Petra Perner (Ed.)

LNAI 7376

Machine Learning and Data Mining in Pattern Recognition

8th International Conference, MLDM 2012
Berlin, Germany, July 2012
Proceedings

Lecture Notes in Artificial Intelligence 7376

Subseries of Lecture Notes in Computer Science

Petra Perner (Ed.)

Machine Learning and Data Mining in Pattern Recognition

8th International Conference, MLDM 2012
Berlin, Germany, July 13-20, 2012
Proceedings

Springer

Volume Editor

Petra Perner
Institute of Computer Vision
and Applied Computer Sciences, IBaI
Kohlenstr. 2, 04107 Leipzig, Germany
E-mail: pperner@ibai-institut.de

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-31536-7 e-ISBN 978-3-642-31537-4
DOI 10.1007/978-3-642-31537-4

Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012940654

CR Subject Classification (1998): I.2, F.4, I.4, I.5, H.3, H.2.8

LNCS Sublibrary: SL 7 – Artificial Intelligence

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

Printed on acid-free paper

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

Preface

The eighth event of the International Conference on Machine Learning and Data Mining (MLDM) was held in Berlin (www.mldm.de) under the umbrella of the World Congress on "The Frontiers in Intelligent Data and Signal Analysis, DSA 2012".

For this edition the Program Committee received 212 submissions. After the peer-review process, we accepted 71 high-quality papers for oral presentation, from which 51 are included in this proceedings book. The topics range from theoretical topics for classification, clustering, association rule and pattern mining to specific data mining methods for the different multimedia data types such as image mining, text mining, video mining and Web mining. Extended versions of selected papers will appear in the*International Journal Transactions on Machine Learning and Data Mining* (www.ibai-publishing.org/journal/mldm).

Eight papers were selected for poster presentations and are published in the *MLDM Poster Proceedings* by *ibai-publishing* (www.ibai-publishing.org).

A tutorial on Data Mining, a tutorial on Case-Based Reasoning, a tutorial on Intelligent Image Interpreation and Computer Vision in Medicine, Biotechnology, Chemistry and Food Industry and a tutorial on Standardization in Immunofluorescence were held before the conference.

We were pleased to give out the best paper award for the fourth time this year (www.mldm.de). The final decision was made by the Best Paper Award Committee based on the presentation by the authors and the discussion with the auditorium. The ceremony took place at the end of the conference. This prize is sponsored by ibai solutions (www.ibai-solutions.de), one of the leading companies in data mining for marketing, Web mining and e-commerce.

The conference was rounded up by an outlook of new challenging topics in machine learning and data mining before the Best Paper Award ceremony.

We would like to thank the members of the Institute of Applied Computer Sciences, Leipzig, Germany (www.ibai-institut.de), who handled the conference as secretariat. We appreciate the help and understanding of the editorial staff at Springer Verlag, and in particular Alfred Hofmann, who supported the publication of these proceedings in the LNAI series.

Last, but not least, we wish to thank all the speakers and participants who contributed to the success of the conference. See you in 2013 in New York to the next World Congress on "The Frontiers in Intelligent Data and Signal Analysis, DSA2013" (www.worldcongressdsa.com) will be held in New York, in 2013, combining under its roof the following three events: International Conferences Machine Learning and Data Mining (MLDM), the Industrial Conference on Data Mining (ICDM), and the International Conference on Mass Data Analysis of Signals and Images in Medicine, Biotechnology, Chemistry and Food Industry (MDA).

July 2012 Petra Perner

International Conference on Machine Learning and Data Mining, MLDM 2012

Program Chair

Petra Perner	IBaI Leipzig, Germany

Program Committee

Agnar Aamodt	NTNU, Norway
Jacky Baltes	University of Manitoba, Canada
Christoph F. Eick	University of Houston, USA
Ana Fred	Technical University of Lisbon, Portugal
Giorgio Giacinto	University of Cagliari, Italy
Makato Haraguchi	Hokkaido University Sapporo, Japan
Robert J. Hilderman	University of Regina, Canada
Eyke Hüllermeier	University of Marsburg, Germany
Atsushi Imiya	Chiba University, Japan
Abraham Kandel	University of South Florida, USA
Dimitrios A. Karras	Chalkis Institute of Technology, Greece
Adam Krzyzak	Concordia University, Montreal, Canada
Brian Lovell	University of Queensland, Australia
Mariofanna Milanova	University of Arkansas at Little Rock, USA
Thang V. Pham	University of Amsterdam, The Netherlands
Maria da Graca Pimentel	Universidade de Sao Paulo, Brazil
Petia Radeva	Universitat Autonoma de Barcelona, Spain
Michael Richter	University of Calgary, Canada
Fabio Roli	University of Cagliari, Italy
Linda Shapiro	University of Washington, USA
Sameer Singh	Loughborough University, UK
Harald Steck	Bell Laboratoris, USA
Francesco Tortorella	Università degli Studi di Cassino, Italy
Patrick Wang	Northeastern University, USA

Additional Reviewers

Pål Sætrom (Paal Saetrom)	NTNU, Norway
Gleb Sizov	NTNU, Norway
Theoharis Theoharis	NTNU, Norway
Luigi Atzori	University of Cagliari, Italy
Davide Ariu	University of Cagliari, Italy
Giuliano Armano	University of Cagliari, Italy

Battista Biggio	University of Cagliari, Italy
Igino Corona	University of Cagliari, Italy
Luca Didaci	University of Cagliari, Italy
Giorgio Fumera	University of Cagliari, Italy
Danilo Pani	University of Cagliari, Italy
Ignazio Pillai	University of Cagliari, Italy
Luca Piras	University of Cagliari, Italy
Riccardo Satta	University of Cagliari, Italy
Roberto Tronci	University of Cagliari, Italy
Eloisa Vargiu	Barcelona Digital Technolgic Centre, Spain

Table of Contents

Theory

Theory: Evaluation of Models and Performance Evaluation Methods

Theory: Learning

Theory: Clustering

WebMining

Image Mining

Data Mining in Biometry and Security

Data Mining in Medicine

Data Mining in Environment and Water Quality Detection

Image Mining in Medicine

Text Mining

Data Mining in Network

Applications of Data Mining in Process Automation, Organisation Change Management, Telecommunication and Post Services

Data Mining in Biology

Bayesian Approach to the Concept Drift in the Pattern Recognition Problems

Pavel Turkov[1], Olga Krasotkina[1], and Vadim Mottl[2]

[1] Tula State University, 92 Lenina Ave., Tula, 300600 Russia
[2] Computing Center of the Russian Academy of Science, 40 Vavilov St., Moscow, 119333 Russia

Abstract. We can face with the pattern recognition problems where the influence of hidden context leads to more or less radical changes in the target concept. This paper proposes the mathematical and algorithmic framework for the concept drift in the pattern recognition problems. The probabilistic basis described in this paper is based on the Bayesian approach to the estimation of decision rule parameters. The pattern recognition procedure derived from this approach uses the general principle of the dynamic programming and has linear computational complexity in contrast to polynomial computational complexity in general kind of pattern recognition procedure.

1 Introduction

As a rule in the pattern recognition problem the properties of the regarded concept are supposed to be constant during the learning process. However, we can face with other problems where some hidden context occurs. So such problems exist in case of the data-handling procedure extended in time The influence of this context leads to more or less radical changes in the target concept. In data mining this situation is known as concept drift. In this case the classical pattern recognition methods are inapplicable.

There are some methods for pattern recognition problem under concept drift [1]. A certain quantity of such methods include a single classifier. Usually these methods use a sliding window to choose a group of new instances to train a model, a group size called window length (size). In different methods during the learning procedure this parameter can be constant, for example FLORA [2], or vary, then such method contains drift detection mechanism, e.g. ADWIN [3].

Another group is the ensemble-based methods. They have become very popular lately since they have a low error in comparison with single classifier methods. Learned on the training data the set of classifiers are combined as voting or weight voting. Ensemble-based approaches can be constructed in two ways: given new data,

1. retrain the old ensemble members on new data, e.g. Accuracy Weighted Ensemble (AWE) [4];

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 1–10, 2012.

2. drop one worst classifier of ensemble and add a new classifier learned on incoming data, such as a streaming ensemble algorithm (SEA) [5].

In general, we can note that the existing algorithms with a single classifier are more or less heuristic and a certain set of this heuristics is determined by the specificity of the current task. On the other hand, ensemble-based methods are sometimes too difficult. Also the accurate mathematical statement of concept drift doesn't exist. We propose the probabilistic basis for the problem of concept drift. This basis results from Bayesian approach to the pattern recognition problem. The method received from this approach uses the general principle of the dynamic programming procedure and has the computation complexity proportional to the length of the training sequence.

