph for the 31 HIV-1 inhibitors (Table 1) reveals no conflicting relationship (*cf.* Fig. 28). Compounds 12, 16 and 30 appear superposed. The splits graph is in general agreement with partial correlation diagrams, etc. (Figs. 23, 24).

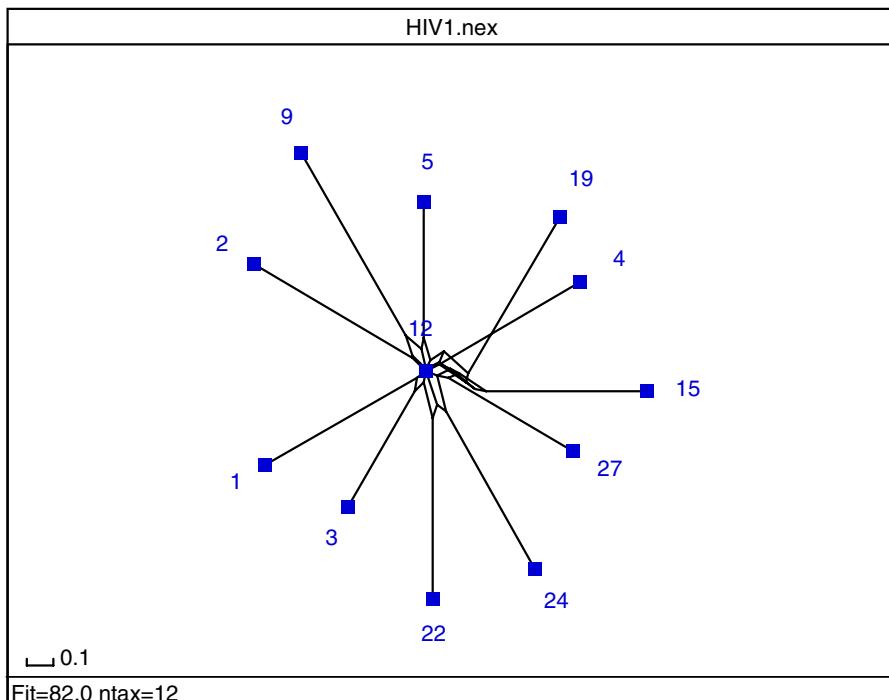


Fig. 28. Splits graph for the human immunodeficiency virus type 1 inhibitors

In the PCA [332] F_2 - F_1 plot (*cf.* Fig. 29), those HIV-1 inhibitors with the same vector property appear superposed. In particular, inhibitors 13, 14 and 20 (class NRTI) come out placed over compounds 4 and 8 (class NNRTI), inhibitors 17 and 18 (class NRTI), and 29 (class PI), over substances 3 and 11 (class NNRTI), inhibitor 30 (class PI), over molecules 12 and 16 (class NRTI), inhibitor 28 (class PI), over drug 2 (class NNRTI), and inhibitor 31 (class PI), over medicine 5 (class NNRTI). Four classes of inhibitors are clearly distinguished in agreement with the classical classification, *viz.* class NNRTI with 11 units ($F_1 < F_2 \approx 0$, *left*), class NRTI (10 units, $0 \approx F_1 < F_2$, *top*), class NtRTI (5 units, $F_1 > F_2$, *right*), and class PI (5 units, $0 \approx F_1 < F_2$, *right top*). The classification is in agreement with the partial correlation diagrams, dendrograms, radial trees and splits graph (Figs. 23–25).

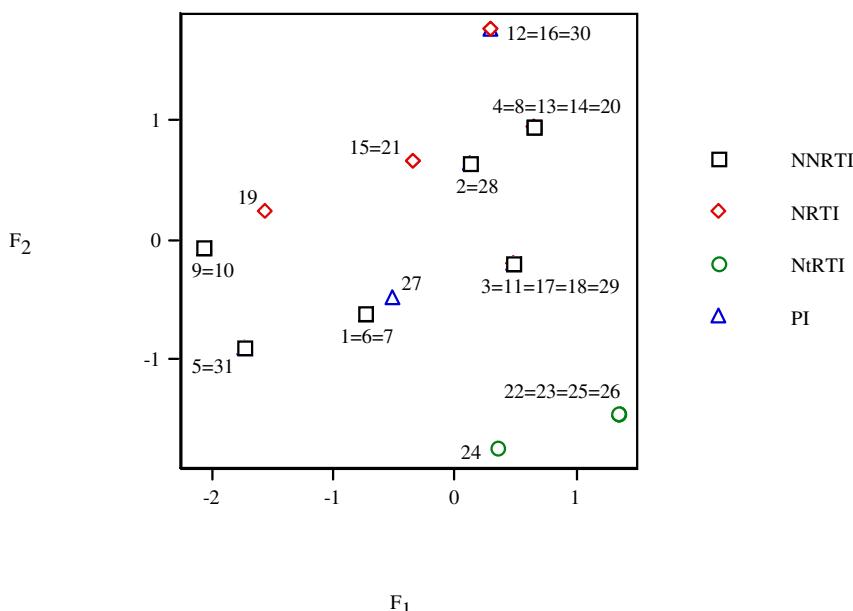


Fig. 29. Principal component analysis F_2 vs. F_1 plot for the HIV-1 inhibitors

Instead of 31 HIV-1 inhibitors in the \Re^5 space of 5 vector properties, consider 5 properties in the \Re^{31} space of 31 inhibitors. The dendrogram for the properties (*cf.* Fig. 30) separates first properties S_0 and X_0 , then, property N_4 and, finally, properties O_3 and P_0 .

The recommended format for the PT of the HIV-1 inhibitors (*cf.* Table 5) shows that they are classified first by i_5 , then by i_4 , i_3 , i_2 and, finally, by i_1 . Periods of six units are assumed; *e.g.* group g000 stands for $\langle i_1, i_2, i_3 \rangle = \langle 000 \rangle$, *viz.* $\langle 000001 \rangle$ [bis(SATE)ddAMP], $\langle 00010 \rangle$ (thiocarboxanilide, tipranavir), and

<00011> (amprenavir), etc. Those inhibitors in the same column appear close in both partial correlation diagrams, dendrograms, radial trees, splits graph and PCA (Figs. 23–26).

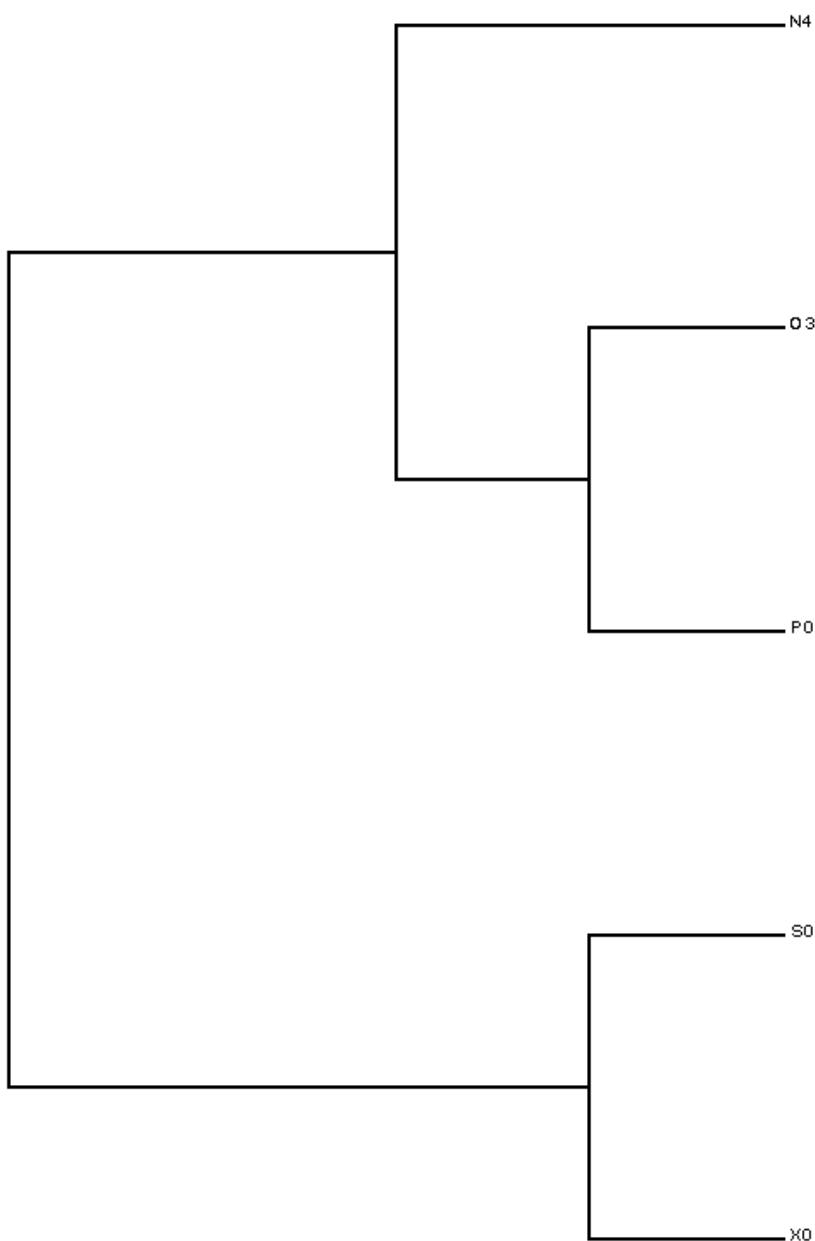


Fig. 30. Dendrogram for the vector properties corresponding to HIV-1 inhibitors

Table 5. Periodic properties for HIV-1 inhibitors

g000	g001	g010
bis(SATE)ddAMP	adefovir dipivoxyl tenofovir disoproxil d4T arylxyphosphoramide <i>cyclosaligenyl d4TMP</i>	
thiocarboxanilide tipranavir	efavirenz DPC 083 TMC125	(-)-FTC
amprenavir	delavirdine (+)-calanolide A AZT ABC atazanavir	3TC dOTC
g011	g100	g101
	capravirine PNU-142721	g111
emivirine SJ-3366 ddC d4T DAPD		nevirapine lopinavir ddI novel mozenavir

The variation of vector property $P = \langle i_1, i_2, i_3, i_4, i_5 \rangle$, as a function of the structural parameters $\{i_1, i_2, i_3, i_4, i_5\}$ for the HIV-1 inhibitors (*cf.* Fig. 31), shows that the line for the structural parameter i_4 appears superposed to i_3 , and that for the structural parameter i_5 , to i_1 , which agree with a PT of properties with vertical groups defined by $\{i_1, i_2, i_3\}$ and horizontal periods described by $\{i_4, i_5\}$.

The variation of vector property $P = \langle i_1, i_2, i_3, i_4, i_5 \rangle$, as a function of the number of the group in PT (*cf.* Fig. 32) for the HIV-1 inhibitors, reveals that the minima correspond to inhibitors with $\langle i_1, i_2, i_3 \rangle \text{ ca. } <000>$ (group g000). The $P(i_1, i_2, i_3, i_4, i_5)$ corresponding function reveals a series of *waves* clearly limited by maxima or minima, which suggest a periodic behaviour that recalls the form of a trigonometric function. For $\langle i_1, i_2, i_3, i_4, i_5 \rangle$ one minimum is clearly shown. The distance in $\langle i_1, i_2, i_3, i_4, i_5 \rangle$ units between each pair of consecutive minima is seven, which coincides with the inhibitor sets in the successive periods. The minima occupy analogous positions in the curve and are in phase. The representative points in phase should correspond to the elements of the same group in PT. For the $\langle i_1, i_2, i_3, i_4, i_5 \rangle$ minima there is coherence between both representations; however, the consistency is not general. Wave comparison shows two differences: (1) periods 1–2 are incomplete and (2) period 3 is somewhat sawtooth-like. Most characteristic points of the plot are minima, which lie about group g000. The values of $\langle i_1, i_2, i_3, i_4, i_5 \rangle$ are repeated as PL states.

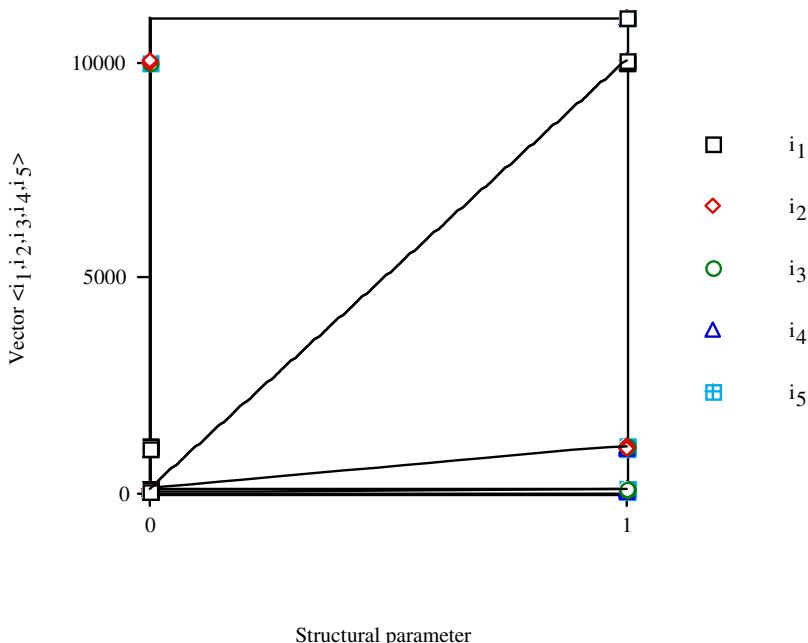


Fig. 31. Change of vector property of HIV-1 inhibitors vs. counts $\{i_1, i_2, i_3, i_4, i_5\}$

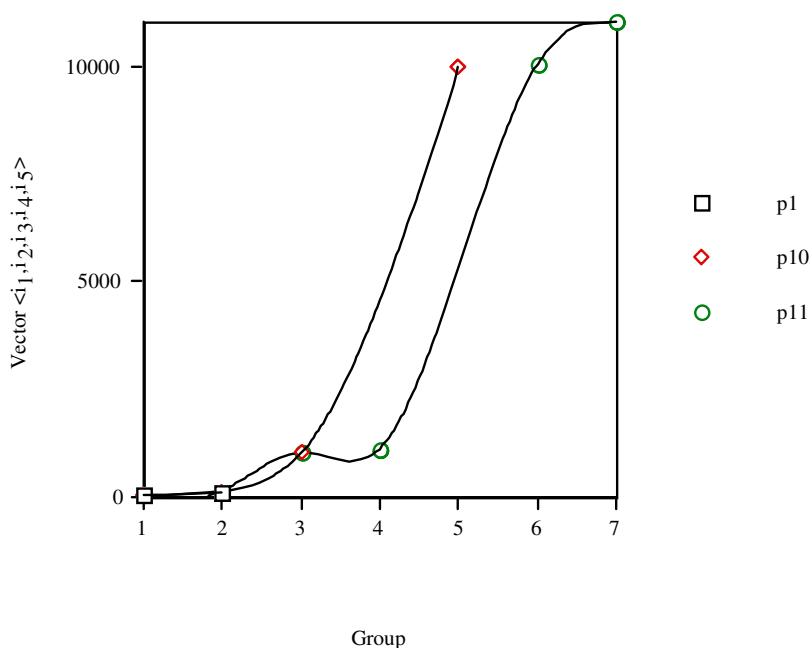


Fig. 32. Variation of vector property of the HIV-1 inhibitors vs. group number

An empirical function $P(p)$ reproduces the different $\langle i_1, i_2, i_3, i_4, i_5 \rangle$ values. A minimum of $P(p)$ has meaning only if it is compared with the former $P(p-1)$ and later $P(p+1)$ points, needing to fulfil:

$$\begin{aligned} P_{\min}(p) &< P(p-1) \\ P_{\min}(p) &< P(p+1) \end{aligned} \quad (8)$$

Relations (8) should repeat at determined intervals and are equivalent to:

$$\begin{aligned} P_{\min}(p) - P(p-1) &< 0 \\ P(p+1) - P_{\min}(p) &> 0 \end{aligned} \quad (9)$$

As relations (9) are valid only for minima more general others are desired for all the values of p . The $D(p) = P(p+1) - P(p)$ differences are calculated by assigning each of their values to inhibitor p :

$$D(p) = P(p+1) - P(p) \quad (10)$$

Instead of $D(p)$ the $R(p) = P(p+1)/P(p)$ values can be taken by assigning them to inhibitor p . If PL were general, the elements in the same group in analogous positions in different waves would satisfy:

$$D(p) > 0 \quad \text{or} \quad D(p) < 0 \quad (11)$$

$$R(p) > 1 \quad \text{or} \quad R(p) < 1 \quad (12)$$

However, the results show that this is not the case so that PL is not general existing some anomalies; *e.g.*, the variation of $D(p)$ vs. group number (*cf.* Fig. 33) presents lack of coherence between the $\langle i_1, i_2, i_3, i_4, i_5 \rangle$ Cartesian and PT representations. If consistency were rigorous all the points in each period would have the same sign. In general there is a trend in the points to give $D(p) > 0$ for the lower groups, and $D(p) < 0$, for the greater groups. In detail, however, there are irregularities in which the inhibitors for successive periods are not always in phase.

The change of $R(p)$ vs. group number (*cf.* Fig. 34) confirms the lack of constancy (Fig. 33) between the Cartesian and PT charts. If steadiness were exact all the points in each period (Fig. 34) would show $R(p)$ either lesser or greater than one. There is a trend in the points to give $R(p) > 1$ for the lower groups. Notwithstanding, there are confirmed incongruities (Fig. 33) in which the inhibitors for successive waves are not always in phase. The novel molecule does not violate Lipinski's Rule of Five [333–338], *i.e.*, molecular weight < 500 (395 g·mol⁻¹), H-bond acceptors ≤ 10 (3), H-bond donors ≤ 5 (1) and $\log P < 5$ (2.73).

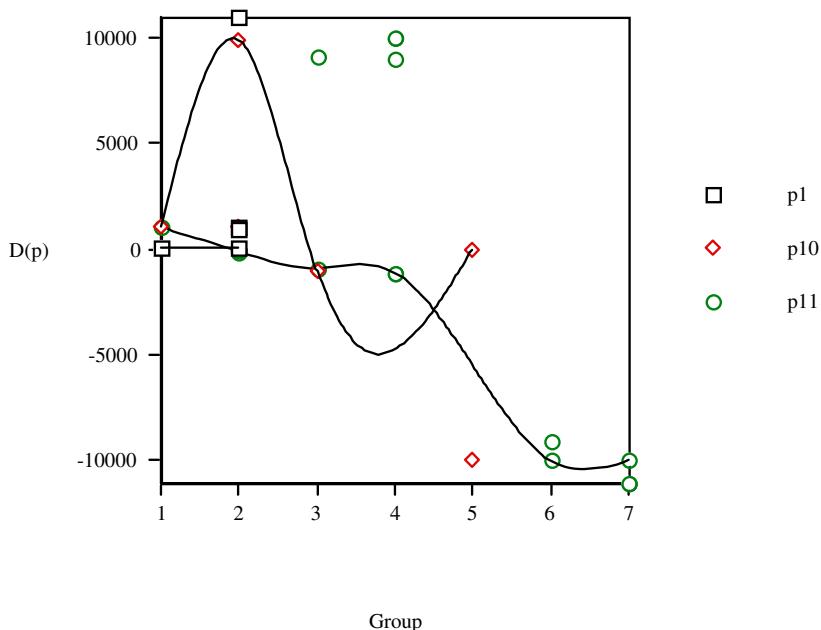


Fig. 33. Variation of $D(p) = P(p+1) - P(p)$ vs. group number

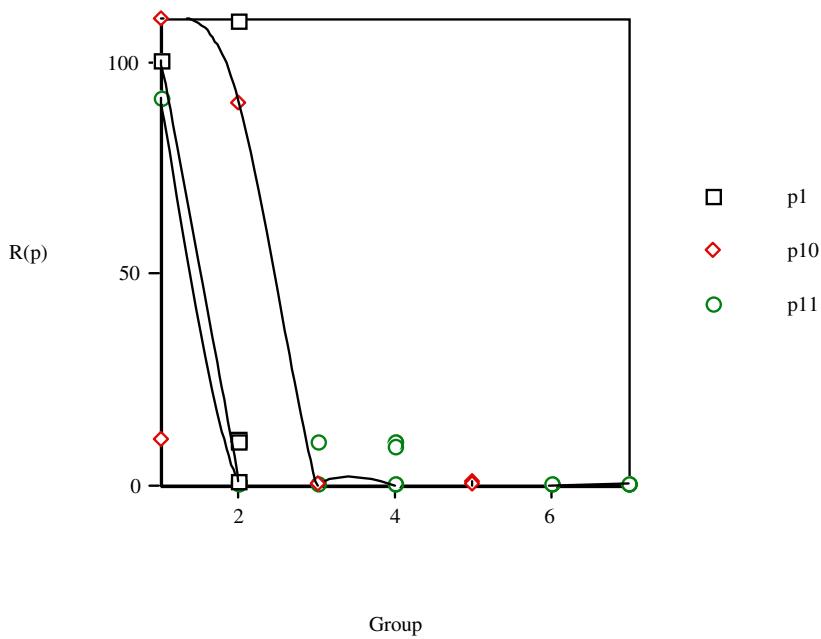


Fig. 34. Variation of $R(p) = P(p+1)/P(p)$ vs. group number

17 Perspectives

Many classification algorithms are based on *information entropy*. For sets of moderate size an excessive number of results appear compatible with data, and the number suffers a combinatorial explosion. However, after the *equipartition conjecture*, one has a selection criterion between different variants resulting from classification between hierarchical trees. According to the conjecture, the best configuration of a flowsheet is the one in which the entropy production is most uniformly distributed. The method avoids the problem of other methods of continuum variables, because for compounds with the same vector, the null standard deviation causes a Pearson correlation coefficient of one. The lower-level classification processes show lower entropy and may be more parsimonious. If, in the calculation of entropy associated with the phylogenetic tree, a species is systematically omitted, the difference between the entropy with and without this species can be considered as a measure of the species entropy. Such contributions may be studied with the equipartition conjecture. Obviously, it is not within the scope of our simulation method to replace biological tests of drugs or field data in palaeontology, but such simulation methods can be useful to assert priorities in detailed experimental research. Available experimental and field data should be examined by different classification algorithms to reveal possible features of real biological significance.

Program MolClas is a simple, reliable, efficient and fast procedure for molecular classification, based on the equipartition conjecture of entropy production. It has been written not only to analyze the equipartition conjecture of entropy production, but also to explore the world of molecular classification.

Clusters can form new stable structures, which can be the basis of new solids; they can be used for assembling new materials and be a basis for new technologies. One of the stable cluster structures with close packing is icosahedral, which is realized in various solids and is essential for various processes and interactions in systems with a short-range interaction of atoms. The main problem of large-cluster physics relates to their correspondence with macroscopic particles. Properties of large clusters differ from macroscopic ones. One would expect that a cluster, as a system intermediate between molecules and macroscopic particles, have properties of one of these depending on its size. A large cluster is an object with specific properties in a wide region of the number of atoms in it. Properties depend on cluster structure. Several criteria have been selected to reduce the analysis to a manageable quantity of structures from the enormous set of fullerene isomers. They refer to the structural parameters related with the presence of contiguous pentagons $\{p,q,r\}$, which destabilize the structures, and hexagons $\{u,v,w\}$. Considering the structure of adjacent hexagons three parameters u , v and w are used. A simple linear correlation is a good model for the permanent of the adjacency matrix of fullerenes; $\{q,r,v,w\}$ is redundant information. The $\{p,u\}$ contains the essential characters of the permanent for fullerene structures. The method allows rapid estimation of the permanent for large fullerenes. Linear methods require that fewer parameters be estimated and may be more parsimonious. A discussion of fullerene aromatic character is problematic because of

difficulty in choosing a *benchmark* molecule. Use of planar conjugated molecules as primary reference points demonstrates a failure to capture the significance of the synthesis of fullerenes, for organic chemistry. If C₆₀ is not to be considered aromatic benzene will be condemned to a lonely existence, which makes benzene a plausible choice as a building block for carbon nanostructures.

In accordance with the similarity laws, the dimensionless Debye temperatures θ_0 for all crystals belonging to the considered class should be close. Temperatures θ_0 are determined *via* similarity relation from experimental and estimated data. Fullerite θ_0 is twice that for inert-gas crystals, which is due to the fact that near the Debye point the crystal is orientationally ordered so that its structure is dissimilar to face-centred cubic. A fullerene molecule whose thermal rotation is frozen cannot be considered as a spherically symmetric particle. The fulfilment of the similarity laws, which are valuable for particles with spherically symmetric interaction potential, would hardly be expected; some contribution to the Debye spectrum of fullerite is made by intramolecular vibrations, which are not considered in formulating similarity laws. If intramolecular and intermolecular vibration frequencies in the Debye temperature range are of the same order, this should be reflected in the Debye frequency.

Calculated elementary polarizability relationships of any SWNT (*n,m*) is similar to that of its neighbour (*n-1,m+1*). The trend is approximately repeated for each period. The (9,0) and (5,5), which join smoothly to a C₆₀ hemisphere, are the smallest diameter SWNTs that can be properly capped.

The detailed comparison of the sequences (primary structures) of enzyme lysozyme has allowed for the reconstruction of a molecular phylogenetic tree for birds. Single- and complex-linkage perform a binary taxonomy of the parameters that separates avian birds by the following scheme with successive branchings: (1,...,5) → (1,4,5)(2,3) → (1,5)(2,3)(4) → (1)(2,3)(4)(5) → (1)(2)(3)(4)(5). Genetic analyses were applied in judicial investigations [339–341].

Topical anaesthetics remain a powerful, new advancement for minimizing pain during cutaneous procedures. While several new topical anaesthetic agents have been released that claim increased efficacy and faster onset, EMLA remains the most widely used topical anaesthetic given its proven efficacy and safety by several clinical trials. As options for practitioner continue to grow, the need for studies comparing onset of action, efficacy and safety continues to be of paramount importance. MolClas provides a way to classify the local anaesthetics for difficult cases that are hard to sort *a priori* (procaine–ice). EMLA and ice decrease the discomfort associated with needle injection. Although EMLA performs better in pain control, ice has advantages in easy of use, fast action and is less expensive than EMLA. EMLA and ice are good topical anaesthetics, each with advantages and disadvantages in clinical use.

The comparison 4-alkylanilines/phenyl alcohols shows that the smaller polar character of the former causes their less negative Gibbs solvation energy and greater hydrophobicity. Both series are distinguished by molecular *rugosity*. The correlations point not only to homogeneous molecular structures of the phenyl alcohols and of 4-alkylanilines, but also to the ability to predict and tailor drug properties. The latter is nontrivial in pharmacology.

Several criteria, selected to reduce the analysis to a manageable quantity of structures from the large set of HIV-1 inhibitors, refer to the structural parameters related with base derivative, *etc.* Good comparison of our classification results, with other taken as *good*, confirm the adequacy of the property vector selected for the molecular structures of HIV-1 inhibitors. Information entropy and principal component analyses permit classifying HIV-1 inhibitors and agree. The HIV-1 inhibitors are grouped into different classes. In general the classical classes of HIV-1 inhibitors are recognized. The final classification is shown more precise and with lower bias. Inhibitors are classified by structural chemical properties. The *structural elements* of an inhibitor can be *ranked* according to their inhibitory activity, in the order: number of N atoms > number of O atoms > number of S atoms > number of P atoms > number of halogens. The ddI contains four N atoms, *etc.* ($N_4O_3S_0P_0X_0$, X = F, Cl, Br); its associated vector is <11111>. It was selected as a *reference*. Most inhibitors contain no S atom (ddI, ddC, d4T, novel proposed ligand, $N_{3-4}O_3S_0P_0X_0$), while 3TC includes one S atom ($N_3O_3S_1P_0X_0$). Analysis is in agreement with principal component analysis. It compares well with other classification taken as *good* based on docking, density functional, molecular dynamics, the Rule of Five, and absorption, distribution, metabolism, excretion and toxicity. The analysis of the interactions of proposed novel ligand with reverse-transcriptase active site, *via* the Rule of Five, absorption, *etc.*, strongly suggests that proposed novel ligand could be a good potential inhibitor for anti-HIV chemotherapy. Periodic law has not the rank of the laws of physics: (1) properties of HIV-1 inhibitors are not repeated; perhaps their chemical character; (2) order relationships are repeated with exceptions. Analysis forces the statement: The relationships that any inhibitor p has with its neighbour $p + 1$ are approximately repeated for each period. Periodicity is not general; however, if a natural order of inhibitors is accepted the law must be phenomenological.

References

1. Da Silva, C.H.T.P., Carvalho, I., Taft, C.A.: Homology modeling and molecular interaction field studies of α glucosidases as a guide to structure-based design of novel proposed anti HIV inhibitors. *J. Comput. Aided Mol. Design* 19, 83–92 (2005)
2. Da Silva, C.H.T.P., del Ponte, G., Neto, A.F., Taft, C.A.: Rational design of novel diketoacid-containing ferrocene inhibitors of HIV-1 integrase. *Bioorg. Chem.* 33, 274–284 (2005)
3. Da Silva, C.H.T.P., Almeida, P., Taft, C.A.: Density functional and docking studies of retinoids for cancer treatment. *J. Mol. Model.* 10, 38–43 (2004)
4. Da Silva, C.H.T.P., Taft, C.A.: Molecular dynamics, database screening, density functional and docking studies of novel RAR ligands in cancer chemotherapy. *Biophys. Chem.* 117, 73–77 (2005)
5. Da Silva, C.H.T.P., Taft, C.A.: Computer-aided molecular design of novel glucosidase inhibitors for AIDS treatment. *J. Biomol. Struct. Dynam.* 22, 59–64 (2004)
6. Arissawa, M., Taft, C.A., Felcman, J.: Investigation of nucleoside analogs with anti HIV activity. *Int. J. Quantum. Chem.* 93, 422–432 (2003)

7. Kuno, M., Palangsuntikul, R., Hannongbua, S.: Investigation on an orientation and interaction energy of the water molecule in the HIV 1 reverse transcriptase active site by quantum chemical calculations. *J. Chem. Inf. Comput. Sci.* 43, 1584–1590 (2003)
8. Sharma, B., Kaushik, N., Singh, K., Kumar, S., Pandey, V.N.: Substitution of conserved hydrodynamic residues in motifs B and C of HIV-1 RT alters the geometry of its catalytic pocket. *Biochemistry* 41, 15685–15697 (2002)
9. De Clercq, E.: New developments in anti-HIV chemotherapy. *Biochim. Biophys. Acta* 1587, 258–275 (2002)
10. Kasai, N., Mizushima, Y., Sugawara, F., Sakaguchi, K.: Three-dimensional structural model analysis of the binding site of an inhibitor, nervonic acid, of both DNA polymerase β and HIV-1 reverse transcriptase. *J. Biochem.* 132, 819–828 (2002)
11. Painter, G.R., Andrews, C.W., Furman, P.A.: Conformation and local environment of nucleotides bound to HIV type 1 reverse transcriptase (HIV-1 RT) in the ground state. *Nucleosides Nucleotides Nucl. Acids* 19, 13–29 (2000)
12. Mlinaric, A., Kreft, S., Umek, A., Strukelj, B.: Screening of selected plant extracts for *in vivo* inhibitory activity on HIV-1 reverse transcriptase (HIV-1 RT). *Pharmazie* 55, 75–77 (2000)
13. Da Silva, C.H.T.P., Carvalho, I., Taft, C.A.: Molecular dynamics, docking, density functional, and ADMET studies of HIV-1 reverse transcriptase inhibitors. *J. Theor. Comput. Chem.* 5, 579–586 (2006)
14. Varmuza, K.: Pattern recognition in chemistry. Springer, New York (1980)
15. Benzecri, J.-P.: L'analyse des données. Dunod, Paris, vol. 1 (1984)
16. Tondeur, D., Kvaalen, E.: Equipartition of entropy production. An optimality criterion for transfer and separation processes. *Ind. Eng. Chem. Fundam.* 26, 50–56 (1987)
17. Torrens, F., Castellano, G.: Periodic table of local anaesthetics (procaine analogues). In: Seijas, J.A., Vázquez-Tato, M.P. (eds.) *Synthetic Organic Chemistry IX*, pp. 1–18. MDPI, Basel (2005)
18. Torrens, F., Castellano, G.: Periodic classification of local anaesthetics (procaine analogues). *Int. J. Mol. Sci.* 7, 12–34 (2006)
19. Torrens, F., Castellano, G.: Information entropy and the classification of local anaesthetics. In: Seijas, J.A., Vázquez-Tato, M.P. (eds.) *Synthetic Organic Chemistry XI*, pp. 1–17. MDPI, Basel (2007)
20. Torrens, F., Castellano, G.: Periodic classification of human immunodeficiency virus inhibitors. In: Seijas, J.A., Vázquez-Tato, M.P. (eds.) *Synthetic Organic Chemistry XI*, pp. 1–10. MDPI, Basel (2007)
21. Kaufmann, A.: Introduction à la théorie des sous-ensembles flous, vol. 3. Masson, Paris (1975)
22. Shannon, C.E.: A mathematical theory of communication: Part I, discrete noiseless systems. *Bell. Syst. Tech. J.* 27, 379–423 (1948)
23. Shannon, C.E.: A mathematical theory of communication: Part II, the discrete channel with noise. *Bell. Syst. Tech. J.* 27, 623–656 (1948)
24. White, H.: AI Expert 12, 48 (1989)
25. Iordache, O., Corriou, J.P., Garrido-Sánchez, L., Fonteix, C., Tondeur, D.: Neural network frames. Application to biochemical kinetic diagnosis. *Comput. Chem. Eng.* 17, 1101–1113 (1993)

26. Meneses, A., Rojas, L., Sifontes, R.S., López, Y., Sariego, R.I.: Aplicación de un método alternativo al conteo en cámara de Neubauer para determinar concentración de Trichomonas vaginalis. *Rev. Cubana Med. Trop.* 53, 180–188 (2001)
27. Marrero-Ponce, Y., Machado-Tugores, Y., Montero-Pereira, D., Escario, J.A., Gómez-Barrio, A., Nogal-Ruiz, J.J., Ochoa, C., Arán, V.J., Martínez-Fernández, A.R., García-Sánchez, R.N., Montero-Torrens, A., Torrens, F., Meneses-Marcel, A.: A computer-based approach to the rational discovery of new trichomonacidal drugs by atom-type linear indices. *Curr. Drug Discov. Technol.* 2, 245–265 (2005)
28. Meneses, A., Marrero-Ponce, Y., Machado, Y., Montero, A., Montero, D., Escario, J.A., Nogal, J.J., Ochoa, C., Arán, V.J., Martínez-Fernández, A.R., García, R.N.: A linear discrimination analysis based virtual screening of trichomonacidal lead-like compounds. Outcomes of in silico studies supported by experimental results. *Bioorg. Med. Chem. Lett.* 17, 3838–3843 (2005)
29. Montero, A., Vega, M.C., Marrero-Ponce, Y., Rolón, M., Gómez-Barrio, A., Escario, J.A., Arán, V.J., Martínez-Fernández, A.R., Meneses, A.: A novel non-stochastic quadratic fingerprints-based approach for the in silico discovery of new antitrypanosomal compounds. *Bioorg. Med. Chem.* 13, 6264–6275 (2005)
30. Montero-Torres, A., García-Sánchez, R.N., Marrero-Ponce, Y., Machado-Tugores, Y., Nogal-Ruiz, J.J., Martínez-Fernández, A.R., Arán, V.J., Ochoa, C., Meneses-Marcel, A., Torrens, F.: Non stochastic quadratic fingerprints and LDA-based QSAR models in hit and lead generation through virtual screening: Theoretical and experimental assessment of a promising method for the discovery of new antimalarial compounds. *Eur. J. Med. Chem.* 41, 483–493 (2006)
31. Vega, M.C., Montero-Torres, A., Marrero-Ponce, Y., Rolón, M., Gómez-Barrio, A., Escario, J.A., Arán, V.J., Nogal, J.J., Meneses-Marcel, A., Torrens, F.: New ligand-based approach for the discovery of antitrypanosomal compounds. *Bioorg. Med. Chem. Lett.* 16, 1898–1904 (2006)
32. Marrero-Ponce, Y., Meneses-Marcel, A., Castillo-Garit, J.A., Machado-Tugores, Y., Escario, J.A., Gómez-Barrio, A., Montero-Pereira, D., Nogal-Ruiz, J.J., Arán, V.J., Martínez-Fernández, A.R., Torrens, F., Rotondo, R., Ibarra-Velarde, F., Alvarado, Y.J.: Predicting antitrichomonal activity: A computational screening using atom-based bilinear indices and experimental proofs. *Bioorg. Med. Chem.* 14, 6502–6524 (2006)
33. Marrero-Ponce, Y., Meneses-Marcel, A., Rivera-Borroto, O.M., Montero, A., Escario, J.A., Gómez-Barrio, A., Montero-Pereira, D., Nogal, J.J., Grau, R., Torrens, F., Ibarra-Velarde, F., Rotondo, R., Alvarado, Y.J., Vogel, C., Rodriguez-Machin, L.: Quick access to potential trichomonacids through bond linear indices – Trained ligand-based virtual screening models. In: Seijas, J.A., Vázquez-Tato, M.P. (eds.) *Synthetic Organic Chemistry X*, pp. 1–41. MDPI, Basel (2006)
34. Meneses-Marcel, A., Rivera-Borroto, O.M., Marrero-Ponce, Y., Montero, A., Machado-Tugores, Y., Escario, J.A., Gómez-Barrio, A., Montero-Pereira, D., Nogal, J.J., Kouznetsov, V.V., Ochoa-Puentes, C., Bohórquez, A.R., Grau, R., Castañedo-Cancio, N., Torrens, F., Ibarra-Velarde, F., Rotondo, R., Alvarado, Y.J., Vogel, C., Rodriguez-Machin, L.: Bond-based quadratic TOMOCOMD-CARDD molecular indices and statistical techniques for new antitrichomonal drug-like compounds discovery. In: Seijas, J.A., Vázquez-Tato, M.P. (eds.) *Synthetic Organic Chemistry X*, pp. 1–53. MDPI, Basel (2006)

35. Marrero-Ponce, Y., Meneses-Marcel, A., Rivera-Borroto, O.M., García-Domenech, R., de Julián-Ortiz, J.V., Montero, A., Escario, J.A., Gómez-Barrio, A., Montero-Pereira, D., Nogal, J.J., Grau, R., Torrens, F., Vogel, C., Arán, V.J.: Bond-based linear indices in QSAR: Computational discovery of novel anti-trichomonal compounds. *J. Comput.-Aided Mol. Design* 22, 523–540 (2008)
36. Rivera-Borroto, O.M., Marrero-Ponce, Y., Meneses-Marcel, A., Escario, J.A., Gómez-Barrio, A., Montero-Pereira, D., Nogal, J.J., Torrens, F., Ibarra-Velarde, F., Rotondo, R., Alvarado, Y.J., Vogel, C.: Discovery of novel trichomonacidal using LDA-driven QSAR models and bond-based bilinear indices as molecular descriptors. *QSAR Comb. Sci.* (in press)
37. Torrens, F.: Computing the Kekulé structure count for alternant hydrocarbons. *Int. J. Quantum. Chem.* 88, 392–397 (2002)
38. Torrens, F.: Computing the permanent of the adjacency matrix for fullerenes. *Internet Electron. J. Mol. Des.* 1, 351–359 (2002)
39. Torrens, F.: Principal component analysis of structural parameters for fullerenes. *Internet Electron J. Mol. Des.* 2, 96–111 (2003)
40. Torrens, F.: Principal component analysis of new structural parameters for fullerenes. *Internet Electron J. Mol. Des.* 2, 546–563 (2003)
41. Torrens, F.: New structural parameters of fullerenes for principal component analysis. *Theor. Chem. Acc.* 110, 371–376 (2003)
42. Torrens, F.: Table of periodic properties of fullerenes based on structural parameters. *J. Chem. Inf. Comput. Sci.* 44, 60–67 (2004)
43. Torrens, F.: Table of periodic properties of fullerenes based on structural parameters. *J. Mol. Struct. (Theochem)* 709, 135–142 (2004)
44. Torrens, F., Castellano, G.: Cluster origin of the solubility of single-wall carbon nanotubes. *Comput. Lett.* 1, 331–336 (2005)
45. Torrens, F., Castellano, G.: Cluster origin of the solubility of single-wall carbon nanotubes. In: Maroulis, G. (ed.) *Structures and Properties of Clusters: From a few Atoms to Nanoparticles*, Brill, Leiden. Lecture Series on Computer and Computational Sciences, vol. 5, pp. 187–192 (2006)
46. Torrens, F., Castellano, G.: Effect of packing on cluster solvation of nanotubes. In: Bandyopadhyay, S., Cahay, M. (eds.) *Nanotechnology VI*, Institute of Electrical and Electronics Engineers, Piscataway (NJ), pp. 1–4 (2006)
47. Torrens, F., Castellano, G.: Cluster origin of the transfer phenomena of single-wall carbon nanotubes. *J. Comput. Theor. Nanosci.* 4, 588–603 (2007)
48. Torrens, F., Castellano, G.: Cluster nature of the solvation features of single-wall carbon nanotubes. In: Columbus, F. (ed.) *Progress in Nanotechnology Research*, Nova, Hauppauge (NY), pp. 1–28 (2007)
49. Torrens, F., Castellano, G.: Nuevo diseño y aproximaciones no ortodoxas con nanotubos de carbono. In: García-Breijo, E., Gil-Sánchez, L., Maquieira-Catalá, Á., Marcos-Martínez, M.D., Martínez-Máñez, R., Puchades-Pla, R., Ros-Lis, J.V., Sancenón-Galarza, F., Soto-Camino, J. (eds.) *Workshop on Sensors: A Local Approach*, Universidad Politécnica de Valencia, València, pp. 409–415 (2008)
50. Torrens, F., Castellano, G.: Effect of packing on the cluster nature of C nanotubes: An information entropy analysis. *Microelectron. J.* 38, 1109–1122 (2007)
51. Torrens, F., Castellano, G.: Asymptotic analysis of coagulation–fragmentation equations of carbon nanotube clusters. *Nanoscale Res. Lett.* 2, 337–349 (2007)
52. Torrens, F.: Effect of elliptical deformation on molecular polarizabilities of model carbon nanotubes from atomic increments. *J. Nanosci. Nanotech.* 3, 313–318 (2003)

53. Torrens, F.: Molecular and atomic polarizabilities of model carbon nanotubes. In: Shaw, L.L., Suryanarayana, C., Mishra, R.S. (eds.) *Processing and Properties of Structural Nanomaterials*, pp. 35–42. TMS, Warrendale (2003)
54. Torrens, F.: Effect of elliptical deformation on molecular polarizabilities of model carbon nanotubes from atomic increments. In: Kamat, P.V., Guldi, D.M., D’Souza, F. (eds.) *Fullerenes and Nanotubes: The Building Blocks of Next Generation Nanodevices*, Fullerenes No. 13. pp. 383–389. The Electrochemical Society, Pennington (2003)
55. Torrens, F.: Effect of type, size and deformation on polarizability of carbon nanotubes from atomic increments. *Nanotechnology* 15, S259–S264 (2004)
56. Torrens, F.: Effect of size and deformation on polarizabilities of carbon nanotubes from atomic increments. *Future Generation Comput. Syst.* 20, 763–772 (2004)
57. Torrens, F.: Periodic table of carbon nanotubes based on the chiral vector. *Internet Electron. J. Mol. Des.* 3, 514–527 (2004)
58. Torrens, F.: Calculation of partition coefficients of single-wall carbon nanotubes. In: Kramer, S.D. (ed.) *Lipophilicity III*, pp. 1–18. ETH, Zurich (2004)
59. Torrens, F.: Calculations on solvents and co-solvents of single-wall carbon nanotubes: Cyclopyranoses. In: Seijas, J.A., Vázquez-Tato, M.P. (eds.) *Synthetic Organic Chemistry VIII*. Universidad de Santiago de Compostela, Santiago de Compostela, pp. 1–14 (2004)
60. Torrens, F.: Calculations on cyclopyranoses as co-solvents of single-wall carbon nanotubes. *Mol. Simul.* 31, 107–114 (2005)
61. Torrens, F.: Periodic properties of carbon nanotubes based on the chiral vector. *Internet Electron. J. Mol. Des.* 4, 59–81 (2005)
62. Torrens, F.: Some calculations on single-wall carbon nanotubes. *Probl. Nonlin. Anal. Eng. Syst.* 11(2), 1–16 (2005)
63. Torrens, F.: Calculations on solvents and co-solvents of single-wall carbon nanotubes: Cyclopyranoses. *Nanotechnology* 16, S181–S189 (2005)
64. Torrens, F.: Calculations on solvents and co-solvents of single-wall carbon nanotubes: Cyclopyranoses. *J. Mol. Struct. (Theochem)* 757, 183–191 (2005)
65. Torrens, F.: Partition of solvents and co-solvents of nanotubes: Proteins and cyclopyranoses. In: Caldwell, G.W., Atta-ur-Rahman, Springer, B.A. (eds.) *Frontiers in Drug Design and Discovery I*, Bentham, Hilversum, Holland, pp. 231–266 (2005)
66. Torrens, F.: Calculations on solvents and co-solvents of single-wall carbon nanotubes: Cyclopyranoses. In: Cahay, M., Urquidi-Macdonald, M., Bandyopadhyay, S., Guo, P., Hasegawa, H., Koshida, N., Leburton, J.P., Lockwood, D.J., Seal, S., Stella, A. (eds.) *Nanoscale Devices, Materials, and Biological Systems: Fundamental and Applications*, pp. 439–458. The Electrochemical Society, Pennington (2005)
67. Torrens, F.: Calculations of organic-solvent dispersions of single-wall carbon nanotubes. *Int. J. Quantum. Chem.* 106, 712–718 (2006)
68. Torrens, F.: Corrigendum: Effect of type, size and deformation on polarizability of carbon nanotubes from atomic increments. *Nanotechnology* 17, 1541 (2006)
69. Torrens, F.: Calculations on solvents and co-solvents of carbon nanotubes: Cyclopyranoses. In: Geckeler, K.E., Rosenberg, E. (eds.) *Functional Nanomaterials*, ch. 5, pp. 1–13. American Scientific, Stevenson Ranch (2006)
70. Hooper, L.V., Wong, M.H., Thelin, A., Hansson, L., Falk, P.G., Gordon, J.I.: Molecular analysis of commensal host-microbial relationships in the intestine. *Science* 291, 881–884 (2001)

71. Herias, M.V., Hessle, C., Telemo, E., Midtvedt, T., Hanson, L.A., Wold, A.E.: Immunomodulatory effects of *Lactobacillus plantarum* colonizing the intestine of gnotobiotic rats. *Clin. Exp. Immunol.* 116, 283–290 (1999)
72. Cebrà, J.J.: Influences of microbiota on intestinal immune system development. *Am. J. Clin. Nutr.* 69, 1046S–1051S (1999)
73. Perdigón, G., Rachid, M., de Budeguer, M.V., Valdez, J.C.: Effect of yogurt feeding on the small and large intestine associated lymphoid cells in mice. *J. Dairy Res.* 61, 553–562 (1994)
74. Link-Amster, H., Rochat, F., Saudan, K.Y., Mignot, O., Aeschlimann, J.M.: Modulation of a specific humoral immune response and changes in intestinal flora mediated through fermented milk intake. *FEMS Immunol. Med. Microbiol.* 10, 55–63 (1994)
75. Marteau, P.R., de Vrese, M., Cellier, C.J., Schrezenmeir, J.: Protection from gastrointestinal diseases with the use of probiotics. *Am. J. Clin. Nutr.* 73(supl. 2), 430S–436S (2001)
76. Isolauri, E., Juntunen, M., Rautanen, T., Sillanaukee, P., Koivula, T.: A human *Lactobacillus* strain (*Lactobacillus casei* sp. strain GG) promotes recovery from acute diarrhea in children. *Pediatrics* 88, 90–97 (1991)
77. Malin, M., Suomalainen, H., Saxelin, M., Isolauri, E.: Promotion of IgA immune response in patients with Crohn's disease by oral bacteriotherapy with *Lactobacillus* GG. *Ann. Nutr. Metab.* 40, 137–145 (1996)
78. Borchers, A.T., Keen, C.L., Gershwin, M.E.: The influence of yogurt/*Lactobacillus* on the innate and acquired immune response. *Clin. Rev. Allergy Immunol.* 22, 207–230 (2002)
79. Ha, C.L., Lee, J.H., Zhou, H.R., Ustunol, Z., Pestka, J.J.: Effects of yogurt ingestion on mucosal and systemic cytokine gene expression in the mouse. *J. Food Prot.* 62, 181–188 (1999)
80. Halpern, G.M., Vruwink, K.G., van de Water, J., Keen, C.L., Gershwin, M.E.: Influence of long-term yoghurt consumption in young adults. *Int. J. Immunother.* 7, 205–210 (1991)
81. Solis-Pereyra, B., Aattouri, N., Lemonnier, D.: Role of food in the stimulation of cytokine production. *Am. J. Clin. Nutr.* 66, 521S–525S (1997)
82. Schiffriñ, E.J., Rochat, F., Link-Amster, H., Aeschlimann, J.M., Donnet-Hughes, A.: Immunomodulation of human blood cells following the ingestion of lactic acid bacteria. *J. Dairy Sci.* 78, 491–497 (1995)
83. Pelto, L., Isolauri, E., Lilius, E.M., Nuutila, J., Salminen, S.: Probiotic bacteria down-regulate the milk-induced inflammatory response in milk-hypersensitive subjects but have an immunostimulatory effect in healthy subjects. *Clin. Exp. Allergy* 28, 1474–1479 (1998)
84. Matsuzaki, T., Yamazaki, R., Hashimoto, S., Yokokura, T.: The effect of oral feeding of *Lactobacillus casei* strain Shirota on immunoglobulin E production in mice. *J. Dairy Sci.* 81, 48–53 (1998)
85. Takagi, A., Matsuzaki, T., Sato, M., Nomoto, K., Morotomi, M., Yokokura, T.: Enhancement of natural killer cytotoxicity delayed murine carcinogenesis by a probiotic microorganism. *Carcinogenesis* 22, 599–605 (2001)
86. Chiang, B.L., Sheih, Y.H., Wang, L.H., Liao, C.K., Gill, H.S.: Enhancing immunity by dietary consumption of a probiotic lactic acid bacterium (*Bifidobacterium lactis* HN019): Optimization and definition of cellular immune responses. *Eur. J. Clin. Nutr.* 54, 849–855 (2000)

87. Haller, D., Blum, S., Bode, C., Hammes, W.P., Schiffrian, E.J.: Activation of human peripheral blood mononuclear cells by nonpathogenic bacteria in vitro: Evidence of NK cells as primary targets. *Infect. Immun.* 68, 752–759 (2000)
88. Puri, P., Rattan, A., Bijlani, R.L., Mahapatra, S.C., Nath, I.: Splenic and intestinal lymphocyte proliferation response in mice fed milk or yogurt and challenged with *Salmonella typhimurium*. *Int. J. Food Sci. Nutr.* 47, 391–398 (1996)
89. DeSimone, C., Vesely, R., Negri, R., Bianchi-Salvadori, B., Zanzoglu, S., Cilli, A., Lucci, L.: Enhancement of immune response of murine Peyer's patches by a diet supplemented with yogurt. *Immunopharmacol Immunotoxicol.* 9, 87–100 (1987)
90. Turchet, P., Laurenzano, M., Auboiron, S., Antoine, J.M.: Effect of fermented milk containing the probiotic *Lactobacillus casei* DN 114001 on winter infections in free-living elderly subjects: A randomised, controlled pilot study. *J. Nutr. Health Aging.* 7, 75–77 (2003)
91. Tejada-Simon, M.V., Pestka, J.J.: Proinflammatory cytokine and nitric oxide induction in murine macrophages by cell wall and cytoplasmic extracts of lactic acid bacteria. *J. Food Prot.* 62, 1435–1444 (1999)
92. Canfield, R.E.: The amino acid sequence of egg white lysozyme. *J. Biol. Chem.* 238, 2698–2707 (1963)
93. Jollès, J., Hermann, J., Niemann, B., Jollès, P.: Differences between the chemical structures of duck and hen egg-white lysozymes. *Eur. J. Biochem.* 1, 344–346 (1967)
94. Blake, C.C.F., Mair, G.A., North, A.C.T., Phillips, D.C., Sarma, V.R.: On the conformation of the hen egg-white lysozyme molecule. *Proc. R. Soc. London B* 167, 365–385 (1967)
95. Hermann, J., Jollès, J.: The primary structure of duck egg-white lysozyme II. *Biochim. Biophys. Acta* 200, 178–179 (1970)
96. Kaneda, M., Kato, T., Tominaga, N., Chitani, K., Narita, K.: The amino acid sequence of quail lysozyme. *J. Biochem. (Tokyo)* 66, 747–749 (1969)
97. LaRue, J.N., Speck Jr., J.C.: Fed Proc. 28, 662 (1969)
98. Jones, P.S.: Strategies for antiviral drug discovery. *Antivir. Chem. Chemother.* 9, 283–302 (1998)
99. Ellis, R.W.: New technologies for making vaccines. *Vaccine* 17, 1596–1604 (1999)
100. Root, M.J., Kay, M.S., Kim, P.S.: Protein design of an HIV-1 entry inhibitor. *Science* 291, 884–888 (2001)
101. Bernstein, J.M.: Antiviral chemotherapy: General overview. Wright State University School of Medicine, Dayton, OH (2000)
102. Crosby, A.W.: America's forgotten pandemic: The influenza of 1918. Cambridge University, Cambridge (2003)
103. Reid, A.H., Taubenberger, J.K.: The origin of the 1918 pandemic influenza virus: A continuing enigma. *J. Gen. Virol.* 84, 2285–2292 (2003)
104. Kash, J.C., Basler, C.F., García-Sastre, A., Cartera, V., Billharz, R., Swayne, D.E., Przygodzki, R.M., Taubenberger, J.K., Katze, J.G., Tumpey, T.M.: Global host immune response: Pathogenesis and transcriptional profiling of type A influenza viruses expressing the hemagglutinin and neuraminidase genes from the 1918 pandemic virus. *J. Virol.* 78, 9499–9511 (2004)
105. Tumpey, T.M., Basler, C.F., Aguilar, P.V., Zeng, H., Solórzano, A., Swayne, D.E., Cox, N.J., Katz, J.M., Taubenberger, J.K., Palese, P., García-Sastre, A.: Characterization of the reconstructed 1918 Spanish influenza pandemic virus. *Science* 310, 77–79 (2005)

106. Taubenberger, J.K., Reid, A.H., Lourens, R.M., Wang, R., Jin, G., Fanning, T.G.: Characterization of the 1918 influenza virus polymerase genes. *Nature* (London) 437, 889–893 (2005)
107. Zhang, X., Luo, J., Yang, L.: New invariant of DNA sequence based on 3DD-curves and its application on phylogeny. *J. Comput. Chem.* 28, 2342–2346 (2007)
108. Bernstein, F.C., Koetzle, T.F., Williams, G.J.B., Meyer Jr., E.F., Brice, M.D., Rodgers, J.R., Kennard, O., Shimanouchi, T., Tasumi, M.: The Protein Data Bank: A computer-based archival file for macromolecular structures. *J. Mol. Biol.* 112, 535–542 (1977)
109. Torrens, F., Sánchez-Marín, J., Sánchez-Pérez, E.: Didàctica empírica de la congelació de l'aigua. In: Riera, S. (ed.) *Actes del II Sympòsium sobre l'Ensenyament de les Ciències Naturals. Documents No. 11*, Eumo, Vic, pp. 595–600 (1989)
110. Torrens, F., Sánchez-Marín, J., Sánchez-Pérez, E.: Estudi interdisciplinari de la congelació de l'aigua. In: Riera, S. (ed.) *Actes del II Sympòsium sobre l'Ensenyament de les Ciències Naturals. Documents No. 11*, Eumo, Vic, pp. 669–669 (1989)
111. Torrens, F., Sánchez-Pérez, E., Sánchez-Marín, J.: Didáctica empírica de la forma molecular. *Enseñanza de las Ciencias Extra-III Congreso* (1), 267–268 (1989)
112. Torrens, F., Ortí, E., Sánchez-Marín, J.: Representación de propiedades moleculares en la didáctica de la química. In: *Colloquy University Pedagogy*. Horsori, Barcelone, pp. 375–379 (1991)
113. Sayle, R.A., Milner-White, E.J.: RASMOL: Biomolecular graphics for all. *Trends Biochem. Sci.* 20, 374–376 (1995)
114. Shindyalov, I.N., Bourne, P.E.: WPDB – PC Windows-based interrogation of macromolecular structure. *J. Appl. Crystallogr.* 28, 847–852 (1995)
115. Shindyalov, I.N., Bourne, P.E.: Protein data representation and query using optimized data decomposition. *CABIOS* 13, 487–496 (1997)
116. Walters, P., Stahl, M.: Program BABEL. University of Arizona, Tucson (1996)
117. Hendlich, M.: Databases for Protein–Ligand Complexes. *Acta Crystallogr. Sect. D* 54, 1178–1182 (1998)
118. Tsai, C.S.: A computer-assisted tutorial on protein structure. *J. Chem. Educ.* 78, 837–839 (2001)
119. MDL, Program Chime. MDL Information Systems, San Leandro, CA (2007)
120. Claros, M.G., Fernández-Fernández, J.M., González-Mañas, J.M., Herráez, Á., Sanz, J.M., Urdiales, J.L.: BioROM 1.0 y 1.1. Sociedad Española de Bioquímica y Biología Molecular, Málaga (2001)
121. Claros, M.G., Fernández-Fernández, J.M., García-Vallvé, S., González-Mañas, J.M., Herráez, Á., Oliver, J., Pons, G., Pujadas, G., Roca, P., Rodríguez, S., Sanz, J.M., Segués, T., Urdiales, J.L.: BioROM 2002. Sociedad Española de Bioquímica y Biología Molecular, Málaga (2002)
122. Claros, M.G., Alonso, T., Corzo, J., Fernández-Fernández, J.M., García-Vallvé, S., González-Mañas, J.M., Herráez, Á., Oliver, J., Pons, G., Roca, P., Sanz, J.M., Segués, T., Urdiales, J.L., Valle, A., Villalaín, J.: BioROM 2003. Sociedad Española de Bioquímica y Biología Molecular–Roche Diagnostics, Málaga (2003)
123. Claros, M.G., Alfama, R., Alonso, T., Amthauer, R., Castro, E., Corzo, J., Fernández-Fernández, J.M., Figueroa, M., García-Vallvé, S., González-Mañas, J.M., Herráez, Á., Herrera, R., Moya, A., Oliver, J., Pons, G., Roca, P., Sanz, J.M., Segués, T., Tejedor, M.C., Urdiales, J.L., Villalaín, J.: BioROM 2005. Sociedad Española de Bioquímica y Biología Molecular–Universidad Miguel Hernández–Universidad del País Vasco, Málaga (2004)