The remainder of the paper is organized as follows. In Section 2, we present the problem description in terms of the Bayesian approach. Section 3 gives the method based on the dynamic programming procedure for estimation of the decision rule. The experimental results of the method application by the model data set are described in Section 4. Section 5 concludes the paper.

2 Bayesian Approach to the Problem of Concept Drift for the Pattern Recognition Problem

Let every instance of the universe $\omega \in \Omega$ be presented by a point in the linear feature space $\mathbf{x}(\omega) = \big(x^1(\omega), \ldots, x^n(\omega)\big) \in \mathbb{R}^n$, and its hidden membership in one of two classes be determined by an index value of the class $y(\omega) \in \{1, -1\}$. We will proceed from the classical approach to the training problem [7] based on treating the model of the universe in the form of a discriminant function. Such function is defined as a hyperplane having a priori unknown direction vector $\mathbf{a}$ and threshold b: $f\big(\mathbf{x}(\omega)\big) = \mathbf{a}^T\mathbf{x} + b$ is primary > 0 if $y(\omega) = 1$, and < 0 if $y(\omega) = -1$. But this problem statement doesn't take into account the presence of concept drift. The drift of the target concept indicates some changes in the universe and consequently our modeling hyperplane must be changed too. Therefore let the behavior of the universe with concept drift be described by time-varying hiperplane $f_t\big(\mathbf{x}(\omega)\big) = \mathbf{a}_t^T\mathbf{x} + b_t$ where $\mathbf{a}_t$ and b_t are the unknown time functions. So every instance $\omega \in \Omega$ is considered only together with the indication of time point when this instance was presented (ω, t). As a result the training set represents as the set of triplet $\big\{(\mathbf{X}_t \in \mathbb{R}^n,\ \mathbf{Y}_t,\ t)\big\}_{t=1}^{T}$, $(\mathbf{X}_t, \mathbf{Y}_t) = \{(\mathbf{x}_{k,t}, y_{k,t})\}_{k=1}^{N_t}$ - a subset of instances, entered in time point t.

In such definition the problem of learning turns into the analysis problem of two-component time series, as it is needed to estimate the hidden component $(\mathbf{a}_t, b_t)$ by the observable component $(\mathbf{X}_t, \mathbf{Y}_t)$. This is the standard problem of time series analysis, the specific character of which consists in the supposed model of connection between hidden and observable components. N. Wiener in [9] introduced classification for the estimation problems of hidden component. According to this classification we can distinguish two types of the learning problems.

Filtering problem of the training set. Let a new object appear at the time moment T when the feature vectors and class-membership indices of the previous are already registered $\{(\mathbf{X}_t,\ \mathbf{Y}_t,\ t)\}_{t=1}^{T}$ including the current moment T. It is required to recurrently estimate the parameters of the discriminant hyperplane $(\mathbf{a}_T, b_T)$ at each time moment T immediately in the process of observation.

Interpolation problem of the training set. Let the training time series be completely registered in some time interval $\{(\mathbf{X}_t,\ \mathbf{Y}_t,\ t)\}_{t=1}^{T}$ before its processing starts. It is required to estimate the time-varying parameters of the discriminant hyperplane in the entire observation interval $\{(\mathbf{a}_t,\ b_t)\}_{t=1}^{T}$

Let us formulate the probabilistic description of the problem. Let $\phi(\mathbf{x}_{j,t}|\ y_{j,t}, \mathbf{a}_t, b_t)$ with $y_t = \pm 1$ be two parametric family of probability densities in the joint feature space $\mathbb{X}_1 \times \cdots \times \mathbb{X}_n$ associated with discriminant hyperplane $\mathbf{a}_t^T \mathbf{x}_{j,t} + b_t \geq 0$ and concentrated predominantly on opposite sides of it. We shall consider that the improper densities

$$\phi_y(\mathbf{x}_{j,t}|\mathbf{a}_t, b_t) = \exp\left[-\frac{1}{2\sigma_y^2}\left(1 - y_{j,t}(\mathbf{a}_t^T \mathbf{x}_{j,t} + b_t)\right)^2\right] \tag{1}$$

express the assumption that the random feature vectors of both classes of objects are uniformly distributed along the separating hyperplane with the parameter σ, controlling the probability of incorrect location.

For all training instances $\mathbf{X}_t$ and their class labels $\mathbf{Y}_t$ obtained in the time point t the joint distribution density function is:

$$\Phi(\mathbf{X}_t|\mathbf{Y}_t, \mathbf{a}_t, b_t) = \prod_{j=1}^{N_t} \phi_{y_j}(\mathbf{x}_j|\mathbf{a}_t, b_t).$$

As there isn't any a priori information about values $\mathbf{a}_1, b_1$ we will suppose that at the zero time moment a priori distributions of the separating hyperplane parameters are uniform. So it means that the distributions are constant on all number axis and consequently its integral isn't equal to one. Such density functions are called improper [11].

The key element of the proposed Bayesian approach to the concept drift [10] is treating the time-varying parameters of hyperplane as hidden processes assumed a priori to possess Markov property

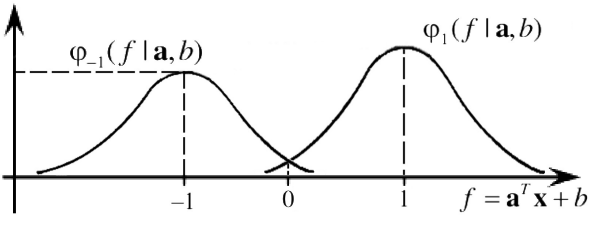


Fig. 1. Probability density functions along direction vector **a**

$$\begin{aligned}\mathbf{a}_t &= q\mathbf{a}_{t-1} + \boldsymbol{\xi}_t, M(\boldsymbol{\xi}_t) = \mathbf{0}, M(\boldsymbol{\xi}_t\boldsymbol{\xi}_t^T) = d\mathbf{I},\\ b_t &= b_{t-1} + \nu_t, M(\nu_t) = 0, M(\nu_t^2) = d',\\ q &= \sqrt{1-d}, 0 \le q < 1,\end{aligned} \tag{2}$$

where variances d and d' determine the assumed hidden dynamics of the concept. The $\boldsymbol{\xi}_t$ and ν_t are the white noises with zero mathematical expectations. Equation (2) determines, actually, the state-space model of the dynamic system, whereas (1) plays the role of its observation model.

The a priori distribution density of the hidden sequence of hyperplane parameters will be:

$$\Psi(\mathbf{a}_t, b_t, t = 2, \ldots, T) = \prod_{t=2}^{T} \psi_t(\mathbf{a}_t, b_t | \mathbf{a}_{t-1}, b_{t-1})$$

$$\begin{aligned}&\psi_t(\mathbf{a}_t, b_t|\mathbf{a}_{t-1}, b_{t-1}) \propto \mathrm{N}(\mathbf{a}_t|\sqrt{1-d}\mathbf{a}_{t-1}, d\mathbf{I})\mathrm{N}(b_t|b_{t-1}, d') =\\ &= \frac{1}{d^{n/2}(2\pi)^{n/2}} \exp\left(-\frac{1}{2d}(\mathbf{a}_t - \sqrt{1-d}\mathbf{a}_{t-1})^T(\mathbf{a}_t - \sqrt{1-d}\mathbf{a}_{t-1})\right) \cdot\\ &\cdot \frac{1}{2\pi d'} \exp\left(-\frac{1}{2d'}(b_t - b_{t-1})^2\right).\end{aligned}$$

So we have defined, first, the conditional a priori distribution of the hidden sequence of hyperplane parameters $\Psi(\mathbf{a}_t, b_t, t = \overline{2,T})$ and, second, the conditional distribution of the training sample $\Phi(\mathbf{X}_t|\mathbf{Y}_t, \mathbf{a}_t, b_t)$. It is clear that the a posteriori distribution density of the hidden sequence of direction vectors and thresholds will be proportional to product:

$$P(\mathbf{a}_t, b_t|\mathbf{Y}_t, t = 2, \ldots, T) \propto \Psi(\mathbf{a}_t, b_t, t = \overline{2,T})\Phi(\mathbf{X}_t|\mathbf{Y}_t, \mathbf{a}_t, b_t, t = \overline{2,T}) \tag{3}$$