124. Claros, M.G., Alfama, R., Alonso, T., Amthauer, R., Castro, E., Corzo, J., Fernández-Fernández, J.M., Figueroa, M., García-Mondéjar, L., García-Vallvé, S., Garrido, M.B., González-Mañas, J.M., Herráez, Á., Herrera, R., Miró, M.J., Moya, A., Oliver, J., Palacios, E., Pons, G., Roca, P., Sanz, J.M., Segués, T., Tejedor, M.C., Urdiales, J.L., Villalaín, J.: BioROM 2006 Sociedad Española de Bioquímica y Biología Molecular-Pearson Educación, Málaga (2005)
125. Herráez, Á.: Biomolecules in the computer: Jmol to the rescue. *Biochem. Mol. Biol. Educ.* 34, 255–261 (2006)
126. Claros, M.G., Alfama, R., Alonso, T., Amthauer, R., Carrero, I., Castro, E., Corzo, J., Fernández-Fernández, J.M., Figueroa, M., García-Vallvé, S., Garrido, M.B., González-Mañas, J.M., Herráez, Á., Herrera, R., Miró, M.J., Moya, A., Oliver, J., Palacios, E., Pons, G., Roca, P., Salgado, J., Sancho, P., Sanz, J.M., Segués, T., Tejedor, M.C., Urdiales, J.L., Villalaín, J.: BioROM 2007. Sociedad Española de Bioquímica y Biología Molecular-Pearson Educación, Málaga (2006)
127. Miró, M.J., Méndez, M.T., Raposo, R., Herráez, Á., Barrero, B., Palacios, E.: Desarrollo de una asignatura virtual de tercer ciclo como un espacio de enseñanza-aprendizaje que permite la participación activa del alumno. In: III Jornada Campus Virtual UCM: Innovación en el Campus Virtual, Metodologías y Herramientas, pp. 304–306. Complutense, Madrid (2007)
128. Torrens, F., Sánchez-Marín, J., Nebot-Gil, I.: Fractals for hybrid orbitals in protein models. In: Laxminarayan, S. (ed.) *Information Technology Applications in Biomedicine*, pp. 1–6. IEEE, Washington (1998)
129. Torrens, F.: Análisis fractal de la estructura terciaria de las proteínas. *Encuentros en la Biología* 8(64), 4–6 (2000)
130. Torrens, F.: Fractal hybrid orbitals in biopolymer chains. *Zh Fiz Khim* 74, 125–131 (2000)
131. Torrens, F.: Fractal hybrid orbitals in biopolymer chains. *Russ. J. Phys. Chem. (Engl. Transl.)* 74, 115–120 (2000)
132. Torrens, F.: Fractals for hybrid orbitals in protein models. *Complexity Int.* 8: torren01-1–13 (2001)
133. Torrens, F.: Fractal hybrid orbitals analysis of tertiary structure of protein molecule. In: Kappe, O., Merino, P., Marzinzik, A., Wennemers, H., Wirth, T., vanden Eynde, J.J., Lin, S.K. (eds.) *Synthetic Organic Chemistry V*, pp. 1–11. MDPI, Basel (2001)
134. Torrens, F.: Fractal hybrid orbitals analysis of the tertiary structure of protein molecules. *Molecules* 7, 26–37 (2002)
135. Covino, B.G.: Local anesthesia. *N. Engl. J. Med.* 286, 975–983 (1972)
136. Covino, B.G.: Local anesthetic agents for peripheral nerve blocks. *Anaesthetist* 29(7), 33–37 (1980)
137. Covino, B.G.: Pharmacology of local anaesthetic agents. *Br. J. Anaesth.* 58, 701–716 (1986)
138. Corriou, J.P., Iordache, O., Tondeur, D.: Classification of biomolecules by information entropy. *J. Chim. Phys. Phys.-Chim. Biol.* 88, 2645–2652 (1991)
139. Fawcett, J.P., Kennedy, J.M., Kumar, A., Ledger, R., Kumara, G.M., Patel, M.J., Zacharias, M.: Comparative efficacy and pharmacokinetics of racemic bupivacaine and S bupivacaine in third molar surgery. *J. Pharm. Pharmaceut. Sci.* 5, 199–204 (2002)
140. Brodin, A., Nyquist-Mayer, A., Wadstein, T.: Phase diagram and aqueous solubility of the lidocaine-prilocaine binary system. *J. Pharm. Sci.* 73, 481–484 (1984)

141. Friedman, P.M., Fogelman, J.P., Nouri, K., Levine, V.J., Ashinoff, R.: Comparative study of the efficacy of four topical anesthetics. *Dermatol. Surg.* 25, 950–954 (1999)
142. Friedman, P.M., Mafong, E.A., Friedman, E.S., Geronemus, R.G.: Topical anesthetics update: EMLA and beyond. *Dermatol. Surg.* 27, 1019–1026 (2001)
143. Kuwahara, R.T., Skinner Jr., R.B.: EMLA versus ice as a topical anesthetic. *Dermatol. Surg.* 27, 495–496 (2001)
144. Díez-Sales, O., Copoví, A., Casabó, V.G., Herráez, M.: A modelistic approach showing the importance of the stagnant aqueous layers in in vitro diffusion studies, and in vitro-in vivo correlations. *Int. J. Pharm.* 77, 1–11 (1991)
145. Díez-Sales, O., Guzmán, D., Cano, D., Martín, A., Sánchez, E., Herráez, M.: A comparative in vitro study of permeability with different synthetic and biological membranes. *Eur. J. Drug. Metab. Pharmacokinet.* (spec. 3), 441–446 (1991)
146. Sánchez-Moyano, E., Seco, C., Santolaria, A., Fabra-Campos, S., Herráez, M., Martín-Villodre, M.: Partition behavior of anilines in bulk-phase and high-performance liquid chromatographic systems: Influence on correlation with biological constants. *J. Pharm. Sci.* 81, 720–725 (1992)
147. Díez-Sales, O., López-Castellano, A., Maiques-Lacer, F.J., Herráez-Domínguez, M.: An in vitro percutaneous absorption study of non ionic compounds across human skin. *Pharmazie* 48, 684–686 (1993)
148. Díez-Sales, O., Pérez-Sayas, E., Martín-Villodre, A., Herráez-Domínguez, M.: The prediction of percutaneous absorption: I. Influence of the dermis on in vitro permeation models. *Int. J. Pharm.* 100, 1–7 (1993)
149. Díez-Sales, O., Watkinson, A.C., Herráez-Domínguez, M., Javaloyes, C., Hadgraft, J.: A mechanistic investigation of the in vitro human skin permeation enhancing effect of Azone®. *Int. J. Pharm.* 129, 33–40 (1996)
150. López, A., Morant, M.J., Guzmán, D., Borrás-Blasco, J., Díez-Sales, O., Herráez, M.: Skin permeation model of phenylalkylcarboxylic homologous acids and their enhancer effect on percutaneous penetration of 5 fluorouracil. *Int. J. Pharm.* 139, 205–213 (1996)
151. López, A., Pellett, M.A., Llinares, F., Díez-Sales, O., Herráez, M., Hadgraft, J.: The enhancer effect of several phenyl alcohols on percutaneous penetration of 5 fluorouracil. *Pharm. Res.* 14, 681–685 (1997)
152. López, A., Faus, V., Díez-Sales, O., Herráez, M.: Skin permeation model of phenyl alcohols: Comparison of experimental conditions. *Int. J. Pharm.* 173, 183–191 (1998)
153. Torrens, F.: Fractal dimension of transdermal-delivery drug models. In: Mastorakis, N., Er, M.J., D'Attelis, C. (eds.) *Non-linear Analysis, Non-linear Systems and Chaos*, pp. 1–6. WSEAS, Athens (2003)
154. Torrens, F.: Fractal dimension of transdermal-delivery drug models. *Leb. Sci. J.* 5(1), 61–70 (2004)
155. Leonard, J.T., Roy, K.: QSAR modeling of anti HIV activities of alkenyldiarylmethanes using topological and physicochemical descriptors. *Drug. Des. Discov.* 18, 165–180 (2003)
156. Leonard, J.T., Roy, K.: Classical QSAR modeling of HIV-1 reverse transcriptase inhibitor 2 amino 6 arylsulfonylbenzonitriles and congeners. *QSAR Comb. Sci.* 23, 23–35 (2004)
157. Roy, K., Leonard, J.T.: QSAR modeling of HIV-1 reverse transcriptase inhibitor 2 amino 6 arylsulfonylbenzonitriles and congeners using molecular connectivity and E state parameters. *Bioorg. Med. Chem.* 12, 745–754 (2004)

158. Leonard, J.T., Roy, K.: Classical QSAR modeling of CCR5 receptor binding affinity of substituted benzylpyrazoles. *QSAR Comb. Sci.* 23, 387–398 (2004)
159. Roy, K., Leonard, J.T.: Classical QSAR modeling of anti-HIV-2,3-diaryl 1,3-thiazolidin-4-ones. *QSAR Comb. Sci.* 24, 579–592 (2005)
160. Roy, K., Leonard, J.T.: QSAR by LFER model of cytotoxicity data of anti -HIV 5-phenyl-1-phenylamino-1H-imidazole derivatives using principal component analysis and genetic function approximation. *Bioorg. Med. Chem.* 13, 2967–2973 (2005)
161. Roy, K., Leonard, J.T.: Topological QSAR modeling of cytotoxicity data of anti-HIV 5-phenyl-1-phenylamino-1H-imidazole derivatives using GFA, G/PLS, FA and PCRA techniques. *Indian J. Chem. Sect. A* 45, 126–137 (2006)
162. Roy, K., Leonard, J.T.: QSAR analyses of 3-(4-benzylpiperidin-1-yl)-N-phenyl-propylamine derivatives as potent CCR5 antagonists. *J. Chem. Inf. Model* 45, 1352–1368 (2005)
163. Leonard, J.T., Roy, K.: QSAR by LFER model of HIV protease inhibitory data of mannitol derivatives using FA-MLR, PCRA and PLS techniques. *Bioorg. Med. Chem.* 14, 1039–1046 (2006)
164. Leonard, J.T., Roy, K.: The HIV entry inhibitors revisited. *Curr. Med. Chem.* 13, 911–934 (2006)
165. Calvo, D., Dopazo, J., Vega, M.A.: Cd36, CLA-1 (Cd36L1), and limpia (Cd36L2) gene family – Cellular-distribution, chromosomal location, and genetic evolution. *Genomics* 25, 100–106 (1995)
166. Escarmis, C., Dopazo, J., Davila, M., Palma, E.L., Domingo, E.: Large deletions in the 5'-untranslated region of foot-and-mouth-disease virus of serotype-C. *Virus Res.* 35, 155–167 (1995)
167. Rojas, J.M., Dopazo, J., Santana, M., Lopez-Galindez, C., Tabares, E.: Comparative-study of the genetic-variability in thymidine kinase and glycoprotein B genes of herpes-simplex viruses by the RNase-A mismatch cleavage method. *Virus Res.* 35, 205–214 (1995)
168. Sanchez-Palomino, S., Dopazo, J., Olivares, I., Martín, M.J., Lopez-Galindez, C.: Primary genetic-characterization of HIV-1 isolates from WHO-sponsored vaccine evaluation sites by the RNase-A mismatch method. *Virus Res.* 39, 251–259 (1995)
169. Olivares, I., Menendez-Arias, L., Rodriguez-Bernabe, A., Martín, M.J., Dopazo, J., Lopez-Galindez, C.: Sequence-analysis of HIV-1 vif gene in Spanish isolates. *Virus Genes.* 9, 283–288 (1995)
170. Trellés-Salazar, O., Zapata, E.L., Dopazo, J., Coulson, A.F.W., Carazo, J.M.: An image-processing approach to dotplots – An X-Window-based program for interactive analysis of dotplots derived from sequence and structural data. *Comput. Appl. Biosci.* 11, 301–308 (1995)
171. Rodrigo, M.J., Dopazo, J.: Evolutionary analysis of the picornavirus family. *J. Mol. Evol.* 40, 362–371 (1995)
172. De la Fraga, L.G., Dopazo, J., Carazo, J.M.: Confidence-limits for resolution estimation in image averaging by random subsampling. *Ultramicroscopy* 60, 385–391 (1995)
173. Quinones-Mateu, M.E., Dopazo, J., Este, J.A., Rota, T.R., Domingo, E.: Molecular characterization of human-immunodeficiency-virus type-1 isolates from Venezuela. *AIDS Res. Human Retrovir.* 11, 605–616 (1995)
174. Martín, M.J., González-Candelas, F., Sobrino, F., Dopazo, J.: Program ORF (Optimal Region Finder). Spanish EMBnet Node, CNB (1995)

175. Martín, M.J., Dopazo, J.: Program OSA (Optimal Sequence Analysis). Software Distribution (1995)
176. Martín, M.J., González-Candelas, F., Sobrino, F., Dopazo, J.: A method for determining the position and size of optimal sequence regions for phylogenetic analysis. *J. Mol. Evol.* 41, 1128–1138 (1995)
177. Suarez, P., Zardoya, R., Martín, M.J., Prieto, C., Dopazo, J., Solana, A., Castro, J.M.: Phylogenetic-relationships of European strains of porcine reproductive and respiratory syndrome virus (PRRSV) inferred from DNA-sequences of putative ORF-5 and ORF-7 genes. *Virus Res.* 42, 159–165 (1996)
178. Quinones-Mateu, M., Holguin, A., Dopazo, J., Najera, I., Domingo, E.: Point mutant frequencies in the pol gene of human-immunodeficiency-virus type-1 are 2-fold to 3-fold lower than those of env. *AIDS Res. Human Retrovir.* 12, 1117–1128 (1996)
179. Rzhetsky, A., Dopazo, J., Snyder, E., Dangler, C.A., Ayala, F.J.: Assessing Dissimilarity of genes by comparing their RNase-A mismatch cleavage patterns. *Genetics* 144, 1975–1983 (1996)
180. Blancourgoiti, B., Sanchez, F., Dopazo, J., Ponz, F.: A strain-type clustering of potato-virus-Y based upon the genetic-distance between isolates calculated by RFLP analysis of the amplified coat. *Arch. Virol.* 141, 2425–2442 (1996)
181. Weber, J., Fenyo, E.M., Beddows, S., Kaleebu, P., Bjorndal, A., Osmanov, S., Heyward, W., Esparza, J., Galvao-Castro, B., van de Perre, P., Karita, E., Wasi, C., Sempala, S., Tugume, B., Biryahwaho, B., Rubsamenaigmann, H., von Briesen, H., Esser, R., Grez, M., Holmes, H., Newberry, A., Ranjbar, S., Tomlinson, P., Bradac, J., McCutchan, F., Louwagie, J., Hegerich, P., Lopez-Galindez, C., Olivares, I., Dopazo, J., Mullins, J.: Neutralization serotypes of human-immunodeficiency-virus type-1 field isolates are not predicted by genetic subtype. *J. Virol.* 70, 7827–7832 (1996)
182. Taberner, A., Dopazo, J., Castanera, P.: Genetic-characterization of populations of a de-novo arisen sugar-beet pest, *Aubeonymus-mariaefranciscae* (coleoptera, curculionidae), by Rapd-analysis. *J. Mol. Evol.* 45, 24–31 (1997)
183. Aranda, M.A., Fraile, A., Dopazo, J., Malpica, J.M., García-Arenal, F.: Contribution of mutation and RNA recombination to the evolution of a plant pathogenic RNA. *J. Mol. Evol.* 44, 81–88 (1997)
184. Dopazo, J., Carazo, J.M.: Phylogenetic reconstruction using an unsupervised growing neural network that adopts the topology of a phylogenetic tree. *J. Mol. Evol.* 44, 226–233 (1997)
185. Dopazo, J.: A new index to find regions showing an unexpected variability or conservation in sequence alignments. *Comput. Appl. Biosci.* 13, 313–317 (1997)
186. Martinez, I., Dopazo, J., Melero, J.A.: Antigenic structure of the human respiratory syncytial virus G-glycoprotein and relevance of hypermutation events for the generation of antigenic variants. *J. Gen. Virol.* 78, 2419–2429 (1997)
187. Núñez, J.I., Blanco, E., Hernandez, T., Gomez-Tejedor, G., Martín, M.J., Dopazo, J., Sobrino, F.: A RT-PCR assay for the differential-diagnosis of vesicular viral diseases of swine. *J. Virol. Methods* 72, 227–235 (1998)
188. Martín, M.J., Núñez, J.I., Sobrino, F., Dopazo, J.: A procedure for detecting selection in highly variable viral genomes – Evidence of positive selection in antigenic regions of capsid protein VP1 of foot-and-mouth-disease virus. *J. Virol. Methods* 74, 215–221 (1998)
189. Wang, H.C., Dopazo, J., Carazo, J.M.: Self organizing tree growing network for classifying amino acids. *Bioinformatics* 14, 376–377 (1998)

190. Trelles, O., Ceron, C., Wang, H.C., Dopazo, J., Carazo, J.M.: New phylogenetic venues opened by a novel implementation of the DNAmI algorithm. *Bioinformatics* 14, 544–545 (1998)
191. Saiz, J.C., Lopez-Labrador, F.X., Ampurdanes, S., Dopazo, J., Forns, X., Sanchez-Tapias, J.M., Rodes, J.: The prognostic relevance of the nonstructural 5A gene interferon sensibility determining region is different in infections with genotype 1B and 3A isolates of hepatitis-C virus. *J. Infect. Diseases* 177, 839–847 (1998)
192. Blancourgoiti, B., Sanchez, F., Desanroman, C.P., Dopazo, J., Ponz, F.: Potato-virus-Y group-C isolates are a homogeneous pathotype but 2 different genetic strains. *J. Gen. Virol.* 79, 2037–2042 (1998)
193. Wang, H.C., Dopazo, J., de la Fraga, L.G., Zhu, Y.P., Carazo, J.M.: Self organizing tree-growing network for the classification of protein sequences. *Prot. Sci.* 7, 2613–2622 (1998)
194. Núñez, J.I., Blanco, E., Hernandez, T., Dopazo, J., Sobrino, F.: RT-PCR in foot-and-mouth-disease diagnosis. *Vet Q* 20, 34–36 (1998)
195. Oliveros, J.C., Blaschke, C., Herrero, J., Dopazo, J., Valencia, A.: Expression profiles and biological function. In: *Genome Informatics Workshop*, vol. 11, pp. 106–117 (2000)
196. Martín, M.J., Dopazo, J.: Program OSA. *Bioinformatics Catalogue of Molecular Biology Programs AC BC00469* (2000)
197. Núñez, J.I., Martín, M.J., Piccone, M.E., Carrillo, E., Palma, E.L., Dopazo, J., Sobrino, F.: Identification of optimal regions for phylogenetic studies on VP1 gene of foot-and-mouth disease virus: Analysis of types A and O Argentinean viruses. *Vet. Res.* 32, 31–45 (2001)
198. Herrero, J., Valencia, A., Dopazo, J.: A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics* 17, 126–136 (2001)
199. Dopazo, J., Zanders, E., Dragoni, I., Amphlett, G., Falciani, F.: Methods and approaches in the analysis of gene expression data. *J. Immunol. Meth.* 250, 93–112 (2001)
200. Dopazo, J., Mendoza, A., Herrero, J., Caldara, F., Humbert, Y., Friedli, L., Guerrier, M., Grand-Schenk, E., Gandin, C., de Francesco, M., Polissi, A., Buell, G., Feger, G., García, E., Peitsch, M., García-Bustos, J.F.: Annotated draft genomic sequence from a *Streptococcus pneumoniae* type 19F clinical isolate. *Microbial. Drug Resistance* 7, 99–125 (2001)
201. Elena, S., Dopazo, J., de la Peña, M., Flores, R., Diener, T.O., Moya, A.: Phylogenetic analysis of viroid and viroid-like satellite RNAs from plants: A reassessment. *J. Mol. Evol.* 53, 155–159 (2001)
202. Mateos, Á., Herrero, J., Dopazo, J.: Using perceptrons for supervised classification of DNA microarray samples: Obtaining the optimal level of information and finding differentially expressed genes. In: Dorronsoro, J.R. (ed.) *ICANN 2002*. LNCS, vol. 2415, pp. 577–582. Springer, Heidelberg (2002)
203. Tamames, J., Clark, D., Herrero, J., Dopazo, J., Blaschke, C., Fernández, J.M., Oliveros, J.C., Valencia, A.: Bioinformatics methods for the analysis of expression arrays: Data clustering and information extraction. *J. Biotechnol.* 98, 269–283 (2002)
204. Herrero, J., Dopazo, J.: Combining hierarchical clustering and self organizing maps for exploratory analysis of gene expression patterns. *J. Proteome Res.* 1, 467–470 (2002)

205. Tracey, L., Villuendas, R., Ortiz, P., Dopazo, A., Spiteri, I., Lombardía, L., Rodríguez-Peralto, J.L., Fernández-Herrera, J., Hernández, A., Fraga, J., Domínguez, O., Herrero, J., Alonso, M.A., Dopazo, J., Prirs, M.A.: Identification of genes involved in resistance to Interferon- α a in cutaneous T-cell lymphoma. *Am. J. Pathol.* 161, 1825–1837 (2002)
206. Mateos, Á., Dopazo, J., Jansen, R., Tu, Y., Gerstein, M., Stolovitzky, G.: Systematic learning of gene functional classes from DNA array expression data by using multi-layer perceptrons. *Genome. Res.* 12, 1703–1715 (2002)
207. Herrero, J., Díaz-Uriarte, R., Dopazo, J.: An approach to inferring transcriptional regulation among genes form large-scale expression data. *Comparative and Functional Genomics* 4, 148–154 (2003)
208. Herrero, J., Díaz-Uriarte, R., Dopazo, J.: Gene expression data preprocessing. *Bioinformatics* 19, 655–656 (2003)
209. Martín, M.J., Herrero, J., Mateos, Á., Dopazo, J.: Comparing bacterial genomes through conservation profiles. *Genome. Res.* 15, 991–998 (2003)
210. Herrero, J., Al-Shahrour, D., Díaz-Uriarte, R., Mateos, Á., Vaquerizas, J.M., Santoyo, J., Dopazo, J.: GEPAS, a web-based resource for microarray gene expression data analysis. *Nucl. Acids. Res.* 31, 3461–3467 (2003)
211. Karzynski, M., Mateos, Á., Herrero, J., Dopazo, J.: Using a genetic algorithm and a perceptron for feature selection and supervised clase learning in DNA microarray data. *Artificial Intelligence Rev.* 20, 39–51 (2003)
212. Conde, L., Mateos, Á., Herrero, J., Dopazo, J.: Improved class prediction in DNA microarray gene expression data by unsupervised reduction of the dimensionality followed by supervised learning with a perceptron. *J. VLSI Signal Processing-Syst. Signal, Image, Video Technol.* 35, 245–253 (2003)
213. Moreno-Bueno, G., Sánchez-Estévez, C., Cassia, R., Rodríguez-Perales, S., Díaz-Uriarte, R., Domínguez, O., Hardisson, D., Andujar, M., Prat, J., Matías-Guiu, X., Cigudosa, J.C., Palacios, J.: Differential gene expression profile in endometrioid and nonendometrioid endometrial carcinoma: STK15 is frequently overexpressed and amplified in nonendometrioid carcinomas. *Cancer Res.* 63, 5697–5702 (2003)
214. Dopazo, H., Santoyo, J., Dopazo, J.: Phylogenomics and the number of characters required for obtaining an accurate phylogeny of eukaryote model species. *Bioinformatics* 20, 116–121 (2004)
215. Al-Shahrour, F., Díaz-Uriarte, R., Dopazo, J.: FatiGO: A web tool for finding significant associations of gene ontology terms with groups of genes. *Bioinformatics* 20, 578–580 (2004)
216. Conde, L., Vaquerizas, J.M., Santoyo, J., Al-Shahrour, F., Ruiz-Llorente, S., Robledo, M., Dopazo, J.: PupaSNP Finder: A web tool for finding SNPs with putative effect at transcriptional level. *Nucl. Acids Res.* 32, W242–W248 (2004)
217. Herrero, J., Vaquerizas, J.M., Al-Shahrour, F., Conde, L., Mateos, Á., Santoyo, J., Díaz-Uriarte, R., Dopazo, J.: New challenges in gene expression data analysis and the extended GEPAS. *Nucl. Acids Res.* 32, W485–W491 (2004)
218. Rodríguez-Perales, S., Meléndez, B., Gribble, S.M., Valle, L., Carter, N.P., Santamaría, I., Conde, L., Urioste, M., Benítez, J., Cigusoda, J.C.: Cloning of a new familial t(3;8) translocation associated with conventional renal cell carcinoma reveals a 5 kb microdeletion and no gene involved in the rearrangement. *Human Mol. Genet.* 13, 983–990 (2004)
219. Vaquerizas, J.M., Dopazo, J., Díaz-Uriarte, R.: DNMMAD: Web-based diagnosis and normalization for microarray data. *Bioinformatics* 20, 3656–3658 (2004)

220. Meléndez, B., Díaz-Uriarte, R., Martínez-Ramírez, Á., Fernández-Piqueras, J., Rivas, C., Dopazo, J., Martínez-Delgado, B., Benítez, J.: Gene expression analysis on chromosomal regions of gain or loss in genetic material detected by comparative genomic hybridization. *Genes, Chromosomes, Cancer* 41, 353–365 (2004)
221. Martínez-Delgado, B., Meléndez, B., Cuadros, M., Alvarez, J., Castrillo, J.M., Ruiz de la Parte, A., Mollejo, M., Bellas, C., Díaz-Uriarte, R., Lombardía, L., Al-Shahrour, F., Domínguez, O., Cascón, A., Robledo, M., Rivas, C., Benítez, J.: Expression profiling of T cell lymphomas differentiates peripheral and lymphoblastic lymphomas and defines survival related genes. *Clin. Cancer Res.* 10, 4971–4982 (2004)
222. Hoffmann, R., Dopazo, J., Cigudosa, J.C., Valencia, A.: HCAD, closing the gap between breakpoints and genes. *Nucl. Acids Res.* 33, D511–D513 (2005)
223. Alvarez de Andrés, S., Díaz-Uriarte, R., Osorio, A., Barrosa, A., Paz, M.F., Honrado, E., Rodríguez, R., Urioste, M., Valle, L., Diez, O., Cigudosa, J.C., Dopazo, J., Esteller, M., Benítez, J.: A predictor based on the somatic changes of the BRCA1/2 breast cancer tumors identifies the non BRCA1/2 tumors with BRCA1 promoter hypermethylation. *Clin. Cancer Res.* 11, 1146–1153 (2005)
224. Santoyo, J., Vaquerizas, J.M., Dopazo, J.: Highly specific and accurate selection of siRNAs for high-throughput functional assays. *Bioinformatics* 21, 1376–1382 (2005)
225. Palacios, J., Honrado, E., Osorio, A., Cazorla, A., Sarrio, D., Barroso, A., Rodriguez, S., Cigudosa, J.C., Diez, O., Alonso, C., Lerma, E., Dopazo, J., Rivas, C., Benítez, J.: Phenotypic characterization of BRCA1 and BRCA2 tumors based in a tissue microarray study with 37 immunohistochemical markers. *Breast Cancer Res. Treat* 90, 5–14 (2005)
226. Cascón, A., Ruiz-Llorente, S., Rodriguez-Perales, S., Honrado, E., Martínez-Ramírez, A., Letón, R., Montero-Conde, C., Benítez, J., Dopazo, J., Cigudosa, J.C., Robledo, M.: A novel candidate region linked to development of both pheochromocytoma and head/neck paraganglioma. *Genes, Chromosomes, Cancer* 42, 260–268 (2005)
227. Dopazo, H., Dopazo, J.: Genome-scale evidence of the nematode arthropod clade. *Genome Biol.* 6, R41–1–10 (2005)
228. Al-Shahrour, F., Minguez, P., Vaquerizas, J.M., Conde, L., Dopazo, J.: Babelomics: A suite of web-tools for functional annotation and analysis of group of genes in high-throughput experiments. *Nucl. Acids Res.* 33, W460–W464 (2005)
229. Conde, L., Vaquerizas, J.M., Ferrer-Costa, C., Orozco, M., Dopazo, J.: PupasView: A visual tool for selecting suitable SNPs, with putative pathologic effect in genes, for genotyping purposes. *Nucl. Acids Res.* 33, W501–W505 (2005)
230. Vaquerizas, J.M., Conde, L., Yankilevich, P., Cabezon, A., Minguez, P., Díaz-Uriarte, R., Al-Shahrour, F., Herrero, J., Dopazo, J.: Gepas an experiment-oriented pipeline for the analysis of microarray gene expression data. *Nucl. Acids Res.* 33, W616–W620 (2005)
231. Al-Shahrour, F., Díaz-Uriarte, R., Dopazo, J.: Discovering molecular functions significantly related to phenotypes by combining gene expression data and biological information. *Bioinformatics* 21, 2988–2993 (2005)
232. Gabaldón, T.: Evolution of proteins and proteomes: A phylogenetics approach. *Evol. Bioinformatics Online* 1, 51–61 (2005)
233. Azuaje, F., Dopazo, J. (eds.): Data analysis and visualization in genomics and proteomics. Wiley, Bognor Regis (2005)