It appears natural to take the maximum point of this a posteriori density as the sequence of time-varying hyperplane parameters in the model:

$$\begin{aligned}(\hat{\mathbf{a}}_t, \hat{b}_t) &= \arg\max_{\mathbf{a}_t, b_t} P(\mathbf{a}_t, b_t|\mathbf{Y}, t = 2, \ldots, T) =\\ &= \arg\max_{\mathbf{a}_t, b_t} \Psi(\mathbf{a}_t, b_t, t = \overline{2,T})\Phi(\mathbf{X}_t|\mathbf{Y}_t, \mathbf{a}_t, b_t, t = \overline{2,T})\end{aligned}$$

Theorem 1. *The maximum point of the a priori density (3) by $\mathbf{a}_t, b_t$ is the minimum point of the criterion:*

$$J_T(\mathbf{a}_t, b_t, t = 0, \ldots, T) = \min_{\mathbf{a}_t, b_t, t=1,\ldots,T} \left[\sum_{t=1}^{T} \sum_{j=N_{t-1}+1}^{N_t} C_y \left(1 - y_j(\mathbf{a}_t^T\mathbf{x}_j + b_t)\right)^2 + \right. \tag{4}$$

$$\left. + \frac{1}{d}\sum_{t=2}^{T}(\mathbf{a}_t - \sqrt{1-d}\mathbf{a}_{t-1})^T(\mathbf{a}_t - \sqrt{1-d}\mathbf{a}_{t-1}) + \frac{1}{2d'}\sum_{t=2}^{T}(b_t - b_{t-1})^2\right]$$

where $N_0 = 0$.

The first term $\sum_{t=1}^{T}\sum_{j=N_{t-1}+1}^{N_t} C_y\left(1-y_j(\mathbf{a}_t^T\mathbf{x}_j+b_t)\right)^2$ of the criterion stands for the approximation of observation y_t. As the second $\frac{1}{d}\sum_{t=2}^{T}(\mathbf{a}_t-\sqrt{1-d}\mathbf{a}_{t-1})^T(\mathbf{a}_t-\sqrt{1-d}\mathbf{a}_{t-1})$ and third $\frac{1}{2d'}\sum_{t=2}^{T}(b_t-b_{t-1})^2$ terms are responsible for overall time-volatility of direction vector and threshold.

The criterion (4) is pair-wise separable, i.e. representing a sum of private functions every of which depends on the variables connected with one or two time points in their increasing order. Let us introduce the following notation for the criterion (4):

$$J(\mathbf{z}_1,\ldots,\mathbf{z}_T)=\sum_{t=1}^{T}(\mathbf{z}_t-\mathbf{z}_t^0)^T\mathbf{Q}_t(\mathbf{z}_t-\mathbf{z}_t^0)+ \\ +\sum_{t=2}^{T}(\mathbf{z}_t-\mathbf{A}\mathbf{z}_{t-1})^T\mathbf{U}(\mathbf{z}_t-\mathbf{A}\mathbf{z}_{t-1}) \tag{5}$$

where

$$\mathbf{z}_t=\begin{bmatrix}\mathbf{a}_t\\ b_t\end{bmatrix};\mathbf{z}_t^0=\left(\mathbf{Q}_t{}^T\mathbf{Q}_t\right)^{-1}\mathbf{Q}_t\sum_{j=N_{t-1}}^{N_t}\mathbf{g}_j$$

$$\mathbf{Q}_t=\sum_{j=N_{t-1}}^{N_t}C_y\mathbf{g}_j\mathbf{g}_j^T;\mathbf{g}_j=\begin{bmatrix}y_j\mathbf{x}_j\\ y_j\end{bmatrix}$$

$$\mathbf{U}=\begin{bmatrix}\frac{1}{d} & 0 & \ldots & 0 & 0\\ 0 & \frac{1}{d} & \ldots & 0 & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \ldots & \frac{1}{d} & 0\\ 0 & 0 & \ldots & 0 & \frac{1}{d'}\end{bmatrix};\mathbf{A}=\begin{bmatrix}\sqrt{1-d} & 0 & \ldots & 0 & 0\\ 0 & \sqrt{1-d} & \ldots & 0 & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & \ldots & \sqrt{1-d} & 0\\ 0 & 0 & \ldots & 0 & \sqrt{1-d'}\end{bmatrix}$$

3 Dynamic Programming Procedure for the Estimation of the Decision Rule Parameters under Concept Drift

For the parameters estimation we use the general principle of dynamic programming, based on the concept of a sequence of Bellman functions $\tilde{J}_t(\mathbf{z}_t)=\min_{\mathbf{z}_0,\ldots,\mathbf{z}_{t-1}}J_t([\mathbf{z}_s]_{s=1}^{t})$, $[\mathbf{z}_s\in Z_s]_{s=1}^{t-1}$, connected with partial criteria

$$J(\mathbf{z}_1,\ldots,\mathbf{z}_t)=\sum_{s=1}^{t}\zeta_s(\mathbf{z}_s)+\sum_{s=2}^{t}\gamma_s(\mathbf{z}_{s-1},\mathbf{z}_s). \tag{6}$$

Bellman functions are recurrently evaluated for $t=1,\ldots,T$ by the fundamental property of Bellman function

$$\tilde{J}_t(\mathbf{z}_t)=\zeta_t(\mathbf{z}_t)+\min_{\mathbf{z}_{t-1}}\left[\gamma_t(\mathbf{z}_{t-1},\mathbf{z}_t)+\tilde{J}_{t-1}(\mathbf{z}_{t-1})\right], \tag{7}$$

which is called the direct recurrence relation. The function sequence starts from the first function: $\tilde{J}_1(\mathbf{z}_1) = \zeta_1(\mathbf{z}_1)$. The optimal value of the last variable is directly calculated as

$$\hat{\mathbf{z}}_T = \arg\min_{\mathbf{z}_T} \tilde{J}_T(\mathbf{z}_T)$$

Search of other optimal values is carried out in the reverse order by the following expression:

$$\hat{\mathbf{z}}_{t-1} = \tilde{\mathbf{z}}_{t-1}(\hat{\mathbf{z}}_t) = \arg\min_{\mathbf{z}_{t-1}} \left[\gamma_t(\mathbf{z}_{t-1}, \hat{\mathbf{z}}_t) + \tilde{J}_{t-1}(\mathbf{z}_{t-1})\right] \tag{8}$$

where recurrence relations $\tilde{\mathbf{z}}_{t-1}(\hat{\mathbf{z}}_t)$ are determined and stored when the Bellman functions were calculated in the forward order. The procedure described above is the classical dynamic programming procedure called the "forward then back" procedure [12].

In the case of quadratic functions $\zeta_t(\mathbf{z}_t)$ and $\gamma_t(\mathbf{z}_{t-1}, \mathbf{z}_t)$ all Bellman functions are quadratic too and consequently can be written as

$$\tilde{J}_t(\mathbf{z}_t) = (\mathbf{z}_t - \tilde{\mathbf{z}}_t)^T \tilde{\mathbf{Q}}_t (\mathbf{z}_t - \tilde{\mathbf{z}}_t) + \tilde{c}_t. \tag{9}$$

Then the "forward" procedure consists in the reccurent recalculation of the quadratic form parameters $\tilde{\mathbf{Q}}_t$, $\tilde{\mathbf{z}}_t$, $\tilde{c}_t$. The value $\tilde{\mathbf{z}}_t$ from the last Bellman function is equal the optimal value of the last variable: $\hat{\mathbf{z}}_T = \tilde{\mathbf{z}}_t$.