234. Largo, C., Alvarez, S., Saez, B., Blesa, D., Martin-Subero, J.I., González-García, I., Brieva, J.A., Dopazo, J., Siebert, R., Calasanz, M.J., Cigudosa, J.C.: Identification of overexpressed genes in frequently gained/amplified chromosome regions in multiple myeloma. *Haematologica* 91, 184–191 (2006)
235. Arbiza, L., Duchi, S., Montaner, D., Burguet, J., Pantoja-Uceda, D., Pineda-Lucena, A., Dopazo, J., Dopazo, H.: Selective pressures at a codon-level predict deleterious mutations in human disease genes. *J. Mol. Biol.* 358, 1390–1404 (2006)
236. Dopazo, J., Aloy, P.: Discovery and hypothesis generation through bioinformatics. *Genome Biol.* 7, 307–1–3 (2006)
237. Conde, L., Vaquerizas, J.M., Dopazo, H., Arbiza, L., Reumers, J., Rousseau, F., Schymkowitz, J., Dopazo, J.: PupaSuite: Finding functional SNPs for large-scale genotyping purposes. *Nucl. Acids Res.* 34, W621–W625 (2006)
238. Arbiza, L., Dopazo, J., Dopazo, H.: Positive selection, relaxation, and acceleration in the evolution of the human and chimp genomes. *PLoS Comp. Biol.* 2, e38–1–13 (2006)
239. Gabaldón, T., Snel, B., van Zimmeren, F., Hemrika, W., Tabak, H., Huynen, M.A.: Origin and evolution of the peroxisomal proteome. *Biol. Direct.* 1, 8–1–14 (2006)
240. Goñi, J.R., Vaquerizas, J.M., Dopazo, J., Orozco, M.: Exploring the reasons for the large density of triplex-forming oligonucleotide target sequences in the human regulatory regions. *BMC Genomics* 7, 63–1–10 (2006)
241. Al-Shahrour, F., Minguez, P., Tárraga, J., Montaner, D., Alloza, E., Vaquerizas, J.M., Conde, L., Blaschke, C., Vera, J., Dopazo, J.: BABELOMICS: A systems biology perspective in the functional annotation of genome-scale experiments. *Nucl. Acids Res.* 34, W472–W476 (2006)
242. Montaner, D., Tárraga, J., Huerta-Cepas, J., Burguet, J., Vaquerizas, J.M., Conde, L., Minguez, P., Vera, J., Mukherjee, S., Valls, J., Pujana, M., Alloza, E., Herrero, J., Al-Shahrour, F., Dopazo, J.: Next station in microarray data analysis: GEPAS. *Nucl. Acids Res.* 34, W486–W491 (2006)
243. Dopazo, J.: Bioinformatics and cancer: An essential alliance. *Clin. Transl. Oncol.* 8, 409–415 (2006)
244. Gabaldón, T.: Computational approaches for the prediction of protein function in the mitochondrion. *Am. J. Physiol. Cell Physiol.* 291, C1121–C1128 (2006)
245. Mine, R.L., Ribas, G., González-Neira, A., Fagerholm, R., Salas, A., Gonzalez, E., Dopazo, J., Nevanlinna, H., Robledo, M., Benítez, J.: ERCC4 associated with breast cancer risk: A two-stage case-control study using high-throughput genotyping. *Cancer Res.* 66, 9420–9427 (2006)
246. Minguez, P., Al-Shahrour, F., Dopazo, J.: A function-centric approach to the biological interpretation of microarray time-series. *Genome Informatics Ser.* 17, 57–66 (2006)
247. Dopazo, J.: Functional interpretation of microarray experiments. *OMICS* 10, 398–410 (2006)
248. Schluter, A., Fourcade, S., Domenech-Estevez, E., Gabaldón, T., Huerta-Cepas, J., Berthommier, G., Ripp, R., Wanders, R.J., Poch, O., Pujol, A.: PeroxisomeDB: A database for the peroxisomal proteome, functional genomics and disease. *Nucl. Acids Res.* 35, D815–D822 (2007)
249. Al-Shahrour, F., Arbiza, L., Dopazo, H., Huerta, J., Minguez, P., Montaner, D., Dopazo, J.: From genes to functional classes in the study of biological systems. *BMC Bioinformatics* 8, 114–1–17 (2007)

250. Conde, L., Montaner, D., Burguet-Castell, J., Tárraga, J., Al-Shahrour, F., Dopazo, J.: Functional profiling and gene expression analysis of chromosomal copy number alterations. *Bioinformation* 1, 432–435 (2007)
251. Medina, I., Montaner, D., Tárraga, J., Dopazo, J.: Prophet, a web-based tool for class prediction using microarray data. *Bioinformatics* 23, 390–391 (2007)
252. Huerta-Cepas, J., Dopazo, H., Dopazo, J., Gabaldón, T.: The human phylome. *Genome. Biol.* 8, R109–1–16 (2007)
253. Hernández, P., Huerta-Cepas, J., Montaner, D., Al Shahroud, F., Valls, J., Gómez, L., Capellá, G., Dopazo, J., Puig, M.A.: Evidence for systems-level molecular mechanisms of tumorigenesis. *BMC Genomics* 8, 115–1–12 (2007)
254. Rico, D., Vaquerizas, J.M., Dopazo, H., Boscá, L.: Identification of conserved domains in the promoter regions of nitric oxide synthase 2: Implications for the species-specific transcription and evolutionary differences. *BMC Genomics* 8, 271–1–10 (2007)
255. Conde, L., Montaner, D., Burguet-Castell, J., Tárraga, J., Medina, I., Al-Shahrour, F., Dopazo, J.: ISACGH: A web-based environment for the analysis of Array CGH and gene expression which includes functional profiling. *Nucl. Acids Res.* 35, W81–W85 (2007)
256. Al-Shahrour, F., Minguez, P., Tárraga, J., Medina, I., Alloza, E., Montaner, D., Dopazo, J.: FatiGO+: A functional profiling tool for genomic data. Integration of functional annotation, regulatory motifs and interaction data with microarray experiments. *Nucl. Acids Res.* 35, W91–W96 (2007)
257. Martí-Renom, M.A., Pieper, U., Madhusudhan, M.S., Rossi, A., Eswar, N., Davis, F.P., Al-Shahrour, F., Dopazo, J., Sali, A.: DBAli tools: Mining the protein structural space. *Nucl. Acids Res.* 35, W393–W397 (2007)
258. Tárraga, J., Medina, I., Arbiza, L., Huerta, J., Gabaldón, T., Dopazo, J., Dopazo, H.: Phylemon: A suite of web tools for molecular evolution, phylogenetics and phylogenomics. *Nucl. Acids Res.* 35, W38–W42 (2007)
259. Martí-Renom, M.A., Rossi, A., Al-Shahrour, F., Davis, F.P., Pieper, U., Dopazo, J., Sali, A.: The AnnoLite and AnnoLyze programs for comparative annotation of protein structures. *BMC Bioinformatics* 8, S4–1–12 (2007)
260. Nueda, M.J., Conesa, A., Westerhuis, J.A., Hoefsloot, H.C., Smilde, A.K., Talón, M., Ferrer, A.: Discovering gene expression patterns in time course microarray experiments by ANOVA SCA. *Bioinformatics* 23, 1792–1800 (2007)
261. Gandía, M., Conesa, A., Ancillo, G., Gadea, J., Forment, J., Pallás, V., Flores, R., Duran-Vila, N., Moreno, P., Guerri, J.: Transcriptional response of Citrus aurantifolia to infection by Citrus tristeza virus. *Virology* 367, 298–306 (2007)
262. Montero-Conde, C., Martín-Campos, J.M., Lerma, E., Giménez, G., Martínez-Guitarte, J.L., Combalía, N., Montaner, D., Matías-Guiu, X., Dopazo, J., de Leiva, A., Robledo, M., Mauricio, D.: Molecular profiling related to poor prognosis in thyroid carcinoma. Combining gene expression data and biological information. *Oncogene* (2007), doi:10.1038/sj.onc.1210792
263. Minguez, P., Al-Shahrour, F., Montaner, D., Dopazo, J.: Functional profiling of microarray experiments using text-mining derived bioentities. *Bioinformatics* 23, 3098–3099 (2007)
264. Gabaldón, T., Peretó, J., Montero, F., Gil, R., Latorre, A., Moya, A.: Structural analysis of a hypothetical minimal metabolism. *Philos. Trans. R Soc. B* 362, 1751–1762 (2007)

265. Ruiz-Llorente, S., Montero-Conde, C., Milne, R.L., Martín-Moya, C., Cebrián, A., Letón, R., Cascón, A., Mercadillo, F., Landa, I., Borrego, S., Pérez de Nanclares, G., Álvarez-Escalá, G., Díaz-Pérez, J.A., Carracedo, A., Urioste, M., González-Neira, A., Benítez, J., Santisteban, P., Dopazo, J., Ponder, B.A., Robledo, M., The MTC Clinical Group: Association study of 69 genes in the ret pathway identifies low penetrance loci in sporadic medullary thyroid carcinoma. *Cancer Res.* 67, 9561–9567 (2007)
266. Stanley, W.A., Fodor, K., Martí-Renom, M.A., Schliebs, W., Wilmanns, M.: Protein translocation into peroxisomes by ring-shaped import receptors. *FEBS Lett.* 581, 4795–4802 (2007)
267. Aragues, R., Sali, A., Bonet, J., Martí-Renom, M.A., Oliva, B.: Characterization of protein hubs by inferring interacting motifs from protein interactions. *PLoS Comput. Biol.* 3, e178–1–11 (2007)
268. Eswar, N., Webb, B., Martí-Renom, M.A., Madhusudhan, M.S., Eramian, D., Shen, M.Y., Pieper, U., Sali, A.: Comparative protein structure modeling using MODELLER. *Cur. Protocols Prot. Sci.* S50, 2–9–1–31 (2007)
269. Gabaldón, T., Huynen, M.A.: From endosymbiont to host-controlled organelle: The hijacking of mitochondrial protein synthesis and metabolism. *PLoS Comp. Biol.* 3, e219–1–10 (2007)
270. Levin, A.M., de Vries, R.P., Conesa, A., de Bekker, C., Talón, M., Menke, H.H., van Peij, N.N.M.E., Wösten, H.A.B.: Spatial differentiation in the vegetative mycelium of *Aspergillus niger*. *Eukaryot Cell* 6, 2311–2322 (2007)
271. Capriotti, E., Arbiza, L., Casadio, R., Dopazo, J., Dopazo, H., Martí-Renom, M.A.: The use of estimated evolutionary strength at the codon level improves the prediction of disease related protein mutations in human. *Human Mutation* 29, 198–204 (2008)
272. Valls, J., Grau, M., Sole, X., Hernández, P., Montaner, D., Dopazo, J., Peinado, M.A., Capella, G., Pujana, M.A., Moreno, V.: CLEAR-test: Combining inference for differential expression and variability in microarray data analysis. *J. Biomed Informatics* 41, 33–45 (2008)
273. Capriotti, E., Martí-Renom, M.A.: Computational RNA structure prediction. *Curr. Bioinformatics* 3, 32–45 (2008)
274. Huerta-Cepas, J., Bueno, A., Dopazo, J., Gabaldón, T.: PhylomeDB: A database for complete collections of gene phylogenies. *Nucl. Acids Res.* 36, D491–D496 (2008)
275. Reumers, J., Conde, L., Medina, I., Maurer-Stroh, S., van Durme, J., Dopazo, J., Rousseau, F., Schymkowitz, J.: Joint annotation of coding and non coding single nucleotide polymorphisms and mutations in the 5 SNPeffect and PupaSuite databases. *Nucl. Acids Res.* 36, D825–D829 (2008)
276. Conesa, A., Götz, S.: Blast2GO: A comprehensive suite for functional analysis in plant genomics. *Int. J. Plant. Genomics* 2008, 619832–1–12 (2008)
277. Marcet-Houben, M., Gabaldón, T.: The tree versus the forest. I Jornada Conjunta GenProt, Red Valenciana de Genómica y Proteómica–Xarxa de Genòmica i Proteòmica, Peñíscola, Castellón, Spain (2008)
278. Felsenstein, J.: Parsimony in systematics: Biological and statistical issues. *Annu. Rev. Ecol. Syst.* 14, 313–333 (1983)
279. Felsenstein, J.: Phylogenies and quantitative characters. *Annu. Rev. Ecol. Syst.* 19, 445–471 (1988)
280. Felsenstein, J.: Phylogenies from molecular sequences: Inference and reliability. *Annu. Rev. Genet.* 22, 521–565 (1988)
281. Felsenstein, J.: Inferring Phylogenies. Sinauer, Sunderland (2003)

282. Avise, J.C., Neigel, J.E., Arnold, J.: Demographic influences on mitochondrial DNA lineage survivorship in animal populations. *J. Mol. Evol.* 20, 99–105 (1984)
283. Rammal, R., Toulouse, G., Virasoro, M.A.: Ultrametricity for physicists. *Rev. Mod. Phys.* 58, 765–788 (1986)
284. Lewin, R.: Molecular clocks run out of time: The theory that we can date the birth of new species by charting the steady accumulation of mutations over evolutionary time is in serious trouble. *New Scientist* 125, 38–41 (1990)
285. Hillis, D.M., Bull, J.J., White, M.E., Badgett, M.R., Molineux, I.J.: Experimental phylogenetics: generation of a known phylogeny. *Science* 255, 589–592 (1992)
286. Steel, M.: The complexity of reconstructing trees from qualitative characters and subtrees. *J. Classification* 9, 91–116 (1992)
287. Sicheritz-Pontén, T., Andersson, S.G.E.: A phylogenomic approach to microbial evolution. *Nucl. Acids Res.* 29, 545–552 (2001)
288. Otu, H.H., Sayood, K.: A new sequence distance measure for phylogenetic tree construction. *Bioinformatics* 19, 2122–2130 (2003)
289. Ouzounis, C.A., Valencia, A.: Early bioinformatics: The birth of a discipline—A personal view. *Bioinformatics* 19, 2176–2190 (2003)
290. Pray, L.A.: Phylogenetics: Even the terminology evolves. *Scientist* 17(11), 14 (2003)
291. Pray, L.A.: Modern phylogeneticists branch out. *Scientist* 17(11), 35–36 (2003)
292. Makarenkov, V., Lapointe, F.J.: A weighted least-squares approach for inferring phylogenies from incomplete distance matrices. *Bioinformatics* 20, 2113–2121 (2004)
293. Chen, D., Eulenstein, O., Fernández-Baca, D.: Rainbow: A toolbox for phylogenetic supertree construction and analysis. *Bioinformatics* 20, 2872–2873 (2004)
294. Parr, C.S., Lee, B., Campbell, D., Bederson, B.B.: Visualizations for taxonomic and phylogenetic trees. *Bioinformatics* 20, 2997–3004 (2004)
295. Peirce, J.L.: Following phylogenetic footprints. *Scientist* 18(18), 34–37 (2004)
296. Roberts, J.P.: New growth in phylogeny programs. *Scientist* 18(24), 22–23 (2004)
297. Brown, J.R.: Putting the bio back in bioinformatics. *Scientist* 18(24), 44–45 (2004)
298. Gorder, P.F.: Computing life's family tree. *Comput. Sci. Eng.* 7(3), 3–6 (2005)
299. Hoef-Emden, K.: Molecular phylogenetic analyses and real-life data. *Comput. Sci. Eng.* 7(3), 86–91 (2005)
300. Ciccarelli, F.D., Doerks, T., von Mering, C., Creevey, C.J., Snel, B., Bork, P.: Toward automatic reconstruction of a highly resolved tree of life. *Science* 311, 1283–1287 (2006)
301. Kimura, M.: *New Scientist* 111, 41 (1985)
302. Stewart, C.B.: The powers and pitfalls of parsimony. *Nature (London)* 361, 603–607 (1993)
303. Li, M., Badger, J.H., Chen, X., Kwong, S., Kearney, P., Zhang, H.: An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics* 17, 149–154 (2001)
304. Dai, Q., Liu, X.Q., Wang, T.M., Vukicevic, D.: Linear regression model of DNA sequences and its application. *J. Comput. Chem.* 28, 1434–1445 (2007)
305. Liao, B., Zhu, W., Liu, Y.: 3D graphical representation of DNA sequence without degeneracy and its applications in constructing phylogenetic tree. *MATCH Commun. Math. Comput. Chem.* 56, 209–216 (2006)
306. Sarich, V.M., Wilson, A.C.: Immunological time scale for hominid evolution. *Science* 158, 1200–1203 (1967)

307. Sibley, C.G., Ahlquist, J.E.: The phylogeny of the hominoid primates, as indicated by DNA–DNA hybridization. *J. Mol. Evol.* 20, 2–15 (1984)
308. Cann, R.L., Stoneking, M., Wilson, A.C.: Mitochondrial DNA and human evolution. *Nature (London)* 325, 31–36 (1987)
309. Hayasaka, K., Gojobori, T., Horai, S.: Molecular phylogeny and evolution of primate mitochondrial DNA. *Mol. Biol. Evol.* 5, 626–644 (1988)
310. Cavalli Sforza, L.L., Piazza, A., Menozzi, P., Mountain, J.: Reconstruction of human evolution: Bringing together genetic, archaeological, and linguistic data. *Proc. Natl. Acad. Sci. USA* 85, 6002–6006 (1988)
311. Ruvolo, M., Disotell, T.R., Allard, M.W., Brown, W.M., Honeycutt, R.L.: Resolution of the African hominoid trichotomy by use of a mitochondrial gene sequence. *Proc. Natl. Acad. Sci. USA* 88, 1570–1574 (1991)
312. Di Rienzo, A., Wilson, A.C.: Branching pattern in the evolutionary tree for human mitochondrial DNA. *Proc. Natl. Acad. Sci. USA* 88, 1597–1601 (1991)
313. Torrens, F.: Filogénesis de los simios antropoides. *Encuentros en la Biología* 8(60), 3–5 (2000)
314. Shen, P., Wang, F., Underhill, P.A., Franco, C., Yang, W.H., Roxas, A., Sung, R., Lin, A.A., Hyman, R.W., Vollrath, D., Davis, R.W., Cavalli-Sforza, L.L., Oefner, P.J.: Population genetic implications from sequence variation in four Y chromosome genes. *Proc. Natl. Acad. Sci. USA* 97, 7354–7359 (2000)
315. Rose, H., Rose, S.: Give us the proof. *New Scientist* 166, 40–43 (2000)
316. Li, W.H., Chen, F.C.: The dawn of man. *Time Mag.* (July 23, 2001)
317. Roberts, E., Eargle, J., Wright, D., Luthey-Schulten, Z.: MultiSeq: Unifying sequence and structure data for evolutionary analysis. *BMC Bioinformatics* 7, 382–1–11 (2006)
318. Ingman, M., Kaessmann, H., Pääbo, S., Gyllensten, U.: Mitochondrial genome variation and the origin of modern humans. *Nature (London)* 408, 708–713 (2000)
319. Zhang, Y., Chen, W.: A new approach to molecular phylogeny of primate mitochondrial DNA. *MATCH Commun. Math. Comput. Chem.* 59, 625–634 (2008)
320. Cooper, A., Mourer-Chauviré, C., Chambers, G.K., von Haeseler, A., Wilson, A.C., Pääbo, S.: Independent origins of New Zealand moas and kiwis. *Proc. Natl. Acad. Sci. USA* 89, 8741–8744 (1992)
321. Foley, R.: Hominid species and stone-tool assemblages: How are they related? *Antiquity* 61, 380–392 (1987)
322. Wood, B.: Origin and evolution of the genus *Homo*. *Nature (London)* 355, 783–790 (1992)
323. Krings, M., Stone, A., Schmitz, R.W., Krainitzki, H., Stoneking, M., Pääbo, S.: Neanderthal DNA sequences and the origin of modern humans. *Cell* 90, 19–30 (1997)
324. Ovchinnikov, I.V., Götherström, A., Romanova, G.P., Kharitonov, V.M., Lidén, K., Goodwin, W.: Molecular analysis of Neanderthal DNA from the Northern Caucasus. *Nature (London)* 404, 490–493 (2000)
325. Aiello, L.C., Collard, M.: Our newest oldest ancestor? *Nature (London)* 410, 526–527 (2001)
326. Torrens, F., Castellano, G.: Periodic classification of human immunodeficiency virus inhibitors. In: Sidhu, A.S., Dillon, T.S., Chang, E. (eds.) *Biomedical Data Applications*. Springer, Heidelberg (in press) (2008)
327. IMSL, Integrated mathematical statistical library (IMSL). IMSL, Houston (1989)
328. Tryon, R.C.: *J. Chronic. Dis.* 20, 511–524 (1939)

329. Jarvis, R.A., Patrick, E.A.: Clustering using a similarity measure based on shared nearest neighbors. *IEEE Trans. Comput.* 22, 1025–1034 (1973)
330. Page, R.D.M.: Program TreeView. Universiy of Glasgow (2000)
331. Huson, D.H.: SplitsTree: Analizing and visualizing evolutionary data. *Bioinformatics* 14, 68–73 (1998)
332. Hotelling, H.: Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.* 24, 417–441 (1933)
333. Lipinski, C.A., Lombardo, F., Dominy, B.W., Feeney, P.J.: Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug. Deliv. Rev.* 23, 3–25 (1997)
334. Lipinski, C.A., Lombardo, F., Dominy, B.W., Feeney, P.J.: Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug. Deliv. Rev.* 46, 3–26 (2001)
335. Oprea, T.I., Davis, A.M., Teague, S.J., Leeson, P.D.: Is there a difference between leads and drugs? A historical perspective. *J. Chem. Inf. Comput. Sci.* 41, 1308–1315 (2001)
336. Ghose, A.K., Viswanadhan, V.N., Wendoloski, J.J.: A knowledge-based approach in designing combinatorial or medicinal chemistry libraries for drug discovery. *J. Combin. Chem.* 1, 55–68 (1999)
337. Talevi, A., Castro, E.A., Bruno-Blanch, L.E.: New solubility models based on descriptors derived from the detour matrix. *J. Argent. Chem. Soc.* 94, 129–141 (2006)
338. Duchowicz, P.R., Talevi, A., Bellera, C., Bruno-Blanch, L.E., Castro, E.A.: Application of descriptors based on Lipinski's rules in the QSPR study of aqueous solubilities. *Bioorg. Med. Chem.* 15, 3711–3719 (2007)
339. Ou, C.Y., Ciesielski, C.A., Myers, G., Bandea, C.I., Luo, C.-C., Korber, B.T.M., Mullins, J.I., Schochetman, G., Berkelman, R.L., Economou, A.N., Witte, J.J., Furman, L.J., Satten, G.A., MacInnes, K.A., Curran, J.W., Jaffe, H.W.: Laboratory Investigation Group, Epidemiologic Investigation Group Molecular epidemiology of HIV transmission in a dental practice. *Science* 256, 1165–1171 (1992)
340. Rouzine, I.M., Coffin, J.M.: Search for the mechanism of genetic variation in the pro gene of human immunodeficiency virus. *J. Virol.* 73, 8167–8178 (1999)
341. De Oliveira, T., Pybus, O.G., Rambaut, A., Salemi, M., Cassol, S., Ciccozzi, M., Rezza, G., Gattinara, G.C., d'Arrigo, R., Amicosante, M., Perrin, L., Colizzi, V., Perno, C.F., Benghazi Study Group: HIV 1 and HCV sequences from Libyan outbreak. *Nature (London)* 444, 836–837 (2006)

Intelligent Finite Element Method and Application to Simulation of Behavior of Soils under Cyclic Loading

A.A. Javadi¹, T.P. Tan², and A.S.I. Elkassas³

¹ University of Exeter

A.A.Javadi@ex.ac.uk

² University of Exeter

T.P.Tan@exeter.ac.uk

³ Ove Arup and Partners

Cardiff ahmed.elkassas@arup.com

Abstract. In this chapter a neural network-based finite element method is presented for modeling of the behavior of soils under cyclic loading. The methodology is based on the integration of a neural network in a finite element framework. In this method, a neural network is trained using experimental (or in-situ) data representing the mechanical response of material to applied load. The trained network is then incorporated in the finite element analysis to predict the constitutive relationships for the material. The development and validation of the method will be presented followed by the application to study of the behavior of soils under cyclic loading. The results of the analyses will be compared with those obtained from standard finite element analyses using conventional constitutive models. The merits and advantages of neural network-based constitutive models and the intelligent finite element model will be highlighted. It will be shown that the neural network-based constitutive models offer an effective and unified approach to constitutive modeling of materials with complex behavior in finite element analysis of boundary value problems.

1 Introduction

Finite element method has, in recent years, been widely used as a powerful tool in the analysis of engineering problems. In this numerical analysis, the behavior of the actual material is approximated with that of an idealized material that deforms in accordance with some constitutive relationships. Therefore, the choice of an appropriate constitutive model that adequately describes the behavior of the material plays an important role in the accuracy and reliability of the numerical predictions. During the past few decades several constitutive models have been developed for various materials. Most of these models involve determination of material parameters, many of which

have no physical meaning [1]. Despite considerable complexities of constitutive theories, due to the erratic and complex nature of some materials such as soils, rocks, composites, etc., none of the existing constitutive models can completely describe the real behavior of these materials under various stress paths and loading conditions.

In conventional constitutive material modeling, an appropriate mathematical model is initially selected and the parameters of the model (material parameters) are identified from appropriate physical tests on representative samples to capture the material behavior. When these constitutive models are used in finite element analysis, the accuracy with which the selected material model represents the various aspects of the actual material behavior and also the accuracy of the identified material parameters affect the accuracy of the finite element prediction.

In the past few decades, attempts have been made by a number of researchers to use artificial neural networks to model the constitutive material behavior. The application of ANN for constitutive modeling of concrete was first proposed by Ghaboussi, Garret and Wu (1991). Ghaboussi and Sidarta (1998) presented an improved technique of ANN approximation for learning the mechanical behavior of drained and undrained sand. Millar, Clarici, Calderbank and Marsden (1995) used neural network for prediction of stress-strain behavior of rocks. The role of ANN in constitutive modeling was also studied by a number of other researchers including the authors. These works indicated that neural network based constitutive models can capture nonlinear material behavior with a high accuracy. These models are versatile and have the capacity to continuously learn as additional material response data become available.

The role of the ANN is to attribute a given set of output vector(s) to a given set of input vectors. When applied to the constitutive description, the physical nature of these input-output data is clearly determined by the measured quantities, e.g., stresses, strains, pore pressures, temperature, etc. In this case, an unknown conventional analytical constitutive description can be directly replaced with a suitably trained neural network. The source of knowledge for ANN is not a symbolic formula but a set of experimental data.

It has been shown that the neural network-based constitutive models can be incorporated in finite element (or finite difference) codes. Shin and Pande (2000) presented a hybrid FE-ANN code and showed that the application of a constitutive law in the form of a neural operator leads to some qualitative improvement in the application of finite element method in engineering practice. They presented a procedure where data for training neural network-based constitutive model (NNCM) were acquired from planned monitoring of structural tests. Lefic and Schrefler (2003) used a neural network for constitutive modeling of nonlinear material behavior and highlighted some of the difficulties in the constitutive description in incremental form. Hashash, Jung and Ghaboussi (2004) described some of the issues related to the numerical implementation of a NNCM in finite element analysis and derived

a closed-form solution for material stiffness matrix for the neural network-based constitutive model. The authors have carried out extensive research on application of neural networks in constitutive modeling of complex materials in general and soils in particular. They have developed an intelligent finite element method (NeuroFE code) based on the incorporation of a back propagation neural network (BPNN) in finite element analysis [2, 3]. The intelligent finite element model has been applied to a wide range of boundary value problems and has shown that ANNs can be very efficient in learning and generalizing the constitutive behavior of complex materials.

In material modeling using ANN, the raw experimental (or in-situ) test results are directly used for training the neural network. In this way, the neural network learns the patterns of the material behavior during the training process and stores the information as the weighting coefficients of the network. If the training data contains enough relevant information, the trained network should be able to generalize the material behavior to new loading conditions.

Although it has been shown that ANNs offer great advantages in constitutive modeling of materials in finite element analysis, majority of the applications so far have been limited to simple boundary value problems and relatively simple aspects of material behavior have been studied so far. This chapter focuses on the application of the intelligent finite element method, developed by the authors, to the simulation of behavior of soils under cyclic loading. The development and validation of the intelligent finite element method are presented and the efficiency of the methodology is examined by application to the complex problem of cyclic loading of soils. It is shown that the proposed methodology can simulate the real behavior of complex materials under cyclic loading with very high accuracy. The main advantages of using a neural network approach are highlighted.

2 Behavior of Soils under Cyclic Loading

Cyclic strain history is an important factor in studying the deformation of soils under cyclic loading. This is especially critical when analyzing the behavior of soil under seismic or earthquake loading. Various research efforts have been dedicated to study the behavior of soils subjected to cyclic loading. Moses, Rao and Rao (2003) conducted research on the behavior of cemented Indian costal marine clay under cyclic loading. Usaborisut, Koike, Bahalayodhin, Niyamapa and Yoda (2001) carried out a similar investigation into the behavior of Bangkok clay. This work involved application of cyclic torsional shear loading to simulate the loading corresponding to turning operation of a tractor in the field and also static loading using triaxial tests. Rao and Panda (1998) studied the nonlinear and hysteretic behavior of soil under cyclic loading conditions. They compared their results with the existing results of two marine clays tested under undrained cyclic triaxial and simple shear conditions. In addition, some research work has been carried out on cyclic loading of soils with the aim of understanding the soil behavior during seismic or

earthquake activity. Yilmaz, Pekcan and Bakir (2004) studied the behavior of a silty clay under cyclic loading through a series of stress-controlled triaxial tests on anisotropically consolidated natural soil samples. The samples were representative of the soil conditions in the city of Adapazari in Turkey which was subjected to an earthquake of magnitude 7.4 in Richter scale in August 1999. Osinov (2003) presented a mathematical model for deformation of soils under irregular cyclic loading and applied the model to study the dynamic deformation and liquefaction of the soil at the Port Island site during the 1995 Hyogoken-Nambu earthquake.

2.1 Finite Element Modeling of Behavior of Soils under Cyclic Loading

Finite element method (FEM) is a well-established engineering tool with wide range of applications in all disciplines of engineering. One of the principal advantages of the FEM is that it offers a unified approach to the solution of diverse engineering problems. The finite element method has been used in many fields of engineering practice over the past 30 years however, only recently it has begun to be widely used for the analysis of geotechnical engineering problems. This is mainly because of many issues that are related to the complex behavior of geological materials (i.e., soils and rocks).

In particular, considering the complexity of the task of site characterization and selection of an appropriate constitutive model and material parameters for soil under cyclic loading, a very complicated finite element model would be required to accurately model various aspects of behavior of soil. The success of such a finite element model will be largely dependant on the constitutive model used for the soil which should cover the variations in soil properties during cyclic loading. Furthermore, these procedures are dependant upon material parameters that are usually difficult to estimate and as a result, limited success has been achieved in producing results that are comparable to field observations.

In this chapter a fundamentally different approach is presented for constitutive modeling of soils under cyclic loading. In the proposed method, an artificial neural network is trained with data from results of cyclic loading tests on samples of soil representative of the in situ conditions. The trained network is then incorporated into the intelligent finite element which in turn can be used to analyse the behavior of the in-situ soil under cyclic loading. The training and generalization capabilities of the neural network in extending the learning to cases of multiple, variable and irregular cycles are investigated.