Apply items mentioned above by the parameters estimation for the problem with concept drift (6). For this rewrited expression (5) in the form (6) use the following symbols:

$$\zeta_t(\mathbf{z}_t) = (\mathbf{z}_t - \mathbf{z}_t^0)^T \mathbf{Q}_t (\mathbf{z}_t - \mathbf{z}_t^0)$$
$$\gamma_t(\mathbf{z}_{t-1}, \mathbf{z}_t) = (\mathbf{z}_t - \mathbf{A}\mathbf{z}_{t-1})^T \mathbf{U} (\mathbf{z}_t - \mathbf{A}\mathbf{z}_{t-1}).$$

It is clear that this functions are quadratic and therefore the Bellman functions for our problem are determined in the form (9). On the other hand the Bellman function for the time moment t can be evaluated by the direct recurrence relation (7). Thus by $t = 1$ Bellman function parameters have a trivial view: $\tilde{\mathbf{z}}_1 = \mathbf{z}_1^0; \tilde{\mathbf{Q}}_1 = \mathbf{Q}_1; \tilde{c}_1 = 0$. Later they are recurrently recounted for the $t = 2, \ldots, T$:

$$\tilde{\mathbf{Q}}_t = \tilde{\mathbf{Q}}_{t-1} \left(\mathbf{AUA} + \tilde{\mathbf{Q}}_{t-1}\right)^{-1} \mathbf{U} + \mathbf{Q}_t \tag{10}$$
$$\tilde{\mathbf{z}}_t = \tilde{\mathbf{Q}}_t^{-1} \left[\left(\mathbf{AUA} + \tilde{\mathbf{Q}}_{t-1}\right)^{-1} \mathbf{AU}\tilde{\mathbf{Q}}_{t-1}\tilde{\mathbf{z}}_{t-1} + \mathbf{Q}_t \mathbf{z}_t^0\right].$$

The required hyperplane parameters in the time moment T are determined the optimal value of the last variable $\hat{\mathbf{z}}_T = \tilde{\mathbf{z}}_T$. With regard to the proposed classification of the learning problems this optimal parameters are the solution of the filtering problem. Other optimal values $\hat{\mathbf{z}}_1, \hat{\mathbf{z}}_2, \ldots, \hat{\mathbf{z}}_{T-1}$ are easily found in the reverse order by using backward recurrence relation

$$\hat{\mathbf{z}}_{t-1} = \left(\mathbf{AUA} + \tilde{\mathbf{Q}}_{t-1}\right)^{-1} \left(\mathbf{AU}\hat{\mathbf{z}}_t + \tilde{\mathbf{Q}}_{t-1}\tilde{\mathbf{z}}_{t-1}\right)$$

So we find the solutions of the interpolation problem.

4 Experimental Evaluation

4.1 "Ground-Truth" Experiments

To verify the proposed method we used the synthetic data which can be presented in the form of two normal distributions. The class labels in these data were equidistributed and take values from the set $\{-1, 1\}$. The two instance features generated from the normal distribution for each class. In the zero moment these distributions had the equal variation 1 and mathematical expectation 3.5 and 6.5 respectively. The distance between the centers of distributions was selected according to the 3-sigma rule. With time the centers of distributions were changed, namely, rotated around the origin of coordinates in the two-dimensional feature space. The training set was comprised by the instances generated for 50 consecutive time moments of 100 instances each. For the test we created the sample with 2000 instances in the next 51st time moment. After learning according to the proposed method the resulting decision rule was applied to the testing instances and the per cent of erroneously classified instances in each class was calculated. The selection of method parameters was realized by the error value on the test sample.

In the Table 1 the experimental results are demonstrated by the different values of variations d and d' in the criterion (4). As the results given in the table show, for the method parameters close to 0 the resulting decision rule classified all instances as belonging to the only class. In this case the very large penalty in the criterion doesn't allow the decision rule to adapt to the toward changes in the data description. Another extreme case was under $d = 1; d' = 1$. Then the reverse situation occurred, when the classifier easily forgot the old information about instances. Since we used the most favorable structure of data the error is rather small. The last row of the Table 1 describes the results by the optimal values of parameters.

To compare, the decision rule being trained on all model sample at once by the classical SVM method has the classification error 0.099% for the -1-class and 13.653% for the $+1$-class. Even in this simple model experiment the SVM results show that the pattern recognition methods aren't applicable in the problems under concept drift. So the proposed method works in the problem under concept drift and demonstrates the good results on the model data.

Table 1. The experimental results: model dataset

Values of variations d and d'	Classification error of class -1, %	Classification error of class $+1$, %
$d \to 0; d' \to 0$	0	$\to 100$
$d = 1; d' = 1$	1.0092	3.1257
$d = 1 \cdot 10^{-8}$; $d' = 1 \cdot 10^{-8}$;	1,0801	2,846

4.2 Case Study: "Spam" E-Mail Problem

The object of the experimental study in this paper is the problem of filtering e-mails. The behavior of advertisement distributors is improving and as a result the spam filter should adapt to the behavior of spammers. Consequently, we come to problem of construction of the time-dependent decision rule.

We took the SPAM E-mail Database [13] from the repository UCI as the dataset. These data contain 4601 instances (e-mails) each of which is characterized 58 features. The last feature is nominal and describes the instance belonging to one of two classes: spam or non-spam. The values of other features are continuous and indicate the frequency of occurrence for particular elements (words or characters) in letter text or measure the length of sequences of consecutive capital letters. Junk e-mails are 39.4% (1813 instances) of all dataset.

To carry out the experiments there have been selected 3600 instances (4 groups of 900 instances each) from this database. The testing set was comprised by other 1001 instances. We standardized features before the experiments. To compare the obtained results we used some algorithms of concept drift realized in the software environment Massive Online Analysis (MOA) [14]:

– OzaBagASHT - bagging using the Adaptive-Size Hoeffding Trees [15]. When the tree size exceeds the maximun size value there restarts building of the tree from a new root. The main parameter of this algorithm is a number of models in the bag. For the choice of its optimal value we made some trial experiments on the SPAM E-mail Database (results are shown in the Table 2) and choose this value by the minimum of the classification error.

Table 2. Classification error depending on the number of models in the bag for OzaBagASHT

Number of models	6	8	10	12	14	16	18
Error, %	39,461	30,569	30,07	31,768	28,871	30,869	25,774

Number of models	20	22	**24**	26	28	30
Error, %	30,17	30,17	**22,278**	31,469	32,468	31,169

– OzaBagAdwin - bagging using ADWIN [15]. ADWIN is a change detector and estimator that solves the problem of tracking the average of a stream of bits or real-valued numbers in a well-specified way. The model in the bag is a decision tree for streaming data with adaptive Naive Bayes classification at leaves. As well as in the previous case we chose the optimal number of models after experiment set (results are shown in the Table 3) by the minimum of the classification error.
– SingleClassifierDrift - single classifier with drift detection method EDDM [16]. Decision tree for streaming data with adaptive Naive Bayes classification at leaves was selected as a classifier for training.
– AdaHoeffdingOptionTree - adaptive decision option tree for streaming data with adaptive Naive Bayes classification at leaves. We set the maximum

Table 3. Classification error depending on the number of models in the bag for OzaBagAdwin

Number of models	6	8	10	12	14	16
Error, %	22,278	23,278	22,877	23,776	26,773	22,977

Number of models	**18**	20	22	24	26
Error, %	**20,879**	22,078	22,677	22,278	24,177

number of options paths per node as 50 (the influence of this parameter on the error isn't detected).

- LimAttClassifier - the ensemble combining restricted hoeffding trees using stacking [17]. It produces a classification model based on the ensemble of restricted decision trees, where each tree is built from a distinct subset of the attributes. The overall model is formed by combining the log-odds of the predicted class probabilities of these trees using sigmoid perceptrons, with one perceptron per class. The minimum of the classification error was got under the following options: when one Adwin detects change, replaced the worst classifier; the number of attributes to use per model was 2.
- NormDistClassifierDrift - the method proposed by us in the previous part. The parameters $C = C_{-1} = C_1, d, d'$ were chosen after the trial tests with their different values by the minimum of error: $C = 1; d = 10^{-8}; d' = 10^{-8}$.

The final results given in the Table 4 show the proposed algorithm turned out to be a bit better on spam dataset than other algorithms.

Table 4. The experimental results: SPAM E-mail Database

Values of variations d and d'	Classification error, %
OzaBagASHT	22,278
OzaBagAdwin	20,879
SingleClassifierDrift	39.361
AdaHoeffdingOptionTree	23.876
LimAttClassifier	29,271
NormDistClassifierDrift	14,785

5 Conclusion

This paper proposed the strictly probabilistic basis for the problem of concept drift. This basis results from Bayesian approach to the pattern recognition problem. The method received from this approach used the general principle of the dynamic programming. On the "ground-truth" data this method demonstrates the required results. In the experiment on the spam e-mail database our method showed the acceptable error. It proves that the proposed method is applicable to the problem of concept drift for the pattern recognition problem.

References

1. Elwell, R., Polikar, R.: Incremental Learning of Concept Drift in Nonstationary Environments. IEEE Transactions on Neural Networks (2011)
2. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine Learning 23, 69–101 (1996)
3. Bifet, A., Gavalda, R.: Learning from time-changing data with adaptive windowing. In: SIAM International Conference on Data Mining (2007)
4. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: Getoor, L., Senator, T.E., Domingos, P., Faloutsos, C. (eds.) KDD, pp. 226–235. ACM Press (2003)
5. Street, W.N., Kim, Y.: A streaming ensemble algorithm (SEA) for large-scale classification. In: KDD, pp. 377–382. ACM Press (2001)
6. Bifet, A., Gama, J., Pechenizkiy, M., Zliobaite, I.: Handling Concept Drift: Importance, Challenges and Solutions. In: PAKDD-2011 Tutorial, Shenzhen, China, May 27 (2011)
7. Vapnik, V.: Statistical Learning Theory. John-Wiley & Sons, Inc. (1998)
8. Tatarchuk, A.I., Sulimova, V.V., Mottl, V.V., Windridge, D.: Method of relevant potential functions for selective combination of diverse information in the pattern recognition learning based on Bayesian approach. In: MMRO-14: Conf. Proc., Suzdal, pp. 188–191 (2009)
9. Wiener, N.: Extrapolation, Interpolation, and Smoothing of Stationary Random Time Series with Engineering Applications, 163p. Technology Press of MIT, John Wiley & Sons (1949)
10. Krasotkina, O.V., Mottl, V.V., Turkov, P.A.: Bayesian Approach to the Pattern Recognition Problem in Nonstationary Environment. In: Kuznetsov, S.O., Mandal, D.P., Kundu, M.K., Pal, S.K. (eds.) PReMI 2011. LNCS, vol. 6744, pp. 24–29. Springer, Heidelberg (2011)
11. De Groot, M.: Optimal Statistical Decisions. McGraw-Hill Book Company (1970)
12. Kostin, A.A., Kopylov, A.V., Mottl, V.V., Muchnik, I.B.: Dynamic Programming Procedures in Nonstationary Signal Analysis. Pattern Recognition and Image Analysis 11(1), 205–208 (2001)
13. Spambase Data Set, http://www.ics.uci.edu/~mlearn/MLRepository.html
14. Bifet, A., Holmes, G., Kirkby, R., Pfahringer, B.: MOA: Massive Online Analysis. Journal of Machine Learning Research, JMLR (2010), http://sourceforge.net/projects/moa-datastream/
15. Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., Gavalda, R.: New ensemble methods for evolving data streams. In: 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2009)
16. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: SBIA Brazilian Symposium on Artificial Intelligence, pp. 286–295 (2004)
17. Bifet, A., Frank, E., Holmes, G., Pfahringer, B.: Accurate Ensembles for Data Streams: Combining Restricted Hoeffding Trees using Stacking. Journal of Machine Learning Research Proceedings Track 13, 225–240 (2010)

Transductive Relational Classification in the Co-training Paradigm

Michelangelo Ceci[1], Annalisa Appice[1], Herna L. Viktor[2], Donato Malerba[1], Eric Paquet[2,3], and Hongyu Guo[3]

[1] Dipartimento di Informatica, Università degli Studi di Bari "A. Moro", Italy
{ceci,appice,malerba}@di.uniba.it
[2] School of Electrical Engineering and Computer Science, University of Ottawa
{hlviktor,epaquet}@site.uottawa.ca
[3] Institute for Information Technology, National Research Council of Canada
{eric.paquet,hongyu.guo}@nrc-cnrc.gc.ca

Abstract. Consider a multi-relational database, to be used for classification, that contains a large number of unlabeled data. It follows that the cost of labeling such data is prohibitive. Transductive learning, which learns from labeled as well as from unlabeled data already known at learning time, is highly suited to address this scenario. In this paper, we construct multi-views from a relational database, by considering different subsets of the tables as contained in a multi-relational database. These views are used to boost the classification of examples in a co-training schema. The automatically generated views allow us to overcome the independence problem that negatively affect the performance of co-training methods. Our experimental evaluation empirically shows that co-training is beneficial in the transductive learning setting when mining multi-relational data and that our approach works well with only a small amount of labeled data.

Keywords: Transductive learning, Co-training, Multi-relational classification.

1 Introduction

Increasingly, sets of structured data including relational databases, spatial data and biological data, amongst others, are available for mining. In these settings, using supervised learning methods often becomes impractical, due to the high costs associated with labeling the data. To this end, during recent years, there has been a growing interest in learning algorithms capable of utilizing both labeled and unlabeled data for prediction tasks. In the literature, two main settings have been proposed to exploit the information contained in both labeled and unlabeled data, namely the semi-supervised setting and the transductive paradigm. The former is based on the principle of inductive learning, where the learned function is used to make predictions, both on currently known and future observations. Transductive learning, on the other hand, makes predictions for the

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 11–25, 2012.

set of unlabeled data that is already known at learning time, which simplifies the learning task. Another strength of transductive learning is that it allows the examples in the training (and working) set to be mutually dependent.

Research further suggests that such structured (or semi-structured) data repositories often consist of subsets that provide us with different, complementary viewpoints of the dataset. That is, the data may be analyzed by exploring multiple complementary *views*, which each brings additional information regarding the problem domain. For example, during social network analysis, a contact may be viewed by considering: links to the person, textual content of the webpage, meta-data, multimedia content, and so on. In a relational database, these units of analysis may naturally be modeled as a set of tables $T_1, \ldots, T_n$, such that each table T_i describes a specific type of object involved in the units of analysis, while foreign key constraints explicitly model the relationships between objects. Multi-view learning, which concerns the selection of multiple independent views to be used for classification, is based on this observation.

When the number of unlabeled data is high, it follows that multi-view learning may be successfully combined with a transductive learning approach. To this end, we introduce the CoTReC (Co-training for Transductive Relational Classification) method, which is inspired by the above-mentioned principle of multi-view learning. Our method differs from the standard setting, since these views are mutually independent and are constructed from separate subsets of the dataset. In our approach, as a first step, we create a number of uncorrelated training and working views. Next, during co-training, the transductive learner proceeds to build a model and to predict the class value of the working set views as accurately as possible. Our approach requires only a small amount of labeled data, in order to construct accurate models.

The use of co-training is motivated by the need of improving the accuracy of weak transductive classifiers. This weakness is due to the low number of labeled examples that transductive algorithms are required to work with. In fact, with co-training, we augment the training set of a classifier learned from a database view by using predictions of unlabeled examples from the classifiers learned on the other views. This is performed with an iterative learning process [2].

This paper is organized as follows. Section 2 introduces the related work. This is followed, in Section 3 with a discussion of the CoTReC approach. Section 4 presents experiments, while Section 5 concludes this paper.

2 Related Work

This section introduces related research concerning transductive learning, multi-view learning and co-training.

2.1 Transductive Learning

Transductive learning has been investigated in classical data mining for SVMs ([8], [11]), for k-NN classifiers ([12]) and even for general classifiers ([16]).

However, more recently, transductive learning has been also investigated in multi-relational data mining, which aims to discover useful patterns across multiple relations (tables) in a relational database [9]. Krogel and Scheffer ([15]) investigated a transformation (known as propositionalization) of a relational description of gene interaction data into a classical double-entry table, and then study transduction with the well-known transductive support vector machines. Taskar et al. ([25]) built a generative probabilistic model that captures interactions between examples, some of which have class labels, while others do not. However, given sufficient data, a discriminative model generally provides significant improvements in classification accuracy over generative models. Malerba et al. [19] proposed a relational classification algorithm that works in a transductive setting and employs a probabilistic classification approach. The transductive learning strategy iterates on a k-NN based re-classification of labeled and unlabeled examples, in order to identify borderline examples, and uses the relational probabilistic classifier Mr-SBC to bootstrap the transductive algorithm.

2.2 Multi-view Learning

Multi-view learning concerns the learning from multiple independent sets of features, i.e. views, of the data. This framework has been applied to real-world applications such as information extraction [2], face recognition [18] and locating users in a WiFi environment [23], amongst others. In essence, multi-view learning utilizes experiential inputs to explore diverse representations so that an increasingly variety of problems may be recognized, formulated and solved. In this way, the strengths of each view are amplified and the weaknesses are alleviated. This allows for cooperation, knowledge sharing and knowledge fusion.

Formally, a multi-view problem with n views may be seen as n uncorrelated or weakly dependent features sets [9]. Here, the correlation estimates the departure of two variables (views) from independence. If we consider two views as being correlated, it implies that knowing the first view helps in predicting the second. On the other hand, if knowing the first brings no (or little) knowledge about the second, the correlation coefficient should be near to zero.