3 Artificial Neural Network

A neural network is an information processing technique developed by inspiration by the way biological nervous systems, such as the brain, process information. Neural networks are composed of a large number of highly interconnected processing elements, or neurons, usually arranged in layers. These

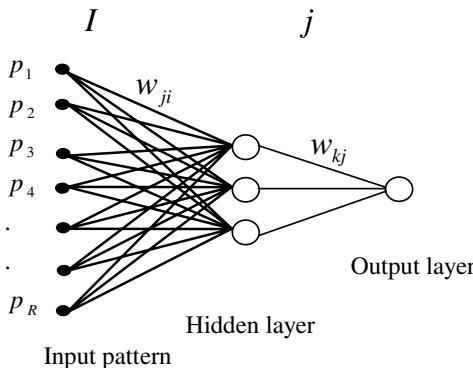


Fig. 1. Structure of a typical feedforward back-propagation neural network

layers generally include an input layer, a number of hidden layers and an output layer. Signals that are generated from the input propagate through the network on a layer-by-layer basis in the forward direction. Neurons in hidden layers are used to find associations within the input data and extract patterns than can provide meaningful outputs. A neural network system uses the human-like technique of learning by examples to resolve problems. Just as in biological systems, learning involves adjustments to the synaptic connections that exist between the neurons. The output of each neuron, responding to a particular combination of inputs, has an influence (or weight) on the overall output. Weighting is controlled by the level of activation of each neurone, and the strength of connection between individual neurons. Patterns of activation and interconnection are adjusted to achieve the desired output from the training data. Corrections are based on the difference between actual and desired output, which is computed for each training cycle. If average error is within a prescribed tolerance the training is stopped, the weights are locked in and the network is ready to use [4].

Generally, there are two main types of learning: supervised and unsupervised. In supervised learning, the external environment also provides a desired output for each one of the training vectors whereas in unsupervised learning, the external environment does not provide the desired network output. In terms of topology, an ANN can be classified as a feedforward or feedback (also called recurrent) network [5]. In a feedforward neural network, a unit only sends its output to units from which it does not receive any input directly or indirectly (via other units). In a feedback network, however, feedback is allowed to exist.

In the present work, supervised learning has been adopted as the desired outputs are usually available from the experimental data on material behavior. A feedforward backpropagation neural network has been used to learn the constitutive relationships for the material from experimental data. Fig. 1 shows the structure of a typical feed forward back-propagation neural

network with one hidden layer. Each layer is composed of several neurons and these are fully connected to neurons of the succeeding layer. In each hidden layer and output layer, the processing unit (neuron) sums the input from the previous layer and applies an activation (transfer) function to compute its output to the next layer.

The net input to the j^{th} node in the hidden layer is calculated as [5]:

$$S_j = \sum_{i=1}^R w_{ji} p_i \quad (1)$$

where w_{ji} is the connection weight from node i in the input layer to node j in the hidden layer; p_i is the i^{th} input element and R is the number of input features. An activation function is then used to calculate the output of the nodes in the hidden layer. The output of the j^{th} node in the hidden layer is:

$$O_j = f(S_j) \quad (2)$$

where f is the activation function. For a sigmoidal activation function, we have:

$$O_j = 1/(1 + e^{-(S_j + b_j)/\theta_0}) \quad (3)$$

where b_j serves as a threshold or bias and θ_0 is the parameter relating to the shape of the sigmoid. Subsequently, the output from the hidden layer is used as input to the output node (or nodes in the next hidden layer). The input and output of the nodes in layer k (see Fig. 1) are calculated in the same way:

$$S_k = \sum_{j=1}^R w_{kj} O_j \quad (4)$$

$$O_k = f(S_k) \quad (5)$$

On the whole, the output of the network will not be the same as the desired target value. During cycles of training, the weights of the network are updated in such a way that the difference between the desired response and the computed response is minimized to a required threshold.

3.1 Neural Network for Constitutive Modeling

In conventional constitutive material modeling, initially an appropriate mathematical model is selected and the parameters of the model (material parameters) are then identified from appropriate physical tests on representative samples to capture the material behavior. There are inevitable noise and errors in such tests. As a result, the accuracy with which the selected material model represents various aspects of the material behavior affects the accuracy of the finite element prediction.

In the past few years, artificial neural networks have shown to have potential in constitutive modeling of materials. In constitutive modeling using

neural network, the raw experimental or in-situ test data are directly used for training the neural network. In this approach, there are no mathematical models to select and the neural network learns the constitutive relationships directly from the raw data during the training process. As a result, there are no material parameters to be identified and as more data become available, the material model can be improved by re-training of the ANN using the additional data. Furthermore, the incorporation of an ANN in a finite element procedure avoids the need for complex yielding/failure functions, flow rules, etc. A trained network can be incorporated in a finite element code/procedure in the same way as a conventional constitutive model. It can be incorporated either as incremental or total stress-strain strategies. In this study both the incremental and total stress-strain strategies have been successfully implemented in the intelligent finite element model.

4 Intelligent Finite Element Method

An intelligent FEM has been developed based on the integration of a back propagation neural network (BPNN) in a finite element framework [2]. In the developed methodology, the neural network is used as a unified constitutive model for materials in finite element analysis. Fig. 2 shows the flowchart of the developed intelligent finite element method (NeuroFE code). As shown in the figure, the neural network replaces the role of more conventional constitutive models. When a neural network is used for constitutive description, the physical nature of the input-output data for the ANN is clearly determined by the measured quantities, e.g., stresses, strains, pore pressures, temperature, etc. In this case, an unknown conventional analytical constitutive description is directly replaced with a suitably trained neural network. The source of knowledge for ANN is not a symbolic formula but a set of experimental data. A neural network is trained using the raw experimental (or in-situ) data representing the mechanical response of the material to applied load.

In contrast to traditional mathematical constitutive models, the features of the material behavior are not explicitly represented by neural network-based constitutive model. However, if a comprehensive set of data is available to train the neural network, the resulting material model should be of sufficient accuracy to represent various aspects of the behaviour of the material.

The neural network is capable of learning the constitutive behavior of material and generalizing it to different conditions not necessarily introduced to the network during training. After training, the generalization capabilities of the network is examined using a set (or sets) of data that do not form a part of the training data. The trained network can then be incorporated in the finite element procedure to represent the constitutive relationship for the material. An intelligent FE code can be used for solving boundary value problems in the same way as a conventional FEM. In what follows, examples of applications of the developed intelligent finite element method to a number of boundary value problems are presented.

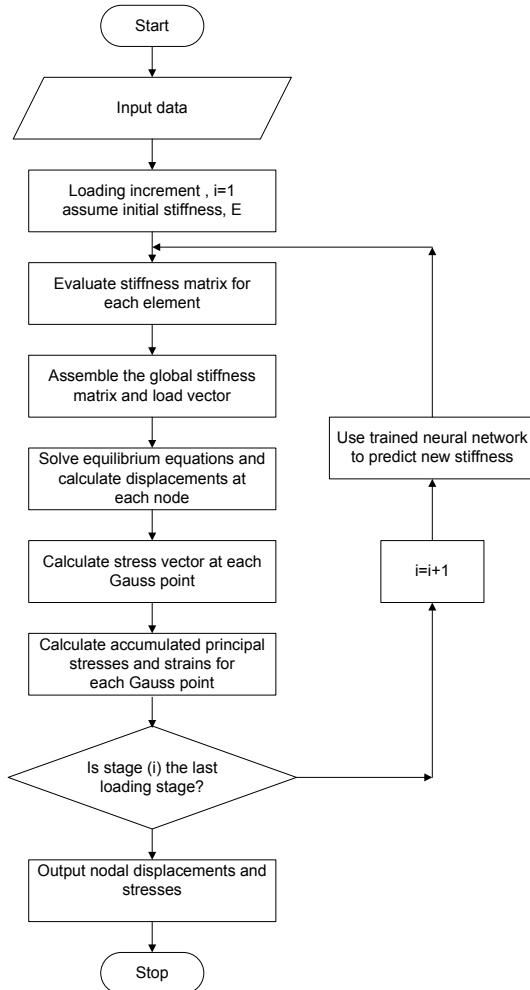


Fig. 2. Flow chart of the intelligent FE model

5 Numerical Examples

To illustrate the developed computational methodology, three numerical examples of application of the developed intelligent finite element method to engineering problems are presented. In the first example, the application of the methodology to a simple case of linear elastic material behavior is examined. In the second example, the method is applied to a boundary value problem involving the analysis of stresses and strains around a tunnel considering nonlinear and elasto-plastic material behavior. In the third example, the application of the method to the analysis of the behavior of soil under cyclic loading is presented.

5.1 Example 1

This example involves a thick circular cylinder conforming to plane strain conditions. Fig. 3 shows the geometric dimensions and the element discretization employed in the solution and it is seen that 9 parabolic isoparametric elements have been used. The cylinder is made of linear elastic material with a Young's modulus of $E=1000$ and a Poisson's ratio of 0.3 [6]. This example was deliberately kept simple in order to verify the computational methodology by comparing the results to available analytical solutions as well as those of a linear elastic finite element model. The loading case considered involves an internal pressure of 10 with boundary conditions as shown in Fig. 3.

Fig. 4a shows a linear elastic stress-strain relationship with a slope of 1000 that represents elastic modulus of 1000 for the material. The data from this figure were used to train the neural network. A neural network was trained to capture the linear stress-strain relationship for the material.

The architecture of the neural network used consisted of an input layer, a hidden layer and an output layer with linear transfer functions between the (input-hidden and hidden-output) layers. Fig. 4b shows the stress-strain relationship as predicted by the trained neural network, together with the original one. It is seen that after training, the neural network has successfully captured the stress-strain relationship with a precise accuracy.

The intelligent finite element model incorporating the trained neural network was used to analyze the behavior of the cylinder under the applied internal pressure. The results are compared with the analytical solutions and well as those obtained using standard linear elastic finite element method in Fig. 5. The figure shows the tangential stresses, radial displacements and radial stresses along a radius of the cylinder, predicted by the three methods. Comparison of the results shows that the results obtained using the Intelligent FEA are in excellent agreement with those attained from the

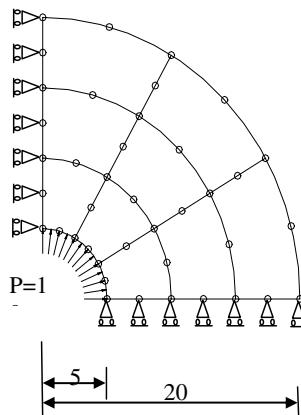


Fig. 3. FE mesh in symmetric quadrant of a thick cylinder

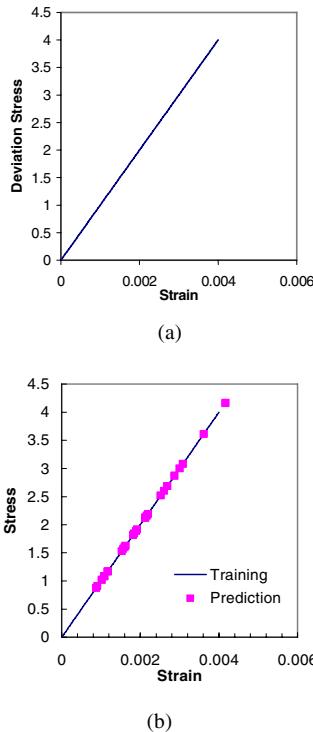


Fig. 4. (a) linear stress-strain relationship used for training, (b) Comparison between stress-strain relationships for training and prediction

analytical solution as well as the standard finite element analysis. This example shows the potential of the developed intelligent finite element method in deriving implicit constitutive relationships from raw data using a neural network and using these relationships to solve boundary value problems.

5.2 Example 2

The second example involves the analysis of the stresses and strains in a tunnel subjected to gravity loading. The geometry of the tunnel and the finite element mesh are shown in Fig. 6. The finite element mesh includes 96 eight-noded iso-parametric elements and 336 nodes. The depth of the tunnel crown from the ground surface is 12 m. The results from a series of triaxial tests were used in this example for the training of the neural network with an incremental stress-strain (tangential stiffness) strategy. It was assumed that the soil tested was representative of the material of the tunnel. The test data were arranged as shown in Tab. 1 and used to train an artificial neural network to model the stress-strain relationship for the soil. The neural

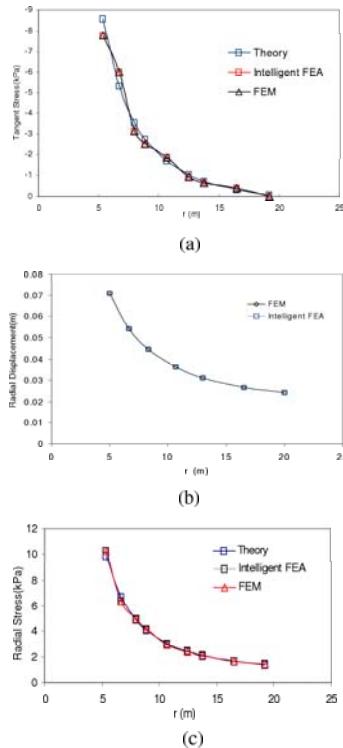


Fig. 5. Comparison of the results of the Intelligent FEA, conventional FEA and analytical solution

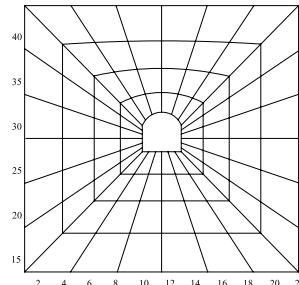


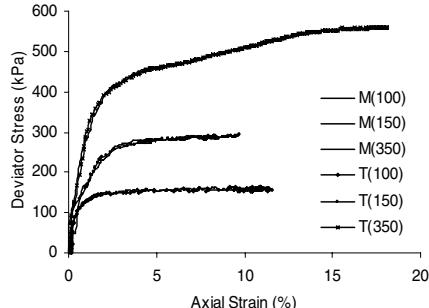
Fig. 6. Geometry of the tunnel and the FE mesh

network comprised 3 hidden layers with 3, 5 and 4 neurons in the first, second and third hidden layers respectively.

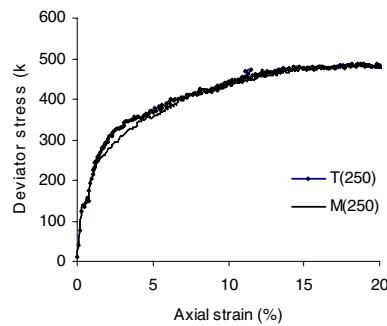
Fig. 7a shows the results of the training of the neural network. The neural network was trained with data from the tests conducted at confining pressures of 100, 150 and 350 kPa. It is clearly seen that, the neural network was able

Table 1. Input and output parameters used for training the artificial neural network

Input parameters					Output parameters	
p'	q	ε_v	ε_1	$\Delta\varepsilon_1$	Δq	



(a)



(b)

Fig. 7. (a) Results of training of the ANN, (b) stress-strain relationship predicted by the trained ANN (M =measured, T =predicted by neural network, the numbers in the brackets show the confining pressures in kPa)

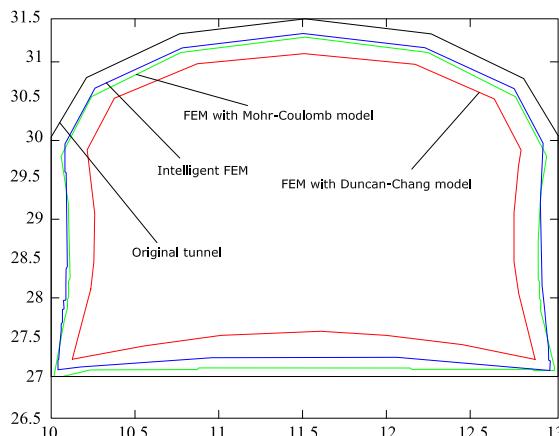
to capture the constitutive (nonlinear) stress-strain relationship for the soil with very good accuracy. The training process required 8000 training epochs. The generalization capability of the trained neural network is shown in Fig. 7b. The data from the test conducted at the confining pressure of 250kPa (which did not form a part of the training data) were used to test the trained neural network. The predicted output values of the trained neural network are compared with the experimentally measured values in Fig. 7b. It is seen that the generalization capability of the trained neural network is excellent.

Table 2. Material parameters for Duncan-Chang (1970) and Mohr-Coulomb models

$C'(kPa)$	$\phi'(deg)$	R_f	K	n	ν	$\gamma(kN/m^3)$
20	30	0.8121	159.44	0.6233	0.33	17

This shows that the neural network was trained sufficiently to adequately model the stress-strain behavior of the soil. The trained neural network was incorporated in the intelligent finite element (NeuroFE) code. The intelligent FE code incorporating the trained neural network was then used to simulate the behavior of the tunnel under gravity loading. For the conventional finite element analyses, the results of the triaxial tests were used to derive the material parameters for the Mohr-Coulomb and Duncan-Chang models for the soil (see Tab. 2).

Fig. 8 shows the comparison between the displacements in the tunnel predicted using standard finite element analyses using the Duncan-Chang nonlinear elastic and Mohr-Coulomb elasto-plastic models as well as the intelligent finite element method where the raw data from the triaxial tests were directly used in deriving the neural network-based constitutive model. The patterns of deformation are similar in all 3 analyses. FE analysis using the Duncan-Chang model seems to have over-predicted the displacements in the tunnel whereas the results of the intelligent FE analysis and the FE analysis using Mohr-Coulomb model are very close. Despite the relatively small difference between the results from the different analyses, it can be argued that the intelligent FE results are more reliable, as this method used the original raw experimental data to learn the constitutive relationships for the material

**Fig. 8.** Comparison of the results of the intelligent FEA and conventional FE analyses using Mohr-Coulomb and Duncan-Chang models

and it did not assume a priori any particular constitutive relationships, yield conditions, etc.

From the results obtained, it is shown that the developed intelligent finite element method is also capable of capturing more complex constitutive relationships of materials and can offer very realistic prediction of the behavior of structures.

5.3 Example 3: Behavior of Soil under Cyclic Loading

In this example, the behavior of a soil is studied in triaxial tests under cyclic axial loading. The test data for this example were generated by numerical simulation of triaxial experiments. In general, generating data by numerical simulation has many advantages including: (i) it is more economic (ii) it is far less time demanding, (iii) it can simulate loading paths and test conditions that cannot be easily achieved in physical testing due to physical constraints of the testing equipment. The data for training and validation of the neural network were created by finite element simulation of triaxial cyclic loading tests at constant cell pressures using the modified Cam caly model. The material parameters assumed for the soil are:

$\lambda = 14$ (slope of the virgin consolidation line),
 $\kappa = 0.015$ (slope of the unloading/reloading lines in the $v:Lnp'$ space),
 $M = 0.8$ (slope of the critical state line in the $q:p'$ space),
 $P_0 = 100.0$ kPa (isotropic preconsolidation pressure indicating the size of the yield surface),
 $N = 2.68$ (intercept of the virgin consolidation line with vertical axis).

The generated data were used to train a neural network and the trained network was then incorporated in the intelligent finite element model to represent the soil constitutive behavior under cyclic loading. The results of the intelligent finite element analyses were compared with those attained using conventional finite element method. The performance of the model was evaluated for three separate cases of loading where the soil was subjected to:

- 1-one cycle of loading and unloading;
- 2-two cycles of loading and unloading; and
- 3-multiple (and irregular) cycles of loading and unloading.

The data generated by numerical simulation of the cyclic loading tests at confining pressures of 100, 150, 200, 250 and 300kPa are shown in Fig. 9. In order to introduce a level of noise that inevitably exists in real triaxial test data, numerical simulation for each confining pressure was repeated by changing the total number of load increments in the simulation and the obtained data were combined and used in training of the neural network. Fig. 10 shows typical results of tests conducted at confining pressure of 150kPa with 4 different load increments.

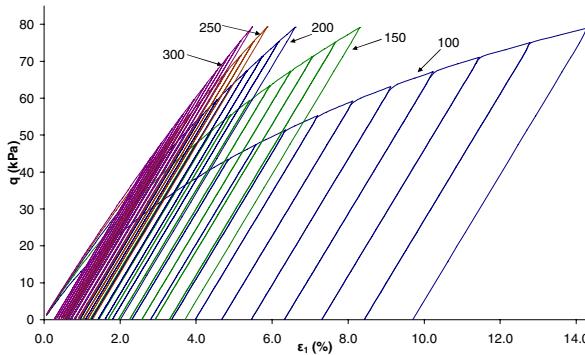


Fig. 9. Typical cyclic loading test data used for training and validation of neural network

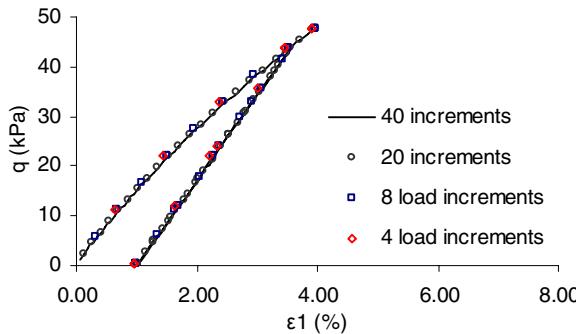


Fig. 10. Typical cyclic loading test results with different load increments at confining pressure of 150 kPa

The data from the tests at confining pressures of 100, 200 and 300KPa were used for the training of the neural network. The trained network was validated using the data from the test at the confining pressure of 250KPa. The input and output parameters used for training of the network are presented in Tab. 3. In the table p' is net mean stress, q is deviator stress, ε_v is volumetric strain and ε_1 is axial strain. The indices t , $t-1$, $t-2$ and $t+1$ represent the state of stress or strain at current (increment) step, previous step, two steps before and the next step respectively.

Data from the two previous load increment steps were provided to the network to help learn the loading history of the soil during transition from loading to unloading and vice versa. The relative values of the stress and strain variables in increments $t-2$, $t-1$ and t indicate whether the soil is on a loading path, unloading path or in transition from loading to unloading or vice versa. The neural network used comprised two hidden layers with eight

Table 3. Input and output parameters used for training and testing of the neural network

Inputs	Output
$p'_{(t-2)}$	
$p'_{(t-1)}$	
$p'_{(t)}$	
$q_{(t-2)}$	
$q_{(t-1)}$	
$q_{(t)}$	$\Delta\varepsilon_{1(t+1)}$
$\varepsilon_v(t-2)$	
$\varepsilon_v(t-1)$	
$\varepsilon_v(t)$	
$\varepsilon_1(t-2)$	
$\varepsilon_1(t-1)$	
$\varepsilon_1(t)$	
$\Delta q_{(t+1)}$	

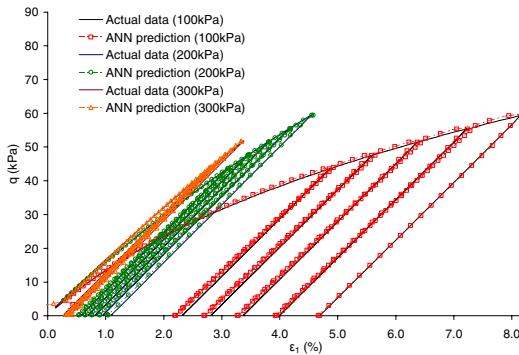


Fig. 11. Results of training of the neural network

and five neurons in the first and second layers respectively. The network was trained with 6000 cycles of training (epochs).

Fig. 11 shows the results of training of the neural network. In the figure, the actual (numerically simulated) data are plotted together with the results of the neural network prediction. It is seen from the figure that the network was capable of learning, with very good accuracy, the constitutive relationship of the soil under cyclic loading paths. The trained network was validated using a data set corresponding to the confining pressure of 250kPa. The results of the validation tests are shown in Fig. 12. It is shown that the trained neural network was able to generalize the training to loading cases that were not introduced to the network during training. After training and validation, the network was incorporated in the intelligent finite element model. The model was then used to simulate the behavior of the soil in triaxial cyclic loading tests at

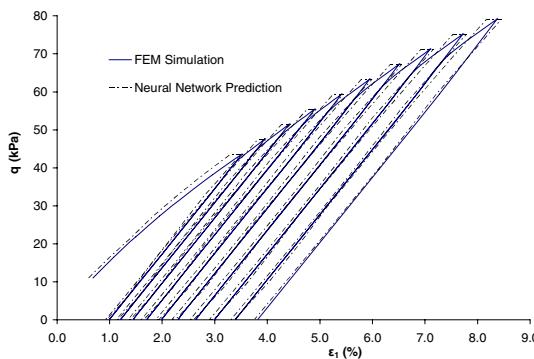


Fig. 12. Results of the validation of the trained neural network: comparison between the actual (numerically simulated) data and the neural network predictions for confining pressure of 250 kPa

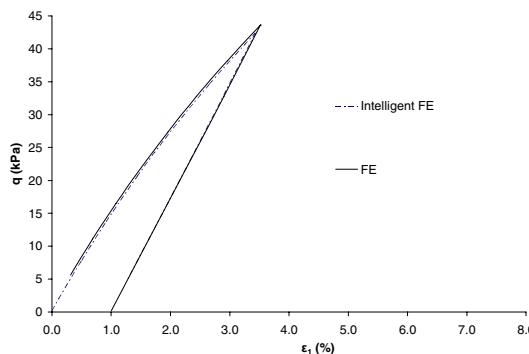


Fig. 13. Comparison between the results of the intelligent FEA and a conventional FEA for a single loading cycle

a confining pressure of 150KPa. Three different cases were simulated using the intelligent FE model and the results were compared with those obtained from a conventional FE simulation using the modified Cam clay model.

One Loading Cycle

In the first case, the intelligent FE model was used to simulate a triaxial test on a sample of the soil subjected to a single loading cycle at a confining pressure of 150KPa. The loading cycle involved the application of a total axial displacement of 2.2mm followed by unloading. The results of the intelligent finite element analysis are compared with those attained using the conventional FE simulation in Fig. 13. It is seen that the results of the intelligent FE model are in very close agreement with those of the conventional FE

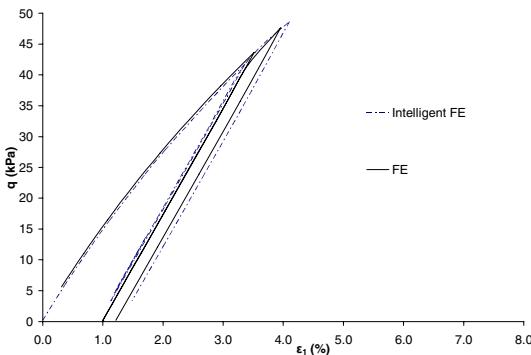


Fig. 14. Comparison between the results of the intelligent FEM and a conventional FEM for two loading cycles

simulation. It can be seen from the figure that, the intelligent FE model is capable of solving boundary value problems involving cyclic loading with a very high accuracy.

Two Loading Cycle

In the second case, the intelligent FE model was used to simulate of the behavior of the soil under two successive cycles of loading and unloading at a constant confining pressure of 150KPa. The loading involved the application (and removal) of displacements of 2.2mm and 2.4mm in the first and second cycles respectively. Fig. 14 shows a comparison between the results obtained using the intelligent FE model and those attained by the conventional FE method. From the figure, it can be seen that the intelligent FEM has produced results that are very close to those of the conventional FEM for the two cycles.

Multiple (and irregular) Loading Cycles

In the first two cases, all the simulations (including those used for training and testing of the ANN) were performed with a regular loading pattern involving regular induced displacements in the cycles. This case is set out to examine if the intelligent FE model, trained with regular cyclic loading data, would be able to generalize the training to predict the behavior of the soil for irregular loading patterns that are different from those used for training of the neural network. Although the loading pattern deferred from that used for training of the ANN, the imposed displacements (and loads) used in the simulation were kept within the ranges of values used for training so as to avoid extrapolation.

In this case, the intelligent FE model was used to simulate the behavior of the soil with an irregular cyclic loading pattern as shown in Fig. 15. The test was simulated at confining pressure of 150kPa that was not introduced to the neural network during training. The test involved the application (and

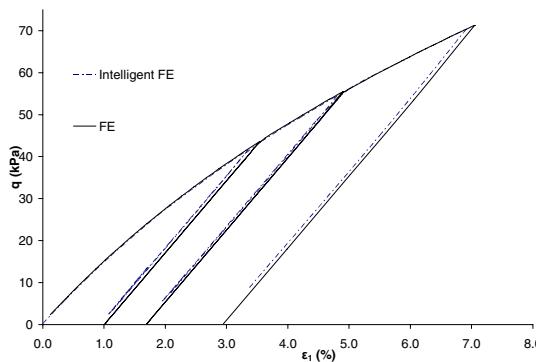


Fig. 15. Comparison between the results of the intelligent FEM and a conventional FEM for three irregular loading cycles

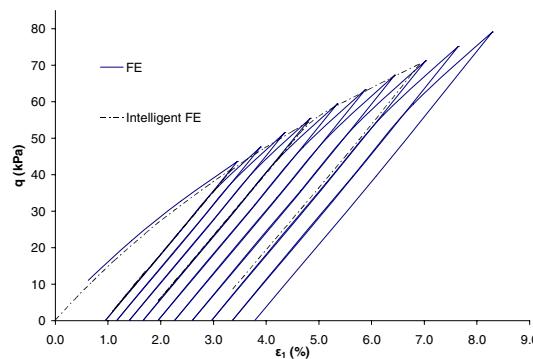


Fig. 16. Comparison between the results of the intelligent FEM for 3 irregular loading cycles and the original cycle loading data used for training

removal) of displacements of 2.2mm, 2.8mm and 3.6mm in the first, second and third cycles respectively. For the first cycle was a regular cycle. The second and third cycles were irregular cycles covering 3 and 4 regular cycles respectively (see Fig. 16). In Fig. 15, the results of the intelligent finite element analysis are compared with those attained using the conventional FE simulation of the same irregular pattern. From the figure, it can be seen that the results of the intelligent FE simulation are in a very good agreement with those obtained using the conventional FEM. The results are also compared with those obtained for a regular 10-cycle pattern with imposed displacements of 2.2, 2.4, 2.6, 2.8, 3.0, 3.2, 3.4, 3.6, 3.8 and 4.0mm in cycles 1 to 10 respectively (see Fig. 16). Comparison of the results shows that, although the neural network was only trained with data from regular cyclic loading tests, the intelligent finite element was able to predict the behavior of the soil under irregular loading patterns. It can be concluded that the intelligent FE model

is also capable of generalizing the behavior of the soil for cyclic loading with different loading and unloading patterns. This further illustrates the robustness of the proposed intelligent FE model and shows the excellent capability of the method in capturing the underlying constitutive relationships for the material from raw data and generalizing it to predict different conditions not introduced to the neural network during training.