Intuitively, multi-view learning is well suited for relational data mining. Within the relational database context, it follows that such sets of disjoint features are typically contained in multiple relations. That is, a relational database schema, as designed by a domain expert, usually groups attributes into relations (or entities in an extended entity-relationship (EER) diagram) with very close semantic meaning. Interested readers are referred to [9] for a detailed discussion of a multi-view learning methodology within the relational database setting.

2.3 Co-training

The iterative co-training framework was originally introduced in [2]. The idea is to split the set of predictor attributes $\mathbf{X}$ into two disjoint subsets $\mathbf{X1}$ and $\mathbf{X2}$. Each labeled example (x, y) is viewed as $(x1, x2, y)$ where $x1$ contains the values of the attributes in $\mathbf{X1}$ and $x2$ the values of the attributes in $\mathbf{X2}$. Then, two

classifiers $f_1(\cdot)$ and $f_2(\cdot)$ (where $f_1 : \mathbf{X1} \rightarrow Y$ and $f_2 : \mathbf{X2} \rightarrow Y$) are learned by bootstrapping one another with labels for the unlabeled data. Intuitively, the initial classifiers f_1 and f_2 are learned over a small sample and the iterative bootstrap takes unlabeled examples $(x1, x2)$ for which f_1 is confident, but f_2 is not (or vice-versa) and using the reliable classification to label such examples for the learning algorithm on f_2 (or f_1), while improving the other classifier.

Two assumptions are made in co-training: *i)* (*Compatibility* assumption) The example distribution is compatible with the classifier $f : \mathbf{X} \rightarrow Y$. That is, either attribute set suffices to learn the target $f(\cdot)$ such that for each example x: $f_1(x) = f_2(x) = f(x)$. *ii)* (*Conditionally independence* assumption) The predictor attributes in **X1** and the predictor attributes in **X2** are conditionally independent given the class label. Formally, $P(x1|f(x), x2) = P(x1|f(x))$ and $P(x2|f(x), x1) = P(x2|f(x))$. While assumption *ii)* justifies an iterative approach, assumption *i)* can be significantly weakened while still permitting the use of unlabelled data to iteratively boost weak classifiers. Accordingly, several studies [21] [14] have shown that co-training may improve classifier performance even when the assumptions are violated to some extent.

The iterative co-training approach has been successfully applied to several learning problems, including named entity classification [5], text classification [22] and image classification [17]. Most of these works extract the multiple views by trying to meet empirically the co-training assumptions (or weakening of them) on the training domain. However, unlike the work presented in this paper, these techniques do not address the problem of extracting multiple views from data spanned in multiple tables of a relational database by trying to meet the compatibility and conditionally independence assumptions.

3 The CoTReC Method

Let D be a set of units of analysis (examples) whose description is spread in multiple tables of a relational database. Examples can be labeled according to an unknown target function whose range is a finite set $Y = \{C_1, \ldots, C_L\}$. The transductive classification problem is formalized as follows: *Given* a training set $TS \subset D$, and a working set $WS = D - TS$ with unknown target values. The problem is to *predict* the class value of each example in the working set WS which is as accurate as possible. The learner receives full information (including labels) for the examples in TS and partial information (without labels) for the examples in WS and is required to predict the class only of the examples in WS.

The use of the co-training framework to solve this problem is illustrated in Algorithm 1. The algorithm takes as input n training and working views[1] and returns the set of labeled working examples. In the while loop, the algorithm iterates by bootstrapping the data labeling on the different views. In particular, it classifies examples in each view and identifies the examples for which the classification is reliable. These examples are added to the other training views (see

[1] Training and Working views are defined according to the same schema, but are used on the training and working database, respectively.

Algorithm 1, lines 16-18) for subsequent iterations. The associated label is identified according to the Bayesian Optimal Classifier [20] that permits to combine the effect of (possibly multiple) reliable classifications of the same example (see Algorithm 1, lines 13-14). Formally, it is defined as:

$$BOC(e, Y'', h_1, \ldots, h_n, a_1, \ldots, a_n) = \underset{c \in Y''}{\operatorname{argmax}} \sum_{j=1,\ldots,n} P(c|h_j) \times a_j \quad (1)$$

where $Y'' \subseteq Y$ is a subset of the set of labels, $P(c|h_j)$ is equal to one if the classifier h_j associates the example e with the label c, 0 otherwise and a_j estimates the probability $P(h_j|A_j)$ and is computed as the accuracy of the classifier h_j on the training view A_j (see Algorithm 1, line 9). The classifier h_j may utilize one of many available propositional classification algorithms.

It is noteworthy that decisions on the classifications are propagated coherently through the training views (and added to L). Once a stopping criterion is satisfied, the Bayesian Optimal Classifier is used in order to classify the remaining unlabeled examples according to the classifiers learned on training views obtained after the last iteration (see Algorithm 1, lines 25-38).

Three stopping criteria are considered in the algorithm, that is, all the examples have been classified; no example is added to the training views during the last iteration; a maximum number of iterations (MAX_ITERS) is reached.

Two issues remain to be discussed: *i)* how views are constructed and *ii)* how the reliable classifications are identified. These issues are discussed in the next.

3.1 Constructing Multi-views from Relational Data

In a relational setting, there is a database $\Re$, which consists of a target table T_{target} and a set of background relations $\{T_i\}_1^n$. The target relation T_{target} has a target variable Y. That is, each tuple in this table (target tuple) is associated with a class label which belongs to Y. Typically, the relational classification task is to find a function $F(x)$ which maps each target tuple x to the category set Y:

$$Y = F(x, T_{1\cdots n}, T_{target}), x \in T_{target} \quad (2)$$

This relational classification task may be mapped to a multi-view learning problem, as follows. Join paths link the target relation with the other relations in the relational database, through following foreign key links. That is, foreign key attributes link to primary keys of other tables: this link specifies a *join* between two tables, and a foreign key attribute of table T_2 referencing table T_1 is denoted as $T_2.T_{1\_key}$. Each one of these paths provides a complementary, potentially information-rich view of the target concept to be learned.

The next section discusses the multi-view construction process which corresponds to the first four steps of the MRC multi-view learner reported in [9].

Multi-View Construction. Algorithm 2 highlights the steps we following when constructing the multiple views. As shown in Algorithm 2, the method initially propagates and aggregates information to form multiple views from a

Algorithm 1. CoTReC

Require: $A_1, \ldots, A_n$ training views, $W_1, \ldots, W_n$ working views.
Ensure: L working examples associated with labels

$L \leftarrow \emptyset$;
while no stopping criterion is satisfied **do**
  for all $i = 1, \ldots, n$ **do**
    Train the classifier h_i on the training view A_i;
    Classify all the examples in W_i according to the classifier h_i;
    Compute the ranked of the class predicted by h_i for each example in W_i;
    Sort the examples in W_i by reliability in decreasing order;
    $T_i \leftarrow$ the reliability threshold for W_i;
    $a_i \leftarrow$ accuracy of the classifier h_i on the training view A_i;
  end for
  for all $i = 1, \ldots, n$ **do**
    for all $e \in W_i$ with reliability greater than T_i **do**
      $Y' \leftarrow \{c \in Y |\ \exists j \in [1, \ldots, n],\ h_j(e) = c$ with reliability greater than $T_j\}$;
      $label \leftarrow BOC(e, Y', h_1, \ldots, h_n, a_1, \ldots, a_n)$
      for all $j = 1, \ldots, n$ **do**
        if $i \neq j$ **then**
          $A_j = A_j \cup \{(e, label)\}$;
        end if
        $W_j = W_j - \{e\}$;
      end for
      $L = L \cup \{(e, label)\}$;
    end for
  end for
end while
for all $i = 1, \ldots, n$ **do**
  Train the classifier h_i on the training set A_i;
  Classify all the examples in W_i according to the classifier h_i;
  $a_i \leftarrow$ accuracy of the classifier h_i on the training view A_i;
end for
for all $i = 1, \ldots, n$ **do**
  for all $e \in W_i$ **do**
    $label \leftarrow BOC(e, Y, h_1, \ldots, h_n, a_1, \ldots, a_n)$
    for all $j = 1, \ldots, n$ **do**
      $W_j = W_j - \{e\}$;
    end for
    $L = L \cup \{(e, label)\}$;
  end for
end for
RETURN L

relational database. Subsequently, the view validation step identifies a subset of uncorrelated views. This set of views may subsequently be used in a transductive learning setting, to co-train classifiers.

1) Information Propagation Stage: The Information Propagation Stage constructs training data sets for use by a number of view learners. The original

Algorithm 2. Multi-view Construction Algorithm

Require: A database $\Re = \{T_{target}, T_1, T_2, \cdots, T_n\}$, a view learner $\mathcal{L}$, a meta learner $\mathcal{M}$, and a view validation data $\mathcal{T}_v$ from $\Re$.

1: Propagate and aggregate information in $\Re$, forming candidate view set $\{V_d^1, \cdots, V_d^n\}$;
2: Remove candidate views with view learners' accuracy less than 50% from $\{V_d^1, \cdots, V_d^m\}$, forming view set $V^{'}$;
3: Generate a validation data set $\mathcal{T}_v^{'}$, using $\mathcal{T}_v$ and $V^{'}$;
4: Select a view feature set $\mathcal{A}^{'}$ from $\mathcal{T}_v^{'}$;
5: Let validated view set $V = \emptyset$;
6: **for each** view V^i ($V^i \in V^{'}$) which has at least one attribute in $\mathcal{A}^{'}$ **do**
7: V.add(V^i);
8: **end for**
9: RETURN V.

relational database is used as input and the process starts from the target table. Our approach was inspired by the so-called *tuple id* propagation, as introduced in CrossMine [26], which is a technique for performing virtual joins between tables. The target and background relations are linked together through their foreign keys, which are identified by means of the *tuple identifiers*. In SQL, this is done by an equi-join of the tables over the values of the key identifiers.

2) Aggregation Stage: The Aggregation Stage summarizes the information embedded in the tuples associated with one to many relationships (i.e., one target tuple is associated with multiple entities in a background relation) by converting them into one row. This procedure is applied to each of the data sets constructed during the Information Propagation Stage. To achieve this objective, aggregation functions are applied. By applying the basic aggregation functions in SQL, new features are created to summarize information stored in the multiple tuples. For nominal attributes, we employ the *count* function. For numeric values, the *count, sum, average, maximum, minimum and standard deviation* are calculated.

3) Multiple Views Construction Stage: In the third phase of the algorithm, the Multiple Views Construction Stage constructs various hypotheses on the target concept, based on the multiple training data sets given by the Aggregation Stage. To this end, view learners such as decision trees or support vector machines (SVMs) are used in order to learn the target concept from each view of the database separately. In this stage, a number of view learners are trained.

4) View Validation Stage: All view learners constructed in the Multiple Views Construction Stage are then evaluated in the View Validation Stage. This processing is needed to ensure that they are, indeed, able to learn the target concept. In addition, strongly uncorrelated view learners are preferred. The aim of this stage is to maintain only views that are able to accurately predict the class.

As a first step, we discard all views that perform worse than random guessing. We eliminate all views with an error rate lower than 50%. Research suggests that disjoint views are preferred by multi-view learning, due to the inherent philosophy of diverse viewpoints [6,5,9]. Following this line of thought, we employ

a *Correlation-based View Validation (CVV)* algorithm that uses the heuristic goodness, from test theory, as utilized by the CFS subset feature selection approach of Hall [10]. The heuristic 'goodness' of a subset of features [10] is:

$$\mathcal{C} = K\overline{R_{cf}}/\sqrt{K + K(K-1)\overline{R_{ff}}} \tag{3}$$

where $\mathcal{C}$ is the heuristic 'goodness' of a selected feature subset. K is the number of features in the selected subset. $\overline{R_{cf}}$ calculates the average feature-to-class correlation, and $\overline{R_{ff}}$ stands for the average feature-to-feature dependence.

We adopt the *Symmetrical Uncertainty* ($\mathcal{U}$) [9] to calculate $\overline{R_{cf}}$ and $\overline{R_{ff}}$. This measure is a modified version of the *information gain* measure which compensates for information gain's bias toward attributes with more values.

The *Symmetrical uncertainty* is defined as follows:
Given features X and Y,

$$\mathcal{U} = 2.0 \times \left[\frac{InfoGain}{H(Y) + H(X)}\right] \quad \text{where } H(Y) = -\sum_{y \in Y} p(y)\log(p(y)).$$

In summary, during view validation, each view is called upon to give a prediction of the target class, against a validation data set. Different subsets of the views are then ranked, according to their C values and the subset with the highest rankings is kept. These views are then used as the input to CoTReC.

3.2 Reliability Measures

In order to identify reliable classifications, it is necessary to use measures that quantify how reliable the classification is. Indeed, many classification algorithms return confidence measures (see, for example naïve Bayes, k-NN and SVM classifiers). However, such measures are algorithm-dependent and cannot be used in a general framework. Several algorithm-independent reliability measures have been proposed in the literature. In [4], twelve reliability measures are compared by concluding that no measure, independently considered, can be robust enough in measuring the reliability of any class in any domain. On the basis of this consideration, authors proposed to learn a decision tree which aggregates basic measures by obtaining a robust piecewise reliability measure. Delany et al. [7], on the other hand, propose three reliability measures that are domain-independent, monotonic in terms of the reliability and do not require a training phase.

In this paper, we employ the results of both these studies. In particular, from [4], we use the idea of combining several reliability measures, while from [7] we identify the algorithm and domain-independent measures to be combined. Recall that we aim to evaluate the reliability of the label predicted for an example e, by considering labels associated to examples in the k-nearest neighborhood.

As basic reliability measures, we select the following three measures:

$$AvgNUNI(e,k) = \sum_{i=1}^{k} IndexOfNUN_i(e) \tag{4}$$

$$SR(e,k) = \frac{\sum_{i=1}^{k} sim(e, NLN_i(e))}{\sum_{i=1}^{k} sim(e, NUN_i(e))} \tag{5}$$

$$SRK(e,k) = \frac{\sum_{i=1}^{k} sim(t, NN_i(e)) \cdot 1(t, NN_i(e))}{\sum_{i=1}^{k} sim(t, NN_i(e)) \cdot (1 - 1(t, NN_i(e)))} \tag{6}$$

where:

- $NN_i(e)$: the i-th nearest neighbor of e,
- $NLN_i(e)$: the i-th nearest like (same label) neighbor of e,
- $NUN_i(e)$: the i-th nearest unlike (different label) neighbor of e,
- $IndexOfNUN_i(e)$ is the index of $NN_i(e)$. The index is the ordinal ranking of the example in the list of nearest neighbors,
- $sim(a,b)$ is the similarity between examples a and b,
- $1(a,b) = 1$ if $label(a) = label(b)$; 0 otherwise .

These measures are used as follows. $AvgNUNI(e,k)$ measures how close e is to the first k nearest unlike neighbors (NUNs); $SR(e,k)$ measures the ratio of the similarity between e and its first k nearest like neighbors (NLNs) and the similarity between e and its first k NUNs; $SRK(e,k)$ is similar to $SR(\cdot,\cdot)$ except that, it considers only the first k examples, independent from the class label.

The similarity function $sim(a,b)$ has range in $[0,1]$ and is computed in a standard way. In particular, we use Manhattan distance for continuous predictor attributes and the Hamming distance for discrete ones. More formally:

$$sim(a,b) = 1 - \frac{m^{(C)} d^{(C)}(a,b) + m^{(D)} d^{(D)}(a,b)}{m^{(C)} + m^{(D)}} \tag{7}$$

where $m^{(C)}$ ($m^{(D)}$) is the number of continuous (discrete) predictor attributes in the view and $d^{(C)}(a,b)$ ($d^{(D)}(a,b)$) is the Euclidean or the Manhattan (Hamming) distance computed on continuous (discrete) attributes. Each continuous attribute is normalized through a linear scaling into $[0,1]$. This guarantees that $sim(a,b)$ belongs to $[0,1]$ and attributes uniformly contribute to $sim(a,b)$.

The aggregated reliability measure $ARM(e,k)$ is computed by averaging normalized values of $AvgNUNI(e,k)$, $SR(e,k)$ and $SRK(e,k)$, as suggested in [4]. Again, the normalization is performed on the interval $[0,1]$:

$$ARM(e,k) = \frac{1}{3}(AvgNUNI'(e,k) + SR'(e,k) + SRK'(e,k)) \tag{8}$$

where $AvgNUNI'(e,k)$, $SR'(e,k)$ and $SRK'(e,k)$ are the normalized versions of $AvgNUNI(e,k)$, $SR(e,k)$ and $SRK(e,k)$, respectively. With a single reliability value we can rank confidences in decreasing order and then select the top examples for which $ARM(e,k)$ is greater than the reliability threshold T_i. This threshold is empirically identified at each iteration, for each view.