6 Discussion and Conclusion

This chapter presented a fundamentally different approach to constitutive modeling of materials in finite element analysis. An intelligent finite element method was presented for modeling engineering problems. The method is based on the integration of a neural network in the finite element procedure. In the developed methodology, the ANN is used as a universal constitutive model for materials in finite element analysis.

The efficiency of the intelligent FE method was illustrated by successful application to a number of boundary value problems. The results of the analysis were compared with those attained from conventional FE analyses using some of the most commonly used constitutive models as well as analytical solutions in some cases. The model was also used to study the behavior of soils under cyclic loading. It was shown that neural network can learn the complex behavior of soils under cyclic loading taking into account the loading history of the soil. It was also shown that the intelligent finite element model, incorporating a NNCM trained with certain patterns of loading and unloading, can be used to predict the behavior of the soil under different (regular or irregular) loading patterns with a very high accuracy.

The main benefits of using a neural network approach are that it provides a unified approach to constitutive modeling of all materials (i.e., all aspects of material behavior can be implemented within a unified environment of a neural network); it does not require any arbitrary choice of the constitutive (mathematical) models; the incorporation of a neural network based constitutive model in a finite element procedure avoids the need for complex yielding/plastic potential/failure functions, flow rules, etc.; there is no need to check yielding, to compute the gradients of the plastic potential curve or to update the yield surface; there are no material parameters to be identified and the network is trained directly from experimental data. The neural network is capable of learning the material behavior directly from raw experimental data, therefore, NNCM is the shortest route from experimental research (data) to numerical modeling. The ANN model is simple and effective if appropriate experimental data are available. Another advantage of neural network based constitutive model is that as more experimental data become available, the quality of the neural network prediction can be improved by learning from the additional data, and therefore, the NNCM can become more effective and robust.

A trained network can be incorporated in a FE code/procedure in the same way as a conventional constitutive model. It can be incorporated either as incremental or total stress-strain strategies. An intelligent FE method can be used for solving boundary value problems in the same way as a conventional FEM.

It should be noted that, for practical problems, the data used for training of NNCM should cover the range of stresses and strains that are likely to be encountered in practice. This is due to the fact that neural networks are good at interpolation but not so good at extrapolation. Therefore, any attempt to use intelligent finite element method for loading conditions that may lead to stresses or strains outside the range of the stresses and strains used in training of the neural network may lead to unacceptable errors.

The method presented in this chapter provides a unified approach for constitutive modeling of materials in finite element analysis. It should be noted that the approach presented in this chapter is generic and can be applied to any type of material.

References

- Shin, H.S., Pande, G.N.: On self-learning finite element codes based on response of structures. *J. Computers and Geotechnics* 27, 161–171 (2000)
- Javadi, A.A., Tan, T.P., Zhang, M.: Neural network for constitutive modeling in finite element analysis. *J. Computer Assisted Mechanics and Engineering Sciences* 10, 523–529 (2003)
- Javadi, A.A., Tan, T.P., Elkassas, A.S.I., Zhang, M.: An Intelligent Finite Element Method: Development of the Algorithm. In: Proceedings of the 6th World Congress on Computational Mechanics. Tsinghua University Press /Springer, Beijing (2004)
- Javadi, A.A., Farmani, R., Tan, T.P.: An intelligent self-learning Genetic Algorithm; development and engineering applications. In: International Journal of Advanced Engineering Informatics (in print) (2005)
- Pao, Y.H.: Adaptive pattern recognition and neural networks. Addison-Wesley Publishing Company, USA (1989)
- Hinton, E., Owen, D.R.J.: Finite element programming. Academic Press, London (1977)
- Duncan, J.M., Chang, C.Y.: Nonlinear analysis of stress and strain in soils. *ASCE Journal of Soil Mechanics and Foundations Division SM5*, 1629–1653 (1970)
- Ghaboussi, J., Garret, J.H., Wu, X.: Knowledge-based modeling of material behavior with neural networks. *Journal of Engineering Mechanics Division (ASCE)* 117, 32–153 (1991)
- Ghaboussi, J., Sidarta, D.: A new nested adaptive neural network for modeling of constitutive behavior of materials. *J. Computers and Geotechnics* 22, 29–52 (1998)
- Hashash, Y.M., Jung, S., Ghaboussi, J.: Numerical implementation of a neural network based material model in finite element analysis. *Int. Journal for Numerical Methods in Engineering* 5, 989–1005 (2004)

11. Lefik, M., Schrefler, B.: Artificial neural network as an incremental nonlinear constitutive model for a finite element code. *J. Computational Methods in Applied Mechanical Engineering* 192, 3265–3283 (2003)
12. Millar, D.L., Clarici, E., Calderbank, P.A., Marsden, J.R.: On the practical use of a neural network strategy for the modeling of the deformability behavior of Croslands Hill sandstone rock. In: *Proceedings of the APCOM XXV Conference*, Brisbane, pp. 457–465 (1995)
13. Moses, G.G., Rao, S.N., Rao, P.N.: Undrained strength behavior of a cemented marine clay under monotonic and cyclic loading. *J. Ocean Engineering* 30, 1765–1789 (2003)
14. Osinov, V.A.: Cyclic shearing and liquefaction of soil under irregular loading: an incremental model for the dynamic earthquake-induced deformation. *J. Soil Dynamic and Earthquake Engineering* 23, 535–548 (2003)
15. Rao, S.N., Panda, A.P.: Non-linear analysis of undrained cyclic strength of soft marine clay. *J. Ocean Engineering* 26(3), 241–253 (1998)
16. Usaborisut, P., Koike, M., Bahalayodhin, B., Niyamapa, T., Yoda, A.: Cyclic torsional shear loading test for an unsaturated, hollowed specimen using Bangkok clayey, soil. *J. Terramechanics* 38, 71–87 (2001)
17. Yilmaz, M.T., Pekcan, O., Bakir, B.S.: Undrained cyclic shear and deformation behavior os silt-clay mixture of Adapazari, Turkey. *J. Soil Dynamic and Earthquake Engineering* 24, 497–507 (2004)

An Empirical Evaluation of the Effectiveness of Different Types of Predictor Attributes in Protein Function Prediction

Fernando Otero¹, Marc Segond², Alex A. Freitas¹, Colin G. Johnson¹, Denis Robilliard², and Cyril Fonlupt²

¹ Computing Laboratory, University of Kent, Canterbury, CT2 7NF, UK
`{febo2,A.A.Freitas,C.G.Johnson}@kent.ac.uk`

² Laboratoire d’Informatique du Littoral, ULCO BP719, 62100 Calais, France
`{segond,robillia,cyril.fonlupt}@lil.univ-littoral.fr`

Summary. Many classification schemes for defining protein functions, such as Gene Ontology (GO), are organised in a hierarchical structure. Nodes near the root of the hierarchy represent general functions while nodes near the leaves of the hierarchy represent more specific functions, giving the flexibility to specify at which level the protein will be annotated. In a data mining perspective, hierarchical structures present a more challenging problem, since the relationship between nodes need to be considered. This chapter presents an empirical evaluation of different protein representations for protein function prediction in terms of maximizing predictive accuracy, investigating which type of representation is more suitable for different levels of the GO hierarchy.

1 Introduction

The recent exponential increase in the number of proteins being identified and sequenced using high throughput experimental approaches has lead to a growth in the number of uncharacterised proteins. Determining protein functions is a central goal of bioinformatics, and it is crucial to improve biological knowledge, diagnosis and treatment of diseases. While biological experiments are the ultimate methods to determine the function of proteins, it is not possible to perform a functional assay for every uncharacterised protein. This is due to time and financial constraints, together with the complex nature of these experiments. Hence, a need for using computational methods to assist the annotation of large amounts of protein data appeared. In particular, this presents a significant opportunity to apply data mining techniques to analyse and extract knowledge from biological databases.

In essence, bioinformatics refers to the research area that combines computational and statistical methods to manage and analyse biological data [1]. It became a very popular research field after the fully sequenced genomes of

numerous organisms enabled biologists to map, sequence and analyse individual genes and their protein products. The information acquired by biological experiments has helped to expand understanding about cellular biology, more specifically about biological functions of proteins.

Proteins are large and complex molecules, assembled from amino acids arranged in a linear sequence using information encoded in genes. Proteins perform most of the functions within a cell. For instance, almost all biological processes, including metabolism, need enzymes to catalyse chemical reactions in order to occur; transport proteins are involved in the movement of small molecules through membranes. The amino acid sequence contains all the information necessary to specify the three-dimensional structure of a protein, enabling the protein to perform its function.

Biological databases accumulate vast amounts of protein data, from protein sequences to three-dimensional structures. To facilitate both collaboration and standardization across different sources, biological databases employ controlled vocabularies (ontologies) to annotate protein sequences and features. Ontologies such as the Enzyme Commission (EC) [26], Gene Ontology [2] and SCOP [18] are organised in a hierarchical structure, allowing the annotation of proteins at a different level of detail. In a hierarchical structure, nodes at the top (near the root of the hierarchy) represent general details while nodes at the bottom (near the leaves of the hierarchy) represent more specific details, giving the flexibility to specify at which level the protein will be annotated. The hierarchy defines a parent-child relationship between nodes, where the child is a specialisation of the parent. In a data mining classification task perspective, hierarchical structures present a more challenging problem [8] than flat (single-layer) problems. It is generally more difficult to discriminate between specific classes represented by leaf nodes than more general classes represented by internal nodes, since the number of examples per leaf node tends to be smaller compared to internal nodes.

In this chapter, we apply data mining methods to induce a classification model which can be used to predict the function of uncharacterised proteins using the Gene Ontology functional classification scheme. The Gene Ontology is a complex case of hierarchical organisation, where its terms are arranged in a directed acyclic graph (DAG) structure. We are particularly interested in comparing the effectiveness of different protein representations in terms of maximizing predictive accuracy. Since the problem of discriminating between terms at deeper levels of the hierarchy is different from terms at higher levels, our focus is on investigating which type of representation is more suitable for different GO terms at different levels of the GO hierarchy.

The remainder of this chapter is organised as follows. Section 2 presents a brief introduction of the basic concepts of molecular biology involved in the problem of predicting protein functions. Section 3 describes the methodology for evaluating different protein representations, including data preparation. Section 4 presents the computational results. Finally, Section 5 presents the conclusion and future research directions.

2 Biological Background

The genetic information of living organisms is stored in DNA (deoxyribonucleic acid) molecules. DNA is a long molecule composed by a sequence of deoxyribonucleic bases, which are linked together by a backbone composed by deoxyribose sugar and phosphate groups. There are four possible nucleotide bases that are found in DNA: adenine (A), guanine (G), cytosine (C) and thymine (T). The DNA molecule structure is a double stranded helix, which is dependent on pairing between the nucleotide bases: adenine is able to pair with thymine, while guanine is able to pair with cytosine. Consequently, the strands in the double helix complement each other in an anti-parallel fashion.

The correlation between genes and proteins is that nucleotide sequences – corresponding to particular genes – in DNA molecules code for amino acid sequences of proteins. This relationship is part of the *central dogma* of molecular biology [1]. The central dogma states that the information flows from DNA to RNA (ribonucleic acid) to protein. In summary, this process works as follows (illustrated in Fig. 1). In the first step (transcription), the genetic information stored in a DNA sequence is used to create a mRNA (*messenger RNA*) molecule. In the second step (translation), this mRNA molecule is used as a template to synthesize proteins. A series of three nucleotides in the mRNA corresponds to a codon, which in turn corresponds to either a specific amino acid or a signal site (start/stop translation). For more details about this process refer to [1].

Proteins are involved in most biological activities and even make up the majority of cellular structures. Since proteins do almost all the work in a cell, understanding the roles of proteins is the key to understanding how the whole cell operates.

2.1 Proteins

Proteins are the building blocks from which every cell in an organism is built [1]. A protein molecule is assembled from a long sequence of amino

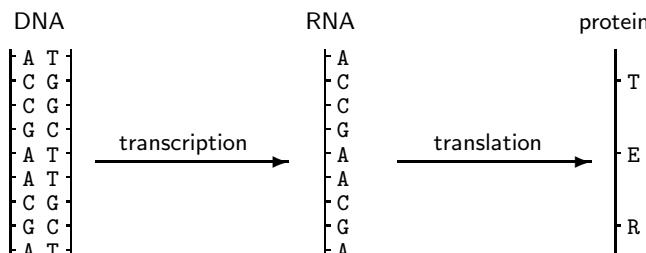


Fig. 1. The central dogma's information flow: from DNA to RNA to protein

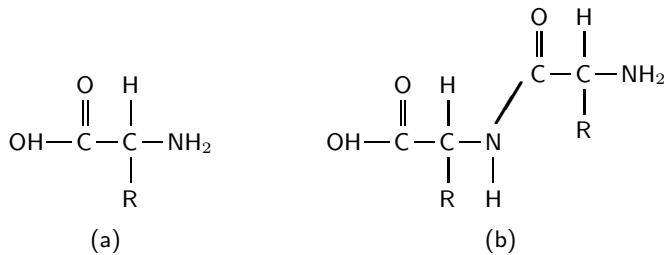


Fig. 2. In (a) basic amino acid structure; (b) The peptide bond between two amino acids (thick line). Amino acids are linked together by a peptide bond between their amino and carboxyl groups, constituting the protein's backbone. This process is repeated many times for polypeptide proteins.

acids using information encoded in genes. Each protein has its own unique sequence of amino acids, which is specified by the nucleotide sequence of the gene encoding the protein. There are 20 different types of amino acids, each with different biochemical properties, that can be found in a sequence. An amino acid is composed by a central carbon (C) – α carbon – attached to an hydrogen (H), an amino group (NH_2), a carboxyl group (COOH) and a variable side chain (R). There are 20 distinct side chains, resulting in 20 different types of amino acids. The amino acids are linked together by a peptide bond between their amino and carboxyl groups, constituting the protein's backbone (illustrated in Fig. 2). In general, proteins are 200-400 amino acids long.

The amino acid sequence of a protein is also known as the protein's *primary structure*. It determines the protein's three-dimensional structure and function. Subsequently bondings between the amino and carboxyl groups from different amino acids allow the linear sequence to fold into structures known as alpha helices and beta sheets. Alpha helices (α -helices) are formed when the backbone twists into right-handed helices. Beta sheets (β -sheets) are formed when the backbone folds back on itself in either a parallel or anti-parallel fashion. These structures constitute the protein's *secondary structure*. The three-dimensional shape of the whole protein is known as the protein's *tertiary structure*, which is defined by the spatial relationship between the secondary structures. The three-dimensional shape of a protein is crucial for its function, hence discovering its tertiary structure can provide important information about how the protein performs its function. Proteins are also capable of assembly into complex structures, known as the protein's *quaternary structure*, as a result of interaction between them. There are some proteins that can only be functional when associated in protein complexes [15].

The complexity of determining the different levels of protein structures increases from primary towards quaternary structures. For instance, the primary structure can be determined by translating the DNA sequence of

Table 1. Summary of biological databases available online

Name	Description
UniProt http://www.ebi.ac.uk/uniprot/	automatically (UniProtKB/TrEMBL) and manually (UniProtKB/Swiss-Prot) annotated protein sequences
IntAct http://www.ebi.ac.uk/intact/	protein interaction data
CATH http://www.cathdb.info/	hierarchical domain classification of protein structures
PROSITE http://www.expasy.org/prosite/	protein domains, families and functional sites
PubMed http://www.ncbi.nlm.nih.gov/pubmed/	biomedical literature
Pfam http://pfam.sanger.ac.uk/	protein domains and families
InterPro http://www.ebi.ac.uk/interpro/	protein families, domains and sequence patterns
DIP http://dip.doe-mbi.ucla.edu/	protein interaction data
MEDLINE http://medline.cos.com/	biomedical literature
TIGRFAMs http://www.tigr.org/TIGRFAMs/	protein families
PRINTS http://www.bioinf.manchester.ac.uk/dbbrowser/PRINTS/	protein families
ArrayExpress http://www.ebi.ac.uk/arrayexpress/	gene expression data

the gene that specifies the protein to an amino acid sequence, while the tertiary structure can be determined using complex X-ray crystallography experiments. Consequently, many more proteins sequences (primary structures) are known than proteins three-dimensional structures (tertiary structures). The process from which a protein in one-dimensional state (primary structure) turns to a three-dimensional state (tertiary structure) is called *folding*. While folding occurs spontaneously within a cell, its inherent details are not known.

2.2 Protein Databases

Protein information is widely available in biological databases. Some databases are dedicated to a particular aspect, such as structural information or protein interaction data, while others provide broad information with links to specialised databases. Table I presents a summary of biological databases. Most of these databases provide an online search interface.

In general, database entries comprise experimental results combined with annotations. Annotations provide valuable information about a protein, including simple information derived from proteins' primary structures (e.g molecular weight and sequence patterns), nature of the experiment and organism where the protein is found. They can be determined by computational methods or manually, where the latter is more preferable for its reliability. For instance, UniProt (Universal Protein Resource) contains two sections: Swiss-Prot and TrEMBL. Swiss-Prot is the richest annotated protein sequence section, containing manually annotated/curated entries, with extensive database cross-references and literature citations. The TrEMBL section contains computationally analysed records that await full manual annotation.

A commonly used source of protein annotation information are motif databases. Motifs are preserved amino acid sequences, which usually represent a protein family, domain or an activation site. PROSITE [12], PRINTS [3], Pfam [7] and InterPro [17] are examples of databases that contain a collection of protein motifs. Biological literature databases, such as MEDLINE (Medical Analysis and Retrieval System Online), are a valuable resource and textual analysis of these databases is an area of growing interest [22]. More specialised databases contain information about protein interaction data, protein secondary structures, gene expression data, among others.

2.3 Classification Schemes

Several classification schemes for defining protein function annotation exists, such as as the Enzyme Commission (EC) [26] scheme for enzyme classification and FunCat [24] for the functional description of proteins from diverse organisms. In order to make classification schemes open for computational processing, they usually employ a controlled vocabulary (ontology) to define protein functions. More complex schemes are hierarchically structured, allowing protein annotations at different levels, depending on the depth of knowledge about the protein in question.

The Gene Ontology (GO) Consortium [2] has developed ontologies to classify proteins in terms of three different domains: molecular function, biological process and cellular component. The ontologies are defined by a hierarchy of terms (categories), where each term has a unique numerical identifier and a textual description, arranged in a DAG-like structure. In DAG-based hierarchies, terms can have more than one parent, as opposed to just one parent in tree-based hierarchies. This makes the hierarchical classification problem a particularly challenging one.

Within the GO, the ontology is divided into three different *domains*, each of which consists of a vocabulary for a particular type of biological knowledge. The Molecular Function (MF) domain describes activities performed at the molecular level, generally accomplished by individual proteins. Examples of molecular functions defined are the general concept ‘transporter activity’ and its specialisation ‘ion transporter activity’, where the latter is represented as a child of transporter activity. The Biological Process (BP) domain describes activities accomplished by a series of events or molecular functions. Examples of activities defined are high-level processes ‘immune response’ and ‘reproductive process’. Finally, the Cellular Component (CC) domain describes locations, at the levels of subcellular structures and macromolecules complexes. In general, proteins are located in or are a subcomponent of a particular cellular component. Examples of locations defined are ‘plasma membrane’ and ‘golgi transport complex’.

In the GO hierarchy, parent-child relationships are governed by the *true path rule*. The true path rule states that the path from a child term towards top-level terms must always be true. In other words, if a protein is annotated with a term A, it automatically inherits the annotation of all ancestor terms of A. For example, a protein annotated with ‘ion transporter activity’ will inherit the annotation ‘transporter activity’, since ‘ion transporter activity’ is a specialisation of ‘transporter activity’. The hierarchical structure allows annotation of proteins at different levels, from general (parent) to more specific (child) terms, depending on the depth of knowledge about the protein in question. The GO classification scheme is currently the preferred approach for computational functional annotation [9].

2.4 Protein Function Prediction

As aforementioned, the exponential increase in the number of proteins being identified and sequenced using high throughput experimental approaches has lead to a growth in the number of uncharacterised proteins (proteins for which the function is unknown). Since the rate at which sequencing methods are producing data is far outperforming the rate at which biological methods can determine protein functions, there is a crescent interest in automated protein function prediction methods.

A commonly used approach is to assign a function by sequence similarity, using BLAST (Basic Local Alignment Search Tool) to perform a similarity search in a protein sequence database. This approach relies on the assumption that proteins with similar sequences perform similar functions. It has been shown that proteins with very different sequences may perform the same function, or proteins with very similar sequences may perform different functions ([10], [27], [25]). Furthermore, this approach is unsuitable if a similar protein with known function cannot be found.

Another approach is to apply data mining techniques to analyse and extract knowledge from biological databases. In this context, an example

(record, data instance) represents a protein, the attributes represent different features of a protein (i.e. sequence length, presence/absence of a particular motif) and the classes correspond to the different functions that the protein can perform. For instance, in [13] neural networks were trained using sequence derived features such as amino acid biochemical properties and secondary structure; [15] used protein interaction data to create a probabilistic model based on markov random fields; PROSITE patterns were used in [19] to extract classification rules using C4.5; and [29] followed a clustering approach using protein domains and textual information from MEDLINE. For further examples refer to [23], [5] and [9].

3 Methods

The data mining task addressed in this work is the classification task, where the goal is to predict the class (function) of an example (protein), given the values of a set of attributes for that example. In essence, the classification task consists of inducing a model from the data by observing relationships between predictor attributes and classes, which can be used later to classify new examples.

We have chosen different types of protein representations to be used as predictor attributes (detailed in Subsection 3.2) and a classification algorithm (detailed in Subsection 3.3) to induce a model for protein function classification. We are interested in evaluating those types of representations at different levels of the GO hierarchy, and this evaluation is performed by measuring the predictive accuracy obtained by the same classification algorithm when using each of the different types of protein representation.

3.1 Data Preparation

The selection of the protein examples was divided into three phases. In the first phase we selected a subset of the Gene Ontology hierarchy to represent the classes in our classification problem. As we are interested in ion channel proteins, all the ancestor and descendant terms of the GO:0005216 (*ion channel activity*) node were selected. Note that in the GO hierarchy one term can have more than one parent. For this reason, for every descendant (child) term of the node GO:0005216, we also retrieved its ancestor nodes. The reason for selecting ancestor terms of the GO:0005216 term was to increase the number of negative examples in the data sets. The class hierarchy after this phase was composed by 88 GO terms.

In the second phase, we retrieved protein interaction data from the IntAct database (release 15/12/2007). Records with database cross-references to the GO terms selected in the previous phase were retrieved. In summary, each record of the IntAct database contains the UniProt accession number and a list of interacting proteins which interact with the protein. It should be noted that interacting proteins do not necessarily have to belong to the

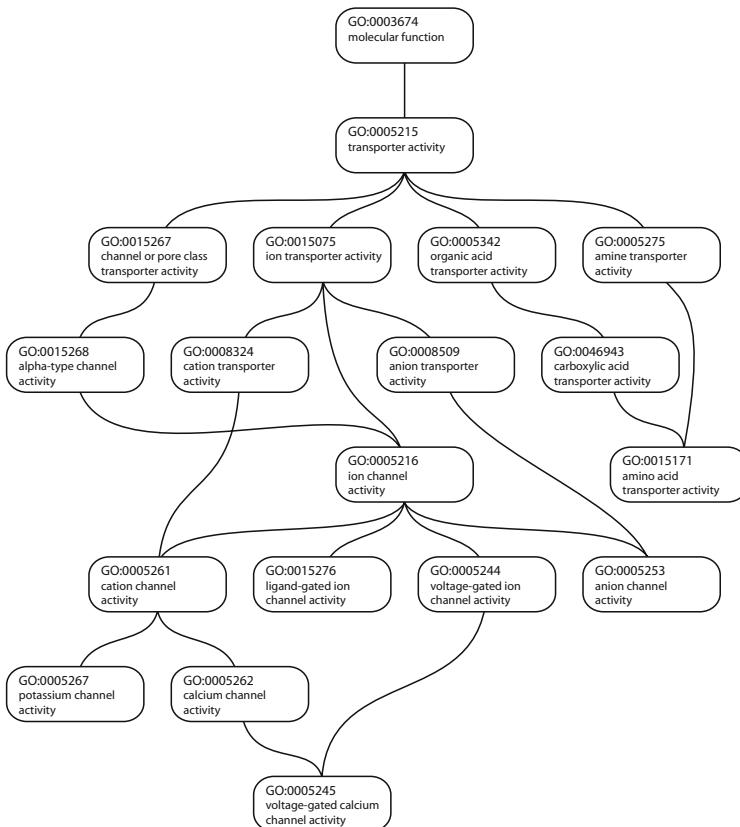


Fig. 3. Subset of the Gene Ontology (GO) ion channel hierarchy used in our experiments. GO terms GO:0003674 and G0:0005215 were not used since they represent the root of the hierarchy.

selected GO hierarchy. It turned out that many GO terms (classes) selected in the previous phase did not have a reasonable number of proteins associated with them. Therefore, we discarded GO terms with less than 10 protein records. At the end of this phase we had selected 147 protein examples and 17 GO terms.

In the third phase, for each protein retrieved in the previous phase we selected the amino acid sequence and MEDLINE document references from the UniProt database (release 12.0). This was accomplished using the database cross-reference to UniProt found in IntAct protein records.

After all three preparation phases, we ended up with a set of 147 protein examples, distributed in 17 GO terms (illustrated in Fig. 3). For each protein, we have retrieved the amino acid sequence, protein interaction data and MEDLINE documents. It should be noted that the number of protein

examples selected was constrained by the amount of protein interaction data available. This set of proteins was used to create seven different data sets, as detailed in Subsection 3.2.

3.2 Predictor Attributes

We have selected three different types of protein representations to be used as predictor attributes, namely amino acid composition, protein interaction data and textual information derived from MEDLINE document references.

Amino acid composition attributes were derived directly from proteins' primary sequences by calculating the ratio between the number of amino acid occurrences and the sequence length. For instance, if the amino acid A (Alanine) occurs 14 times in a protein sequence of length 140, the value of attribute A (attribute representing the composition of amino acid A) would be 0.10 (14 / 140). In other words, for each amino acid out of 20 that can be found in a protein sequence, we compute the percentage of the sequence composition relative to the amino acid in question. Using this procedure, 20 numeric attributes were produced for each protein in the data set.

Protein interaction attributes are useful because many proteins interact with one another to perform their function, by assembling multiprotein complexes or metabolic pathways, for example. If one can establish an interaction between a known-function protein and an unknown-function protein, the protein-protein interaction information can be used to predict the function of the unknown-function protein. Protein interaction data was encoded as binary attributes as follows. First, for each protein we retrieved the list of interactor proteins from IntAct database. Then, the complete list of interactors (interactors of all proteins in our data set) was filtered to remove interactors that only interact with one protein in the data set. This restriction was necessary to remove interactors present only in one protein, which do not have any predictive power. Finally, the filtered list of interactors (2095 interactors in total) was encoded as a binary attribute vector. Each position of the vector indicates, with a 'yes' or 'no' value, if the protein interacts with a particular interactor protein or not.

Textual information attributes derived from MEDLINE documents (titles and abstracts) in the form of keywords were encoded as binary attributes, using document references found in proteins' UniProt records. In total, 1010 documents were linked to the 147 proteins of our data set. We applied a Genetic Programming (GP) algorithm, as detailed in Subsection 3.4, to select relevant words (keywords) from linked documents in a preprocessing step. The selected keywords were encoded as a binary attribute vector. Each position of the vector indicates, with a 'yes' or 'no', if the documents linked to the protein contain a particular keyword or not.

Each of the above three types of attributes was used to produce a single data set, namely "AA" (for amino acid composition attributes), "PI" (for protein interaction attributes) and "TX" (for textual information attributes).

Furthermore, we explored the combination of predictor attributes in pairs, generating another three data sets (“AA-PI”, “AA-TX” and “PI-TX”), and one data set using all types of attributes (“AA-PI-TX”). In total we produced seven data sets, with the same number of examples (as explained in Subsection 3.1) but with a different attribute or combination of attributes in each data set. At the end, two simple predictor attributes derived directly from the proteins’ primary sequences, namely **sequence length** and **molecular weight**, were added to each of these seven data sets.

3.3 Classification Algorithm

In order to extract knowledge from the data sets described in the previous Subsection we used the J48 [28] classification algorithm. J48 is a Java implementation of the well known C4.5 decision tree algorithm [21]. We have chosen J48 mainly because C4.5 is a world-class standard induction algorithm and it produces a comprehensible classification model in a decision tree form.

A decision tree consists of internal (decision) nodes, which represent attribute tests, and leaf nodes, which represent classes of the problem in hand. An internal node has outgoing branches, where each branch represents a test outcome value, which in turn connect the nodes of the tree. A leaf node indicates a class to classify an example. Since all nodes in a decision tree have only one parent, with the exception of the root node which has no parent, there is a unique path from a leaf node to the root node. A path can be represented as a conjunction of attribute test outcomes (i.e. all internal nodes and branches followed by the path).

An example is classified by descending the decision tree in a top-down fashion – starting from the root node – following the branches according to the attribute tests’ outcomes until a leaf node is reached, where the class associated with the leaf node is assigned to the example. The path followed by the example can be analysed in order to explain the classification of an example into a particular class.

As we are dealing with a class hierarchy represented by a GO subset (illustrated in Fig. 3), and J48 is a flat classifier (i.e., it cannot directly cope with hierarchical classes), we have transformed the hierarchical problem into a set of flat classification problems.¹ The transformation procedure works as follows. For each GO term, we have split the data set into positive (which belong to the GO term) and negative (which do not belong to the GO term) examples. An example (protein) belongs to a specific GO term if it is annotated with that GO term or is annotated with one or more of its child GO terms. For instance, an example annotated with the GO term GO:0015171 will be considered as a positive example for GO terms GO:0015171, GO:0046943, GO:0005275 and GO:0005342 (where the latter three terms are ancestors of the former), according to the class hierarchy. This transformation procedure

¹ For a more complete discussion about the differences between flat and hierarchical classification refer to [8].

is required due to the semantics of GO annotation (GO true path rule), where a protein is explicitly annotated only with its most specific GO terms, but it is implicitly considered to have all the ancestral terms of those specific terms [2]. After the transformation step, we trained a classifier for each GO term. That is, each classifier performs a binary classification, predicting whether or not a given example belongs to the classifier's associated GO term.

3.4 Attribute Selection

In order to reduce the number of words to be used as textual information attributes, we performed attribute selection using standard Genetic Programming. Genetic Programming (GP) [14, 4] is an evolutionary technique, based on Darwin's principle of natural selection, which aims at automatically evolving computer programs. In the GP context, a computer program is a solution to the problem at hand, which can be represented as a mathematical equation, a sequence of instructions or an arbitrary combination of input values. GP uses the principle of natural selection to find solutions to complex problems by evolving initially poor solutions into near-optimal ones using a set of genetic operators and a fitness (quality) measure.

Essentially, a GP algorithm consists of a population of candidate solutions to the target problem and an iterative selection process that mimics an evolutionary process. Candidate solutions are selected based on a fitness measure, which measures the quality of candidate solution, to undergo reproduction, recombination and mutation operators in order to form a new population. The new population replaces the old one and a new iteration begins. The fitness-based selection determines that better candidate solutions are more often selected on average, while poor candidate solutions have a smaller chance of being selected. New candidate solutions are generated by applying recombination and mutation operators, which are responsible for performing a global search in the solution space. The iterative selection process is carried out until an arbitrary number of iterations (generations of the evolutionary process) is reached or an optimal or satisfactory solution is found.

The attribute selection process involves elimination of stop words, word stemming and GP-based selection of predictive words (keywords). It was divided in three steps as follows. In the first step, all MEDLINE documents (titles and abstracts) linked to proteins in our data set were pre-processed by applying the Bow library [16] to carry out stop word removal, followed by stemming using Porter's algorithm [20]. The objective of this step is to remove irrelevant words and to group inflected or derived words to their stem (root form). Examples of stop words are 'a' and 'the', which carry no meaning for text mining purposes. An example of stemming would be to replace the words 'learning' and 'learned' by their stem 'learn'. At the end, a list of words W was generated using all the resultant stems. The second step consists of transforming each document to a vector of word frequencies,

Table 2. Parameters of the GP algorithm used for attribute selection

Name	Value
number of generations	50
number of individuals per generation	60000
maximum tree depth	17
mutation rate	5%
crossover rate	90%
reproduction rate (with elitism)	10%

using the word list generated in the previous step. For each document d , a vector v_d of length equal to the number of words in W was created where the value of each position i is given by

$$v_d(i) \equiv \frac{\text{number of occurrences of } w_i \text{ in } d}{\text{number of words in } d}, \quad (1)$$

where w_i is the i -th word of W .

In the third step, a GP algorithm was used to identify a list of relevant words for a particular GO term. The selection of words works as follows. Firstly, we transformed the hierarchical classification problem into a set of flat classification problems, using the same procedure described in Subsection 3.3. Then, for each GO term we trained a standard GP to classify the document vectors generated in the previous step. The function set consists of basic arithmetic operators (addition, subtraction and multiplication) and a “max” (maximum value between two numbers) function. The terminal set consists of ephemeral random constants [14] and input nodes representing each dimension of a document vector. The fitness function used is the area under a ROC curve (AUC) [6]. Table 2 presents the parameters of the GP algorithm. At the end, a list of relevant words for each GO term was selected by analysing the five best GP individuals. Words that appear at least in two out of five best individuals were selected as keywords.

4 Computational Results

All experiments were conducted running the well-known 10-fold cross-validation procedure [28]. In essence, a cross-validation procedure consists of splitting the data set into n ($n = 10$, in our case) partitions of approximately same size (number of examples). In an iterative process, each i th ($i = 1, \dots, n$) partition was used as the test set and the remaining 9 partitions were temporarily merged and used as a training set. In each iteration, a classification algorithm generates a classification model using the training data. Then, the

Table 3. Predictive accuracy (*average* \pm *standard deviation*) obtained with each type of attribute per GO term. An entry in the column “TX” is shown **in bold** if, for the corresponding GO term, the accuracy achieved with textual information attributes is significantly greater than the accuracy achieved with the second best type of attribute, columns “AA” (amino acid composition) or “PI” (protein interaction), for the GO term in question – according to a two-tailed Student’s t-test with significance level $\alpha = 1\%$.

Term	AA	PI	TX
GO:0015267	69.26 \pm 2.01	55.30 \pm 3.77	94.38 \pm 1.78
GO:0015075	71.49 \pm 4.37	59.30 \pm 1.14	88.49 \pm 2.81
GO:0005342	84.32 \pm 3.01	85.75 \pm 2.08	87.18 \pm 1.49
GO:0005275	88.57 \pm 2.80	87.93 \pm 1.53	92.65 \pm 1.52
GO:0015268	65.92 \pm 3.29	57.02 \pm 2.64	93.24 \pm 2.98
GO:0008324	64.55 \pm 3.76	59.92 \pm 1.63	83.77 \pm 2.24
GO:0008509	92.01 \pm 1.86	93.21 \pm 1.42	93.44 \pm 1.87
GO:0046943	86.59 \pm 2.38	89.89 \pm 1.78	88.55 \pm 1.35
GO:0005216	73.32 \pm 3.46	57.71 \pm 3.60	93.18 \pm 1.77
GO:0015171	88.59 \pm 1.62	89.26 \pm 1.74	88.51 \pm 2.27
GO:0005261	71.92 \pm 2.17	68.68 \pm 2.66	88.93 \pm 3.44
GO:0015276	86.37 \pm 2.00	89.18 \pm 1.03	85.18 \pm 2.34
GO:0005244	69.42 \pm 1.95	81.05 \pm 3.02	83.58 \pm 2.58
GO:0005253	94.03 \pm 2.01	94.65 \pm 1.85	97.99 \pm 1.02
GO:0005267	82.38 \pm 1.75	88.52 \pm 0.95	97.29 \pm 1.11
GO:0005262	86.90 \pm 2.64	85.64 \pm 1.67	87.42 \pm 3.15
GO:0005245	89.90 \pm 2.84	93.24 \pm 2.22	90.57 \pm 2.84

classification model is used to classify unseen examples from the test set, in order to evaluate the discovered knowledge. The predictive accuracy rate is then computed as the average accuracy rate over the 10 test sets.

The results concerning the predictive accuracy obtained with each type of attribute per GO term are shown in Table 3. An entry in the column “TX” is shown **in bold** if, for the corresponding GO term, the accuracy achieved with textual information attributes is significantly greater than the accuracy achieved with the second best type of attribute (columns “AA” or “PI”) for the GO term in question – according to a two-tailed Student’s t-test with significance level $\alpha = 1\%$. Overall, the highest predictive accuracy was achieved when using textual information as predictor attributes (data set “TX”). The highest accuracy of “TX” attributes was statistically significant in 7 out of 17 cases. There was no statistically significant difference between the accuracy of “TX” and the accuracy of the other two types of attributes in the remaining

Table 4. Predictive accuracy (*average ± standard deviation*) per GO term for data sets using a combination of different types of predictor attributes. There were no significant differences between the highest predictive accuracy achieved with one combination of attributes and the second best combination of attributes.

Term	AA-PI	AA-TX	PI-TX	AA-PI-TX
GO:0015267	67.38 ± 3.82	94.38 ± 1.78	94.38 ± 1.78	92.95 ± 2.13
GO:0015075	64.17 ± 3.36	87.15 ± 2.30	87.15 ± 2.53	87.87 ± 2.59
GO:0005342	84.32 ± 3.01	87.86 ± 4.23	87.18 ± 1.49	87.86 ± 4.23
GO:0005275	88.57 ± 2.80	93.28 ± 1.99	92.65 ± 1.52	93.28 ± 1.99
GO:0015268	59.20 ± 3.66	88.47 ± 3.19	93.24 ± 2.98	89.90 ± 3.01
GO:0008324	72.54 ± 3.11	85.78 ± 2.08	85.20 ± 2.05	84.49 ± 1.92
GO:0008509	92.01 ± 1.86	94.07 ± 1.48	93.44 ± 1.87	94.07 ± 1.48
GO:0046943	86.59 ± 2.38	86.50 ± 2.16	88.55 ± 1.35	86.50 ± 2.16
GO:0005216	63.12 ± 4.52	91.76 ± 1.73	93.18 ± 1.77	91.76 ± 1.73
GO:0015171	88.59 ± 1.62	89.93 ± 1.76	88.51 ± 2.27	89.93 ± 1.76
GO:0005261	67.71 ± 5.18	86.88 ± 3.56	86.31 ± 2.72	84.88 ± 3.58
GO:0015276	89.18 ± 1.03	86.47 ± 1.68	89.18 ± 1.43	88.52 ± 1.37
GO:0005244	69.42 ± 1.95	84.44 ± 1.33	83.58 ± 2.58	84.44 ± 1.33
GO:0005253	94.03 ± 2.01	95.99 ± 1.79	97.99 ± 1.02	95.99 ± 1.79
GO:0005267	82.38 ± 1.75	96.62 ± 1.13	97.29 ± 1.11	96.62 ± 1.13
GO:0005262	88.10 ± 2.82	86.12 ± 2.86	90.14 ± 3.06	87.55 ± 2.14
GO:0005245	89.90 ± 2.84	89.24 ± 3.00	90.57 ± 2.84	89.24 ± 3.00

10 cases. These results indicate that textual information attributes are useful for predicting GO terms at any level in the hierarchy used in our experiments. Protein interaction attributes achieved the lowest predictive accuracy in 5 out of 17 cases – in GO terms GO:0015267, GO:0015075, GO:0015268, GO:0008324 and GO:0005216. At the same time, they achieved competitive predictive accuracy (when compared to textual information attributes) in 3 cases – GO terms GO:0008509, GO:0005253 and GO:0005245. These results suggest that protein interaction attributes are useful for predicting GO terms at levels near the leaves of the hierarchy, since at levels near the root of the hierarchy they achieved a significantly lower accuracy. Amino acid composition attributes achieved an average predictive accuracy, overall. In 2 cases – GO terms GO:0008509 and GO:0005253 – the achieved accuracy was competitive with textual information attributes.

The results concerning predictive accuracy per GO term for data sets using a combination of different types of predictor attributes are shown in Table 4. There were no statistically significant differences between the highest predictive accuracy achieved with one combination of attributes and the second

Table 5. Summary of the results obtained in our experiments. Each cell represents the number of times in which the attribute type in the corresponding row obtained an accuracy statistically significantly greater (positive value) or worse (negative value) than the attribute type in the corresponding column. The value of a cell is in the range $[-17, +17]$, since we are dealing with 17 different GO terms.

	AA	PI	TX	AA-PI	AA-TX	PI-TX	AA-PI-TX
AA	–	0	-8	0	-8	-8	-8
PI	0	–	-6	0	-6	-6	-6
TX	8	6	–	8	0	0	0
AA-PI	0	0	-8	–	-8	-8	-7
AA-TX	8	6	0	8	–	0	0
PI-TX	8	6	0	8	0	–	0
AA-PI-TX	8	6	0	7	0	0	–

best combination of attributes. Also, the combination of all three types of predictor attributes did not lead to a significant increase of the predictive accuracy, when compared to the best results using data sets with a single type of attribute. All the combination of attributes that contain textual information attributes achieved competitive predictive accuracy when compared to the predictive accuracy achieved using only textual information attributes (“TX” data set).

Table 5 presents a summary of the results obtained in our experiments. Each cell in that table represent the number of times in which the attribute type in the corresponding row obtained an accuracy significantly greater (positive value) or worse (negative value) than the attribute type in the corresponding column. The value of each cell is in the range $[-17, +17]$, since we are dealing with 17 different GO terms. In 9 out of 17 cases – GO terms GO:0005342, GO:0005275, GO:0008509, GO:0046943, GO:0015171, GO:0015276, GO:0005253, GO:0005262 and GO:0005245 – all single attribute types and combination of types of attributes achieved competitive accuracy, with no significant differences between them. The best results were obtained when using textual information attributes (as a single attribute type or in combination with different types of attributes). Experiments using only protein interaction attributes were slightly better than experiments using only amino acid composition, but both these types of attributes led to poor results, when compared to experiments using textual information attributes.

5 Conclusions and Future Research

This chapter has presented an empirical evaluation comparing the effectiveness of different protein representations in terms of maximizing predictive

accuracy using the Gene Ontology (GO) hierarchical functional classification scheme. Hierarchical structures present a challenging problem, since it is generally more difficult to discriminate between specific classes represented by leaf nodes than more general classes represented by internal nodes.

The set of experiments consisted of using different types and combinations of types of predictor attributes for protein function prediction, comparing the predictive accuracy obtained by J48 across a subset of the Gene Ontology ion channel hierarchy. Since J48 is a flat classifier, and we are dealing with a hierarchical classification problem, we have transformed the hierarchical problem into a set of flat classification problems. The results have shown that some types of predictor attributes are more suitable for different levels of the hierarchy. While protein interaction attributes achieved the lowest accuracy at top levels of the hierarchy, they are competitive with the highest accuracy achieved by textual information attributes at lower levels of the hierarchy. Overall, the highest predictive accuracy was achieved when using textual information as predictor attributes, suggesting the importance of using textual information attributes derived from the biological literature for protein function prediction.

As future research, it would be interesting to evaluate different types of predictor attributes, such as gene expression data and post-translational modification data. Also, given the hierarchical nature of predicting function using the GO classification scheme, investigating different measures of predictive accuracy tailored for hierarchical problems may give valuable insights about different protein representations.

Acknowledgements

The authors acknowledge the financial support from an European Union's INTERREG project (Ref. No. 162/025/361). Fernando Otero also acknowledges further financial support from the Computing Laboratory, University of Kent.

References

1. Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., Walter, P.: *The Molecular Biology of the Cell*, 4th edn. Garland Press (2002)
2. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology. *Nature Genetics* 25, 25–29 (2000)
3. Attwood, T.K., Mitchell, A., Gaulton, A., Moulton, G., Tabernero, L.: The prints protein fingerprint database: functional and evolutionary applications. In: Dunn, M., Jorde, L., Little, P., Subramaniam, A. (eds.) *Encyclopaedia of Genetics, Genomics, Proteomics and Bioinformatics*. John Wiley & Sons, Chichester (2006)

4. Banzhaf, W., Nordin, P., Keller, R.E., Francone, F.D.: Genetic Programming—an introduction: on the automatic evolution of computer programs and its applications. Morgan Kaufmann, San Francisco (1998)
5. Bock, J.R., Gough, D.A.: In Silico Biological Function Attribution: a different perspective. BioSilico 2(1), 30–37 (2004)
6. Fawcett, T.: An introduction to ROC analysis. Pattern Recognition Letters 27, 861–874 (2006)
7. Finn, R.D., Tate, J., Mistry, J., Coggill, P.C., Sammut, S.J., Hotz, H.R., Ceric, G., Forslund, K., Eddy, S.R., Sonnhammer, E.L.L., Ateman, A.: The pfam protein families database. Nucleic Acids Research 36, D281–D288 (2008)
8. Freitas, A.A., de Carvalho, A.C.P.L.F.: A tutorial on hierarchical classification with applications in bioinformatics. In: Taniar, D. (ed.) Research and Trends in Data Mining Technologies and Applications. Idea Group (2007)
9. Friedberg, I.: Automated protein function prediction – the genomic challenge. Briefings in Bioinformatics 7(3), 225–242 (2006)
10. Gerlt, J.A., Babbitt, P.C.: Can sequence determine function? Genome Biology 1(5), 1–10 (2000)
11. Higgs, P.G., Attwood, T.: Bioinformatics and Molecular Evolution. Blackwell Publishing, Malden (2005)
12. Hulo, N., Bairoch, A., Bulliard, V., Cerutti, L., De Castro, E., Langendijk-Genevaux, P., Pagniand, M., Sigrist, C.: The prosite database. Nucleic Acid Research 34, D227–D230 (2006)
13. Jensen, L.J., Gupta, R., Stærfeldt, H.H., Brunak, S.: Prediction of human protein function according to gene ontology categories. Bioinformatics 19(5), 635–642 (2003)
14. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
15. Letovsky, S., Kasif, S.: Predicting protein function from protein/protein interaction data: a probabilistic approach. Bioinformatics 19(1), i97–i204 (2003)
16. McCallum, A.K.: Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering (1996), <http://www.cs.cmu.edu/~mccallum/bow>
17. Mulder, N.J., Apweiler, R., Attwood, T.K., Bairoch, A., Bateman, A., Binns, D., Bork, P., Buillard, V., Cerutti, L., Copley, R., Courcelle, E., Das, U., Daugherty, L., Dibley, M., Finn, R., Fleischmann, W., Gough, J., Haft, D., Hulo, N., Hunter, S., Kahn, D., Kanapin, A., Kejariwal, A., Labarga, A., Langendijk-Genevaux, P.S., Lonsdale, D., Lopez, R., Letunic, I., Madera, M., Maslen, J., McAnulla, C., McDowall, J., Mistry, J., Mitchell, A., Nikolskaya, A.N., Orchard, S., Orengo, C., Petryszak, R., Selengut, J.D., Sigrist, C.J.A., Thomas, P.D., Valentin, F., Wilson, D., Wu, C.H., Yeats, C.: New developments in the InterPro database. Nucleic Acid Research 35, D224–D228 (2007)
18. Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C.: Scop: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol. 247, 536–540 (1995)
19. Pappa, G.L., Baines, A.J., Freitas, A.A.: Predicting post-synaptic activity in proteins with data mining. Bioinformatics 21(2), ii19–ii25 (2005)
20. Porter, M.F.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
21. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)

22. Raychaudhuri, S.: Computational Text Analysis for Functional Genomics and Bioinformatics. Oxford University Press, Oxford (2006)
23. Rost, B., Liu, J., Nair, R., Wrzeszczynski, K.O., Ofran, Y.: Automatic prediction of protein function. *Cellular and Molecular Life Sciences (CMLS)* 60(12), 2637–2650 (2003)
24. Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., Tetko, I., Guldener, U., Mannhaupt, G., Munsterkotter, M., Mewes, H.W.: The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acid Research* 32(18), 5539–5545 (2004)
25. Tian, W., Skolnick, J.: How well is enzyme function conserved as a function of pairwise sequence identity? *Journal of Molecular Biology* 333(4), 863–882 (2003)
26. Webb, E.: Enzyme Nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology. Academic Press, London (1992)
27. Whisstock, J.C., Lesk, A.M.: Prediction of protein function from protein sequence and structure. *Q Rev. Biophys.* 36(3), 307–340 (2003)
28. Witten, H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
29. Xie, H., Wasserman, A., Levine, Z., Novik, A., Grebinskiy, V., Shoshan, A., Mintz, L.: Large-scale protein annotation through gene ontology. *Genome Research* 12, 785–794 (2002)

Genetic Selection Algorithm and Cloning for Data Mining with GMDH Method

Marcel Jirina¹ and Marcel Jirina Jr.²

¹ Institute of Computer Science, Pod vodarenskou vezi 2,
182 07 Prague 8 – Liben, Czech Republic
marcel@cs.cas.cz

² Faculty of Biomedical Engineering, Czech Technical University in Prague,
Nam. Sitna 3105, 272 01, Kladno, Czech Republic
jirina@fbmi.cvut.cz

Summary. The Group Method Data Handling Multilayer Iterative Algorithm (GMDH MIA) is modified by use of the selection procedure from genetic algorithms while including cloning of the best neurons generated to obtain even less error. The selection procedure finds parents for a new neuron among already existing neurons according to the fitness and also with some probability from the network inputs. The essence of cloning is slight modifying the parameters of the copies of the best neuron, i.e. the neuron with the largest fitness. The genetically modified GMDH network with cloning (GMC GMDH) can outperform other powerful methods. It is demonstrated on some tasks from the Machine Learning Repository.

1 Introduction

Classification of multivariate data into two or more classes is an important problem of data processing in many fields of data mining. For classification of multivariate data into classes (groups, categories etc.) the well-known GMDH MIA (group method data handling multilayer iterative algorithm) is often used. This approach – in difference to others – can even provide a closed form polynomial solution when needed.

Although GMDH MIA is one of relatively standard methods, there are new findings in the application of genetic optimization for enhancing GMDH behavior and there is the idea of using the cloning principle borrowed from immune networks to further optimize GMDH based algorithms.

The standard GMDH MIA method has been described in many papers since 1971 e.g. in [1], [4], [5], [6], [7], [10]. The basis of the GMDH MIA is that each neuron in the network receives input from exactly two other neurons from previous layer; the first layer is formed using neurons representing the input layer.

The two inputs, x and y are then combined to produce a partial descriptor based on the simple quadratic transfer function (the output signal is z):

$$z = a_1 + a_2x + a_3y + a_4x^2 + a_5y^2 + a_6xy \quad (1)$$

where coefficients a, \dots, f are determined by linear regression and are unique for each neuron. The coefficients can be thought of as analogous to weights found in other types of neural networks.

The network of transfer functions is constructed one layer at a time. The first network layer consists of functions of each possible pair of n input variables (zero-th layer) resulting in $n(n+1)/2$ neurons. The second layer is created using inputs from the first layer and so on. Due to exponential growth of the number of neurons in a layer, a limited number of best neurons are selected and the other neurons are removed from the network after finishing the layer. For illustration see Fig. 1. In the standard GMDH MIA algorithm all possible pairs of neurons from the preceding layer (or inputs when the first layer is formed) are taken as pairs of parents. The selection consists of selecting a limited number of the best descendants, "children", while the others are removed after they arose and were evaluated. In this way, all variants of GMDH MIA are rather ineffective as there are a lot of neurons generated, evaluated and then immediately removed with no other use.

In this Chapter we solve the classification task by use of GMDH MIA, modified by a selection algorithm common in genetic algorithms and by cloning the best neurons generated up to a given moment of the learning process. In our approach, when generating a network of neurons, for each new neuron, its parents are selected from all presently existing neurons and network inputs also. Then six parameters of the new neuron are computed using the least mean squared error method. No explicit layered structure arises because of the random selection of the parents.

The number of neurons grows during learning of one neuron at a time. No neuron is deleted during the learning process and in the selection procedure its chance to become a parent for a neuron is proportional to its fitness. If a new neuron appears to be the best, its clones are generated. Clones are inexact copies of the parent neuron, which was found to be the best neuron generated up to now. The true minimum is searched in the neighborhood of the minimum given by the parameters found by standard linear regression. Therefore, the clones have mutated, i.e. have slightly changed parameters of the parent individual to cover that neighborhood. In this respect we use the idea that a clone is an inexact copy of the parent GMDH neuron.

Note that GMDH is principally a function approximator. When used as a two class classifier, the network approximates the class mark, often 1 and 0 or 1 and -1. To decide to which class an applied sample belongs to, the network output is compared with the threshold θ . If the output is equal to or larger than this threshold, the sample belongs to one class or else to the

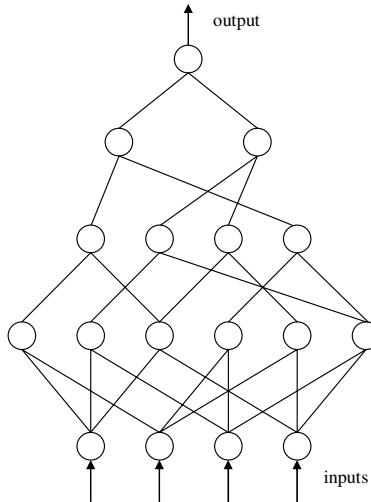


Fig. 1. An example of GMDH MIA structure

other class. The value of the threshold can be optimized to get a minimal classification error. We use this approach here too.

It is shown here that a new algorithm, including cloning, allows tuning the GMDH neural network more effectively than is possible in genetically optimized GMDH networks.

We tested the genetically modified GMDH network with cloning (GMC GMDH) on some tasks from the Machine Learning Repository and compared it with other widely accepted methods. It is demonstrated that GMC GMDH can outperform other powerful methods.

2 Related Works

There are several works dealing with genetic optimization of the GMDH MIA method [3], [7], [8], and [9]. These approaches use the GMDH networks of limited size as the number of neurons in a layer and the number of layers. Some kind of randomization must be used to get a population of GMDH networks because the original GMDH MIA neural network is a purely deterministic. Individuals in a population are subjects of genetic operations of selection, crossover and mutation. As each GMDH network represents a rather general graph, there must be procedures for crossover of graphs similarly as in other kinds of genetically optimized neural networks, e.g., NNSU [10]. For example, in [18] the NN architecture is built by adding hidden layers into the network, while configuration of each neuron's connections is defined by means of GA. An elitist GA with binary encoding, fitness proportional selection, standard operators of crossover and mutation are used in the algorithm. Different genetically optimized GMDH networks in literature differ

in the way how a population, especially the first population of networks, is formed and by variants of the genetic procedures used. It seems, however, that up to now no approach in genetically optimized GMDH networks essentially brings better results than the standard GMDH MIA algorithm. On the other hand, genetically optimized GMDH networks eliminate the necessity to set up, in advance, the number of the best neurons left in each layer at least. In this way, such GMDH networks become even less “parameter-less” than before.

The difficult problem with genetically optimized GMDH method lies in the fact, that a population of GMDH networks must be generated. There must be crossover of individuals. Individuals are networks, which generally have different graphs. The crossover of two different graphs is a rather complex task. Its solution is known and also used in other genetically optimized neural networks [10], [15], [16], and [17].

A very interesting and principally simple application of the selection process to GMDH MIA was published by Hiasaat and Mort in 2004 [8]. Their method does not remove any neuron during learning. Thus it allows unfit individuals from early layers to be incorporated at an advanced layer where they generate fitter solutions. Secondly, it also allows those unfit individuals to survive the selection process if their combinations with one or more of the other individuals produce new fit individuals, and thirdly, it allows more implicit non-linearity by allowing multi-layer variable interaction. The GMDH algorithm is constructed in the same manner as the standard GMDH algorithm except for the selection process. In order to select the individuals that are allowed to pass into the next layer, all the outputs of the GMDH algorithm at the current layer are entered as inputs in the genetic algorithm. It was shown in [8] that this approach can outperform the standard GMDH MIA when used in the prediction of two daily currency exchange rates. No other test of this approach classification ability was performed in the literature cited. The GMDH network [8] has a layered structure where the input to the neuron can be the output of any already existing layer or even network input. To keep the layered structure in this context seems rather complicated. One can use generalization where a new neuron input can be the output of any already existing neuron or even network input. Thus, the strict layered structure disappears.

Immunity-based models including cloning are new optimization techniques. Note first that the authors dealing with artificial immune systems, e.g. [11], [12] use different terminology than used in the neural network community and the genetic algorithm community. So, some translation or mapping is needed. Here especially, antibody – neuron, affinity – fitness, antigen or antigenic stimulus – signal. Leukocytes or white blood cells are divided into three classes, one of which is lymphocyte. Lymphocytes include B cells, which mature in bone marrow, and T cells, which mature in the thymus. White blood cells produce antibodies – proteins that each respond to a specific antigen. Antigens are enemy agents, i.e. viruses or bacteria dangerous

for an organism. There, various mechanisms or processes in the immune system appear, which are investigated in the development of artificial immune systems (AIS) [11].

1. Negative selection is the process that happens during the development of T cells in the thymus. Immature T cells go through a censoring process so those that recognize self-cells are eliminated. Only the rest are deployed into the immune system so T cells will not attack the self-cells. Artificial negative selection algorithms were proposed to mimic that procedure: generating detector candidates randomly, and then eliminating those that match the self-samples. It suits the need of the so-called “one-class classification” problem, e.g. anomaly detection, in which training data from only one of the two classes are available.
2. The immune network model is another widely used model in the AIS field. It was based on the idiotypic¹ network concepts proposed by Jerne in 1974, in which the dynamics of the immune system can be described as a network of antibodies where activation by antigens, and activation and suppression between antibodies co-exist.
3. The relatively new models in AIS include the danger theory, which emphasizes discriminating the danger posed by the invaders instead of whether it is self or nonself.
4. Besides the immune network, the other theoretical framework of immunology, clonal selection theory, which is in fact more widely accepted, is also often used in AIS. The clonal selection principle describes the basic features of an immune response to antigenic stimulus. It establishes the idea that only those cells that recognize the antigen proliferate, thus being selected against those that do not. The main features of the clonal selection theory are that:
 - The new cells are copies of their parents² (clone) subjected to a mutation mechanism with high rates (somatic hypermutation);
 - Elimination of newly differentiated lymphocytes carrying self-reactive receptors;
 - Proliferation and differentiation on contact of mature cells with antigens.

From these ideas we use results of clonal selection theory, especially a cloning procedure derived from the Simple Clonalg algorithm [12]:

BEGIN

Construct the initial population of antibodies

REPEAT

Evaluate antibodies to calculate their affinities

¹ idioype - individual genotype.

² In fact, of the original - parent cell, as there are no parents in the standard sense of word.

```

Select the  $n$  best affinity antibodies and clone them
Maturate the clones and evaluate them
Allow the best antibody from each subpopulation to survive
Replace the  $d$  lowest affinity antibodies with new
antibodies randomly produced
UNTIL a terminal criterion is satisfied or the maximum
generation number of clones is reached
END

```

In biological systems clones are not exact copies of the parent cell because some mutations are in effect. In artificial systems, clones are also not exact copies of the parent cell or neuron, and therefore some mutation must be in effect. The clone, to be a true clone, must have the same parents, i.e. input signals. So, the basic parameters – the two parents are not changed. The problem is, how to change the parameters a, \dots, f of the parent neuron. These changes should be small enough to keep a sufficient similarity of the clone to the original individual, and, at the same time, sufficiently large enough to reach the necessary changes for searching the data space in the neighborhood of the parent neuron.

The application of the cloning process to GMDH MIA was not proposed up to now. Usually – and we do it as well – the parameters of the new neuron are set up by linear regression, i.e. with a least mean squared error method. This method uses the following Gauss-Markov assumptions [19]:

The random errors have an expected value of 0.

The random errors are uncorrelated.

The random errors are homoscedastic, i.e., they all have the same variance.

The errors are not assumed to be normally distributed, nor are they assumed to be independent, nor are they assumed to be identically distributed.

We expect that due to the nonlinearity of the problem as well as the GMDH network, the assumption of homoscedasticity is not met especially for the classification problem. In classification problems each class has a different origin from which a different distribution of regression errors may arise. Even such a weak, but essential, assumption as homoscedasticity is not met. We show it in a detailed analysis in the following paragraph Linear regression. Thus, the true minimum may lie somewhere in the neighborhood of the minimum given by the parameters found by standard linear regression.

3 Linear Regression

In this chapter we show, on an example, that the Gauss-Markov assumptions of an expected value 0 and of homoscedasticity need not be fulfilled in GMDH network. The solution obtained by the least mean squared error method then need not be the optimal solution. Thus, the truly optimal solution may lie somewhere in the neighborhood of the solution found by the LMS method.

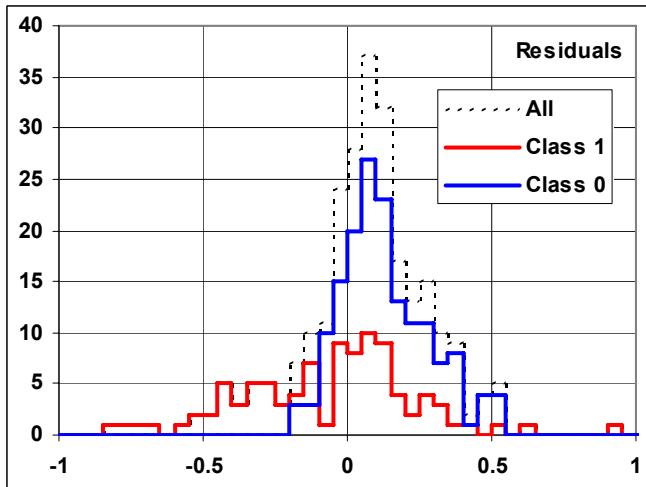


Fig. 2. Histograms of residuals for both classes and both classes together

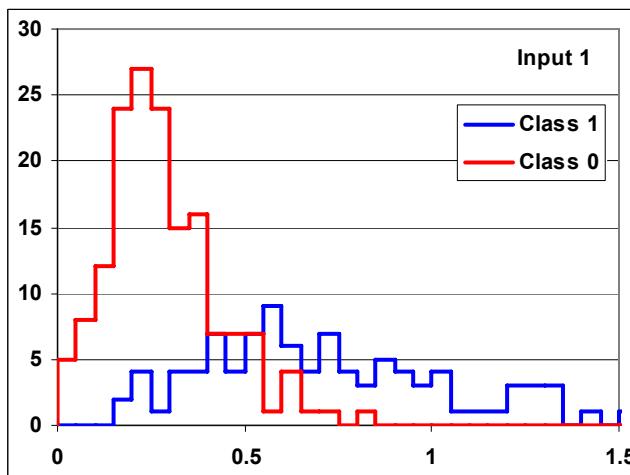


Fig. 3. Histograms of input signals for both classes for input 1 of the neuron

To prove the assertion that homoscedasticity need not be fulfilled in GMDH network, a simple example will suffice. We used the standard GMDH MIA method for the Brest CW data from the UCI Machine learning repository [13]. The Brest CW problem is a two-class classification task. We analyzed the input and output values of the neurons and found that zero mean and homoscedasticity of residuals is often not fulfilled. It is shown on the example of one neuron.

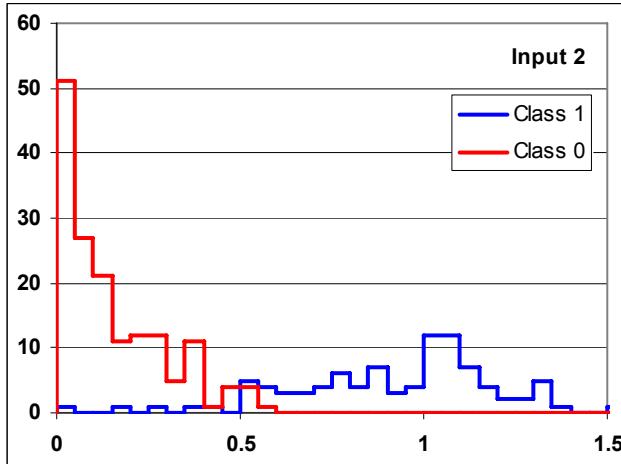


Fig. 4. Histograms of input signals for both classes for input 2 of the neuron

In Fig. 2, histograms of residuals, i.e. histograms of errors for one neuron and for both classes separately and for all data are depicted. There, it is seen that first, the expected value apparently is not zero, and second, residuals are heteroscedastic. The heteroscedasticity originates in the fact that there is data of two classes, each with different statistics. It is shown in Figs. 3 and 4, where histograms of input signals to the neuron analyzed are depicted.

It is seen that classes have very different statistics. After polynomial transformation according to (1) the difference increases because the target of the transformation is to get the output as close as possible to the value 0 for class zero and to the value 1 for class one. From this difference, the difference in statistics of residuals also follows - see Fig. 2 - after transformation (1) where coefficients were set up by linear regression.

Different approaches can be used to find a better solution than the solution obtained by linear regression. The solution obtained by linear regression can be used as the first approximation. We use cloning, i.e. we generate neurons with the same inputs and with parameters a, \dots, f slightly modified with respect to their original values.

4 Genetically Modified GMDH

Here we describe the approaches which result in our construction of a genetically modified GMDH network with cloning.

4.1 The Learning Set

The learning set consists of $n + 1$ dimensional vectors $(x_i, y_i) = (x_{1i}, x_{2i}, \dots, x_{ni}, y_i)$, $i = 1, 2, \dots, N$ where N is the number of learning samples (patterns or examples). The learning set can be written in the matrix form

$$[X, Y].$$

The matrix X has n columns and N rows; Y is a column vector of N elements. In the GMDH, the learning set is usually broken into two disjoint subsets, the training set (or construction or setup set) and the so-called validation set. In the learning process the former one is used for setting up parameters of neurons of the newly created neuron, the latter for evaluation of an error of the newly created neuron. Thus $N = N_s + N_v$, where N_s is the number of rows used for setting up the parameters of neurons (the training set), and N_v is the number of rows used for error evaluation during learning (the validation set).

4.2 New Genetically Modified GMDH Network Algorithm

The standard quadratic neuron is an individual of the genetically modified GMDH network. Its parents are two neurons (or possibly one or two network inputs) from which two input signals are taken. A selection of one neuron or input as one parent and of another neuron or input as the other parent can be made by the use of different criteria. In genetic algorithms, in the selection step, there is a common approach that the probability to be a parent is proportional to the value of the fitness function. Just this approach is used here. The fitness is simply a reciprocal of the mean absolute error on the validation set.

An operation of a crossover in the genetically modified GMDH is, in fact, no crossover in the sense combining two parts of parents' genomes. In our approach, Equation (1) gives us a symmetrical procedure of mixing the parents' influence but not their features, parameters. The parameters a, \dots, f , see (1), are stated separately.

4.3 Selection Procedure

The initial state form n inputs only, there are no neurons. If there are already k neurons, the probability of a selection from inputs and from neurons is given by

$$p_i = n/(n + k),$$

$$p_n = k/(n + k)$$

for $n/(n + k) > p_0$, where p_0 is the minimal probability that one of the network inputs will be selected; we found $p_0 = 0.1$ as optimal. Otherwise,

$$p_i = p_0,$$

$$p_n = (1 - p_0).$$

The fitness function is equal to the reciprocal error on the verification set. Let $\varepsilon(j)$ be the mean error of the j -th neuron on the validating set. The probability that neuron j will be selected is:

$$p_n(j) = (1 - p_n) \frac{1/\varepsilon(j)}{\sum_{s=1}^{N_T} 1/\varepsilon(s)}.$$

Moreover, it must be assured that the same neuron or the same input is not selected as the second parent of the new neuron.

After the new neuron is formed and evaluated it can immediately become a parent for another neuron. Thus, the network has no explicit layers. Each new neuron can be connected to any input or up to any existing neurons at that point.

The computation of six parameters a, \dots, f , see (1), of the new neuron is the same as in the GMDH MIA algorithm.

4.4 Best Neuron

The new neuron added need not be better than all others. Therefore, the index and error value of the best neuron is stored as long as a better neuron arises. Thus every time there is information about the best neuron, i.e. the best network's output without need of sorting. After learning, this output is used as a network output in the recall phase.

4.5 Cloning Mechanism

There are lots of ideas how to do the cloning. From these ideas, we use cloning in the form close to the SIMPLE CLONALG algorithm [12] in this way:

```

BEGIN
  Given the Best GMDH Neuron with parents (i.e. input signals
  from)  $In_1, In_2$  and with six parameters  $a, b, \dots, f$ .
REPEAT
  Produce a copy of the Best Neuron. A copy has the same
  inputs  $In_1$  and  $In_2$  but mutated parameters  $a, \dots, f$ , i.e.
  parameters slightly changed (details see below)
  Evaluate fitness of this clone neuron.
  If this neuron happens to be better than the Best
  Neuron, break this clone generating cycle (and start
  this cloning algorithm from the beginning with
  new Best Neuron again).
UNTIL a terminal criterion is satisfied or the maximum
number of clones is reached
END

```

4.6 Mutation

It has no sense for the clones to be the exact copies of the Best Neuron. Therefore, some mutation must be in effect. The clone, to be a true clone, must have the same parents. So, the basic parameters – the two parents are not changed. A problem is how to change the parameters $a, \dots f$. These changes should be small enough to keep a sufficient similarity of the clone to the original individual (the Best Neuron) and, at the same time, a sufficiently large amount to reach the necessary changes for searching the data space in the neighborhood of the Best Neuron.

The simplest approach spreads value of each parameter randomly around its value in the Best Neuron. For each parameter $a, \dots f$ one can use normal distribution with the mean equal to the value valid for the best neuron and standard deviation equal to $1/6$ of this value. There is also a multiplicative constant for setting up the standard deviation, which a user can set up appropriately; the default value is 1.

4.7 Error Development and Stopping Rule

A stopping rule different from searching for the minimal error on the validating set like in the original GMDH MIA method also follows from the new strategy of network building in the GMC-GMDH method. In our case, the error on the validating set for the best neuron monotonously decreases having no minimum. On the other hand, the indexes of the best neurons became rather distant. For illustration see Figs. 5 and 6. The process can be stopped either when very small change in error is reached, or too many new neurons are built without appearance of a new best neuron or when a predefined number of neurons is depleted.

4.8 Pruning

After learning, the best neuron and all its ancestors have their role in the network function. All others can be removed.

Pruning reduces the size of network graph to the necessary neurons (nodes) and edges. It may appear that some input variables are not used and have no influence to the network's output. This phenomenon appears in the standard GMDH-MIA algorithm as well as in our algorithm. Input variables not used can be omitted and thus, we have an effective means for dimensionality reduction of the data space. If the final GMDH network is not too complex, one can even obtain a closed form of the high order polynomial of the network's function. It was shown already in [2] for the standard GMDH-MIA.

4.9 Network Structure

It can be seen here that resulting network consists of individual neurons connected more or less randomly forming an oriented graph with leaves in

network inputs and the root formed by the network output, the final best neuron. When generating the network, the notion of layer was not defined or used. In the end, the network has no clear layered structure like the network generated by the original GMDH MIA algorithm and most of other GMDH algorithms including those genetically optimized.

4.10 Recall

After learning, the resulting network is a feed-forward network. When a sample is applied to inputs, the outputs of the individual neurons are computed successively according to their order numbers. Thus at any time all information needed for the neuron's output computation is known. The last neuron is the best neuron and its output is the output of the whole network. If the network is used for approximation or prediction, the output gives just the approximation of the value desired. If the network is used as a two class classifier, one must set up a proper threshold θ and an output value larger than or equal to this threshold means that the sample applied belongs to one class or else it belongs to the other class. The value of the threshold can be tuned with respect to the classification error or to the other features of the classifier.

5 Performance Analysis

The experiments described below show that in most cases our genetically modified GMDH algorithm with cloning (GMC GMDH) outperforms 1-NN method in most cases, and in many cases it outperforms naïve Bayes method and also the k -NN method where k equals to the square root of the number of training set points.

The classification ability of the genetically modified GMDH algorithm with cloning (GMC GMDH) was tested using real-life tasks from the UCI Machine Learning Repository [13]. We do not describe these tasks in detail here as all can be found in [13]. For each task, the same approach to testing and evaluation was used as described in [13] for other methods. We also show convergence of the learning process.

For running GMC GMDH program, default parameters were used as follows for all tasks: The no. of neurons generated for stopping computation was 10000. The probability that the new neuron's input was one of the input signals was 10 %; the probability that new neuron's input was one of the already existing neurons was 90 %. The maximal number of clones generated from one parent neuron was limited to $\text{int}(\sqrt{\text{No. of neurons generated up to now}})$. For each method, an optimal threshold θ for the minimal error was used. The fitness function was the reciprocal of the mean absolute error. An experiment was also made with the fitness function equal to the reciprocal of the square of the mean absolute error to make a lower relative probability that the bad neuron is selected as a parent for a new neuron.

In Table 1, the results are shown together with the results for four other well-known and very often used classification methods. In the second column, the cross validation factor is given. The methods for comparison are:

1-NN – standard nearest neighbor method

Table 1. Classification errors for four methods on some data sets from UCI MLR. GMC GMDH is with the fitness function equal to the reciprocal of the mean absolute error, and GMC GMDH 2 is with the fitness function equal to the reciprocal of the squared mean absolute error.

Data set	Algorithm					
	1-NN	sqrt-NN	bay1	LWM1	GMC GMDH	GMC GMDH 2
German	0.4077	0.2028	0.2977	0.7284	0.2947	0.2617
Adult	0.2083	0.2124	0.1637	0.1717	0.1592	0.1562
Brest CW	0.0479	0.0326	0.0524	0.0454	0.0419	0.0436
Shuttle-small	0.0259	0.0828	0.1294	0.0310	0.0259	0.0465
Spam	0.0997	0.1127	0.1427	0.1074	0.1008	0.0917
Splice	0.4035	0.3721	0.2866	0.2587	0.1309	0.1339
Vote	0.1053	0.0602	0.0977	0.0741	0.0667	0.0741

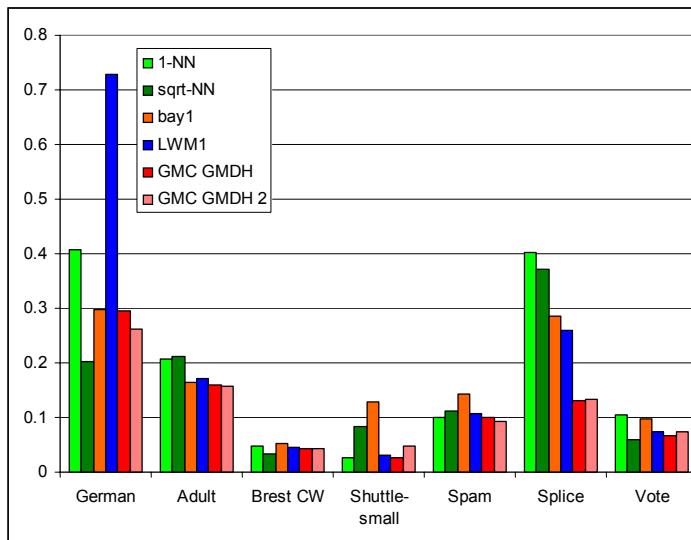


Fig. 5. Classification errors for four methods on some data sets from the UCI MLR. Note that for Shuttle, small data the errors are enlarged ten times in this graph. In the legend, GMC GMDH 2 means the GMC GMDH method with fitness equal to the reciprocal of the square of the mean absolute error.

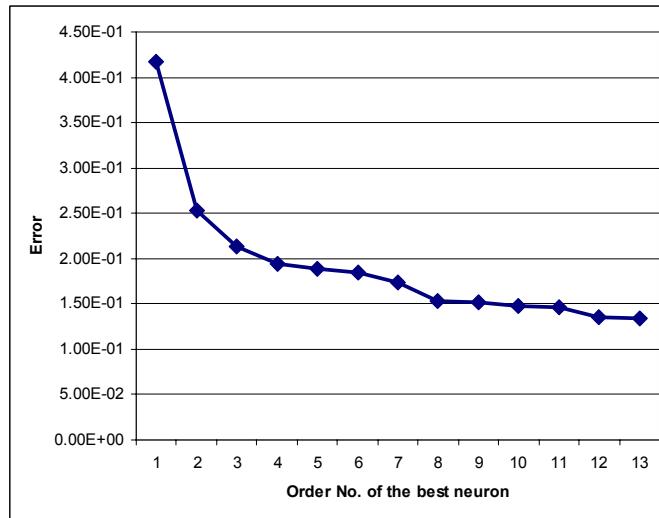


Fig. 6. Error on the validating set count of best neurons successively found during learning

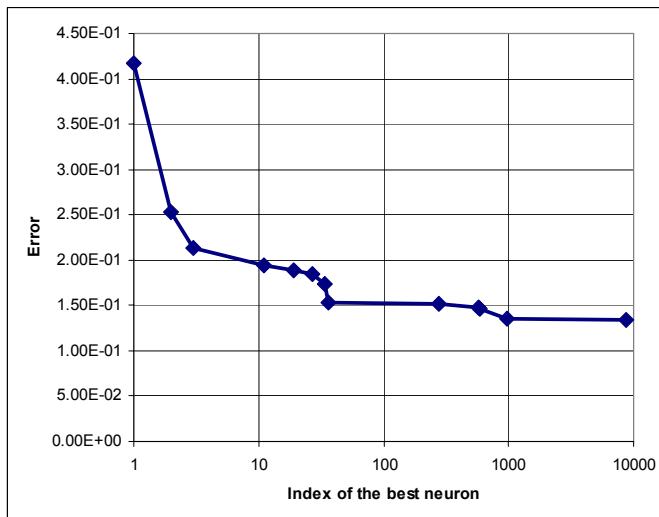


Fig. 7. Error on the validating set vs. the index of the best neuron, i.e. the number of neurons generated

Sqrt-NN – the k -NN method with k equal to the square root of the number of samples of the learning set

Bayes – the naïve Bayes method using ten bins histograms

LWM1 – the learning weighted metrics method [14] modified with nonsmooth learning process.

These results are also depicted in a more distinct form in Fig. 5.

The learning process of the GMC GMDH network is stable and convergent. In Figs. 6 and 7, the successive lowering of the error is depicted. The error is stated on the validating set for the data VOTE from UCI MLR. In difference to the GMDH MIA algorithm, there is no minimum and the stopping rule here is based on the exhausting of the total number of neurons given in advance.

In Fig. 6, it is shown how an error of the best neuron on the validating set decreases with the number of best neurons successively found during learning. It is seen that in this figure, the order number of the best neuron, i.e. true number of neurons generated during learning process, is not seen.

The dependence on the true number of neurons generated is shown in Fig. 7. There, on the horizontal axis are a number of neurons generated and the points on the line show the individual best neurons successively generated. For each point, the corresponding value on horizontal axis is the order number of the best neuron. Note a logarithmic scale on the horizontal axis. It is seen that a new Best Neuron appears in successively longer and longer intervals of neuron generation bringing smaller and smaller improvement in error on the validating set.

6 Conclusion and Discussion

The target of this Chapter was to generalize the idea of the genetically modified GMDH neural network for processing multivariate data appearing in data mining problems and to extend this type of network by cloning. Clones are close, but not identical copies of original individuals. An individual in our case is a new neuron, which appears during the learning process.

The new genetically modified GMDH method with cloning (GMC GMDH) has no tough layered structure. During learning, when a new neuron is added it is randomly connected to two already existing neurons or to network inputs with some probability derived from the fitness and keeping some minor probability that an input is selected. When the inputs are assigned to a new neuron, the six parameters a, \dots, f of the quadratic polynomial (1) are computed. For it, the least mean squared method is used as in the basic GMDH MIA approach using the training part of the learning set. The fitness, i.e. the reciprocal of the mean absolute error, is evaluated using the validating set. If a new neuron generated is found to be the best neuron, the clones are derived to reach even better fitness. The clones have the same two “parent” signals as the best neuron. The mutation operation slightly changes values of six parameters of the best neuron and thus the clones are similar to, but not exact copies of the best neuron. When a clone is found to be the best neuron, the clone generating process is broken and immediately starts again with this new best neuron.

It was found that our expectation that the true optimum may lie somewhere in the neighborhood of parameters of the best neuron computed by linear regression holds. We have shown that some assumptions for least squares method for linear regression are not met and thus the solution does not represent an exact optimum. Cloning is a useful technique to get closer to the true optimum. Cloning with large changes of parameters has a small effect, but with small changes a new best neuron often arises. From it, one can deduce that in practice differences between pairs of parameters corresponding to a minimum found by linear regression and a minimum found by cloning is not too large but not negligible either.

Classification errors for four methods on some data sets from the UCI MLR are depicted in Fig. 5. Note that for Shuttle, small data the errors are enlarged ten times in this graph. In Table 1 and in Fig. 5 it is seen that the GMC GMDH method outperforms other methods in the tasks Adult, Shuttle small, and Splice or nearly outperforms Brest CW, Spam, and Vote. The GMC GHMDH is the second best with a very small difference with respect to the best method considered. It is the second best in the task German. The experiments described above show that the GMC GMDH approach outperforms 1-NN method in most cases, and in many cases outperforms the naïve Bayes method and also the k-NN method where k equals the square root of the number of training set samples.

It is also seen here that for a fitness function equal to the second power of the mean absolute error, the error may be slightly different from the case of the fitness function equal to the mean absolute error. Then, the sensitivity to fitness function definition is rather small in these cases.

In Figs. 6 and 7 it is seen that the learning process converges rather fast, i.e. for a relatively small number of neurons generated, the error on the validating set decreases fast and then for a large number of neurons generated, the error decreases only slightly. Practical tests show that a further enlargement of the number of neurons generated up to the order of a hundred thousand has no practical effect. As there is no searching or sorting like in the nearest neighbor-based methods or in the classical GMDH MIA algorithm, the GMC GMDH is much faster than the methods mentioned, especially for large learning sets.

The genetically modified GMDH method with cloning (GMC GMDH) presented here appears to be an efficient approach giving reliably good results better or comparable to the best results obtained by other methods. The Genetically modified GMDH method is an elegant idea how to improve the efficiency of the popular GMDH MIA method. It is based on the usage of a selection principle of genetic algorithms instead of systematic assignment of all pairs formed by neurons of the last layer. Thus, all neurons once generated remain potential parents for new neurons during the whole learning process. Also, each input signal may be used with some probability as a parent signal for a new neuron. Thus, the strictly layered structure of the GMDH algorithm disappears, as any new neuron can be connected to the output of any already existing neuron or even to any input of the network. The essential

advantage of the genetically modified GMDH with cloning is that one need not set up the number of the best neurons selected in the newly generated layer and thus indirectly control the learning time and size of the network as in the standard GMDH MIA algorithm. The only limitations of the GMC GMDH method are the learning time or the memory size.

Here, we presented a new method of building the GMDH network with genetic selection of parents for each new neuron and with cloning of the best neuron. We have shown efficiency and good behavior for two class classification tasks. As GMDH networks also serve as an approximator and predictor, there is the open possibility to use GMC GMDH for approximating and predicting tasks.

Acknowledgements

This work was supported by the Ministry of Education of the Czech Republic under the project Center of Applied Cybernetics No. 1M0567, and No. MSM6840770012 Transdisciplinary Research in the Field of Biomedical Engineering II.

References

1. Ivakhnenko, A.G.: Polynomial Theory of Complex System. IEEE Trans. on Systems, Man and Cybernetics SMC-1(4), 364–378 (1971)
2. Farlow, S.J.: Self-Organizing Methods in Modeling. GMDH Type Algorithms. Marcel Dekker, Inc., New York (1984)
3. Tamura, H., Kondo, T.: Heuristics-free group method of data handling algorithm of generating optimal partial polynomials with application to air pollution prediction. Int. J. Systems Sci. 11(9), 1095–1111 (1980)
4. Ivakhnenko, A.G., Müller, J.A.: Present State and New Problems of Further GMDH Development. SAMS 20, 3–16 (1995)
5. Ivakhnenko, A.G., Ivakhnenko, G.A., Müller, J.A.: Self-Organization of Neural Networks with Active Neurons. Pattern Recognition and Image Analysis 4(2), 177–188 (1994)
6. Ivakhnenko, A.G., Wunsch, D., Ivakhnenko, G.A.: Inductive Sorting/out GMDH Algorithms with Polynomial Complexity for Active neurons of Neural network. IEEE 6(99), 1169–1173 (1999)
7. Nariman-Zadeh, N., et al.: Modeling of Explosive Cutting process of Plates using GMDH-type neural network and Singular value Decomposition. Journ. of material processes technology 128(1-3), 80–87 (2002)
8. Hiassat, M., Mort, N.: An evolutionary method for term selection in the Group Method of Data Handling. Automatic Control & Systems Engineering, University of Sheffield, www.maths.leeds.ac.uk/statistics/workshop/lasr2004/Proceedings/hiassat.pdf
9. Oh, S.K., Pedrycz, W.: The Design of Self-organizing Polynomial Neural Networks. Information Sciences 141(3-4), 237–258 (2002)

10. Hakl, F., Jiřina, M., Richter-Was, E.: Hadronic tau's identification using artificial neural network. ATLAS Physics Communication, ATL-COM-PHYS-2005-044, <http://documents.cern.ch/cgi-bin/setlink?base=atlnot&categ=Communication&id=com-phys-2005-044> (last revision: August 26, 2005)
11. Ji, Z.: Negative Selection Algorithms: From the Thymus to V-Detector. Dissertation Presented for the Doctor of Philosophy Degree. The University of Memphis (August 2006)
12. Guney, K., Akdagli, A., Babayigit, B.: Shaped-beam pattern synthesis of linear antenna arrays with the use of a clonal selection algorithm. Neural Network world 16, 489–501 (2006)
13. Merz, C.J., Murphy, P.M., Aha, D.W.: UCI Repository of Machine Learning Databases. Dept. of Information and Computer Science, Univ. of California, Irvine (1997), <http://www.ics.uci.edu/~mlearn/MLrepository.html>
14. Paredes, R., Vidal, E.: Learning Weighted Metrics to Minimize Nearest-Neighbor Classification Error. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(7), 1100–1110 (2006)
15. Zhang, M., Cieselski, V.: Using Bask propagation Algorithm and Genetic Algorithm to Train and refine neural networks for Object Detection. In: Bench-Capon, T.J.M., Soda, G., Tjoa, A.M. (eds.) DEXA 1999. LNCS, vol. 1677, pp. 626–635. Springer, Heidelberg (1999)
16. Lu, C., Shi, B.: Hybrid back-propagation/genetic algorithm for multilayer feed forward neural networks. In: 5th International Conference on Signal Processing Proceedings, 2000. WCCC ICSP 2000, vol. 1, pp. 571–574. IEEE, Los Alamitos (2000)
17. Kalous, R.: Evolutionary operators on ICodes. In: Hakl, F. (ed.) Proceedings of the IX PhD. Conference. Institute of Computer Science, Academy of Sciences of the Czech Republic, Matfyzpress, Prague, Paseky nad Jizerou, September 29-October 1, pp. 35–41 (2004)
18. Vasechkina, E.F., Yarin, V.D.: Evolving polynomial neural network by means of genetic algorithm: some application examples. Complexity International 09, 1–13 (2001), <http://www.complexity.org.au/vol09/vasech01/>
19. Wikipedia - Gauss–Markov theorem (2007), http://en.wikipedia.org/wiki/Gauss-Markov_theorem

Author Index

- Berretta, Regina 149
Bollenbeck, Felix 197
- Carvalho, André C.P.L.F. de 177
Castellano, Gloria 243
- Elkassas, A.S.I. 317
- Fonlupt, Cyril 339
Freitas, Alex A. 177, 339
- Gao, Xinbo 223
- Júnior, Luiz Otávio Murta 127
Javadi, A.A. 317
- Jiřina, Marcel 39, 359
Johnson, Colin G. 339
Jr., Marcel Jiřina 39, 359
- Kasahara, Viviani Akemi 57
- Li, Guo-Zheng 3
Li, Xuelong 223
- Moscato, Pablo 149
- Nicoletti, Maria do Carmo 57
- Otero, Fernando 339
- Pancerz, Krzysztof 79
- Ravetti, Martín Gómez 149
Robilliard, Denis 339
- Sant'Anna, Annibal Parracho 107
Segond, Marc 339
Seiffert, Udo 197
- Tan, T.P. 317
Tao, Dacheng 223
Tinós, Renato 127
Torrens, Francisco 243
- Xiao, Bing 223
- Zeng, Xue-Qiang 3