3.3 Computing Reliability Thresholds

In the CoTReC algorithm, we use the $ARM(e,k)$ computed for each training example to compute reliability thresholds. We identify the reliability threshold that allows us to maximize the AUC (Area under the ROC curve) of a threshold-based classifier G_θ that solves the following two classes classification problem: $(+)$ the label of a training example is correctly predicted according to h_i; $(-)$ the label of a training example is incorrectly predicted according to h_i-

Algorithmically, for each candidate threshold θ, the AUC of the classifier G_θ that classifies as $(+)$ examples in A_i for which $ACM(e,k) \geq \theta$ and classifies as $(-)$ examples in A_i for which $ACM(e,k) < \theta$ is computed. The set of candidate thresholds Θ_i is computed by ranking examples in A_i according to $ACM(e,k)$ and by considering the middle values of reliabilities of two consecutive examples.

Therefore, each threshold T_i is computed as:

$$T_i = \underset{\theta \in \Theta}{\operatorname{argmax}} \; AUC(G_\theta, A_i) \tag{9}$$

The advantages of this solution are: *i)* thresholds are local to each view for each iteration and *ii)* thresholds are computed on the current training set.

3.4 Implementation Considerations

With a naïve implementation, at each iteration, for each view, it is necessary to compute the similarities between the training examples and all the examples. These similarities are used both for computing reliability thresholds and deciding which working examples can be considered as reliably classified. The cost of computing similarities, in the worst case, is $O(l_A \times (l_A + l_W) \times MAX\_ITERS)$, where l_A is the total number of training examples and l_W is the total number of working examples. In our implementation, we do not recompute the same similarities from one iteration to the next. Moreover, we organize examples according to an *r-tree* structure in order to compute similarities between close examples in logarithmic time. In this way, at the first iteration, we compute similarities with a time complexity $O(log(l_A) \times log(l_A + l_W))$. From the second iteration, we only compute similarities between examples kept in the working set and examples lastly moved in the training set.

4 Experiments

The empirical evaluation of our algorithm was carried out on three real world datasets: PKDD 99 discovery challenge database, Mutagenesis database (which have been both used to test several MRDM algorithms), and North West England (NWE) Census Database. The CoTReC algorithm is evaluated on the basis of the average accuracy on the same 10-fold cross-validation (10-CV) against each database. For each database, the target table is first divided into 10 blocks of nearly equal size and then a subset of tuples related to the tuples of the target

table block is extracted by means of foreign key constraints. In this way, 10 database instances are created. For each trial, CoTReC is trained on a single database instance and tested on the hold-out nine database instances, forming the working set. It should be noted that the accuracies reported in this work are thus not directly comparable to those reported in other research, which uses a standard 10-fold CV method. This is due to our unusual experimental design, imposed by the transductive co-training paradigm. Indeed, unlike the standard CV approach, here one fold at a time is set aside to be used as the training set (and not as the test set). Small training set sizes allow us to validate the transductive approach, but may result in higher error rates as well.

4.1 Datasets

The first database that we consider is the PKDD 99 discovery challenge database [1], as introduced earlier. Recall that this database concerns the task of determining whether a client will default on his loan. The database contains eight tables. The target Loan table consists of 682 records, including a class attribute status which indicates the status of the loan, i.e. A (finished and good), B (finished but bad), C (good but not finished), or D (bad and not finished). The background information for each loan is stored in the relations *Account, Client, Order, Transaction, Credit Card, Disposition* and *Demographic*. All background relations relate to the target table through directed or undirected foreign key chains. The aim of the learning problem is to classify if a loan is at risk or not.

The Mutagenesis database is related to the problem of identifying some mutagenic compounds [24]. We have considered, similarly to most experiments on relational data mining algorithms, the "regression friendly" dataset consisting of 188 molecules. It consists of data obtained with the molecular modeling package QUANTA. For each compound, it contains the atoms, bonds, bonds types, atom types, and partial charges on atoms plus indicators ind1, inda, logp, and lumo.

The NWE Census database is obtained from both census and digital map data provided by the European project SPIN!. These data concern Greater Manchester, one of the five counties of NWE. Greater Manchester is divided into 214 censual wards. Mortality rate as well as some indexes of the deprivation (Jarman Underprivileged Area Score, Townsend Index, Carstairs Index and Department of the Environment Index) are available at ward level. The goal of the classification task is to predict the value of the Jarman index (low or high) deprivation factor by exploiting the other deprivation factors, mortality rate and geographical factors, represented in some linked topographic maps. Spatial analysis is possible thanks to the availability of vectorized boundaries of the 1998 census wards as well as of other Ordnance Survey digital maps of NWE where urban area (115 spatial objects), green area (9), road net (1687), rail net (805) and water net (716) can be found. Topological non-disjoint relationships between wards and objects in all these layers are materialized as relational tables.

Table 1. Average number of views extracted and validated from MV.

DB	PKDD 99	Mutagensis	NWE
No of views	2	2	6

4.2 Experimental Results

In the experiments, we have considered four learning systems as base classifiers. Specifically, C4.5, the RIPPER rule learner, Naïve Bayes (NB), SMO classifier [13] and 1-nearest neighbours (1-NN) which are all implemented in the Weka system. CoTReC is run by varying k (see (8)) among three values expressed as a percentage (1%,2% and 3%) of the number of the examples; MAX_ITERS is set to the number of unlabeled examples stored in the working database. In Table 1, we report the average number of views extracted within CoTreC. Each view provides a propositionalization of the corresponding database.

In Table 2, we report both the 10-CV average accuracies and the average numbers of iterations for PKDD 99 discovery challenge database and Mutagenesis database. We have chosen these databases, since they have been used to benchmark several state-of-the-art muilti-relational data mining methods. Results, as shown in Table 2, indicate that there is no distance which consistently leads to a better classification accuracy. The accuracy of classification generally improves by increasing the k value. Moreover, the classification accuracy remains high (more than 87.5%) on PKDD 99 discovery challenge database, independently of the base classifier. However, the best accuracy is obtained by using NB. This trend is confirmed when analyzing the Mutagenesis database, where we observe an higher variability of the accuracy by varying the basic classifier. For Mutagenesis, we observe that the best accuracies are obtained with the SMO classifier which, however, is less stable than NB when varying k. By considering the average number of iterations, we observe that it is generally low, except for the case of C4.5 . Here, CoTReC tends to move a small number of examples in the training set at each iteration, since the trees do not significantly change from one iteration to the next one. On the basis of these considerations, NB can be reasonably considered as the classifier that better interacts with CoTReC.

To evaluate the predictive performance of the CoTReC algorithm, we compared our method with the only relational transductive classifier currently available in the literature, namely the TRANSC approach [19]. TRANSC iterates on a k-NN based re-classification of labeled and unlabeled examples, in order to identify class borderline examples, and uses the relational probabilistic classifier Mr-SBC [3] to bootstrap the transductive algorithm. Although similar to CoTReC, TRANSC does not use co-training and does not exploit multi-views. The experimental results reported in [19] include the predictive performance of the TRANSC method as well as the relational naïve Bayesian classifier Mr-SBC [3] against the Mutagenesis and NWE databases. We ran CoTReC against these two databases, using the same CV partitioning as reported in [3]. We compared the accuracy obtained by CoTReC with the best results of the TRANSC and Mr-SBC methods against these two databases, as presented in [3]. We depict

Table 2. PKDD 99 discovery challenge database and Mutagenesis database: Average accuracies/iterations obtained with different k values. Column "Distance Measure" refers to the used distance measure for continuous attributes. The average number of iterations is approximated to the unity.

Learner	Distan.	PKDD 99 discovery challenge			Mutagenesis		
	Measu.	k=1%	k=3%	k=5%	k=1%	k=3%	k=5%
C4.5	Euclid.	87.94% /74	87.94% /124	87.94% /137	50.88% /6	77.51% /4	82.84% /6
	Manha.	87.94% /129	87.94% /66	87.94% /111	53.25% /5	73.96% /5	80.47% /4
RIPPER	Euclid.	87.94% /4	87.94% /4	87.94% /1	37.86% /4	79.88% /3	82.24% /2
	Manha.	87.94% /1	87.94% /3	87.94% /4	43.78% /5	78.69% /3	82.24% /3
NB	Euclid.	88.11% /2	88.11% /2	88.11% /2	69.23% /5	84.02% /3	74.55% /5
	Manha.	88.11% /2	88.11% /2	88.11% /2	68.08% /3	87.57% /4	85.79% /6
SMO	Euclid.	87.94% /9	87.94% /5	87.94% /6	60.94% /5	78.69% /3	88.16% /3
	Manha.	87.94% /10	87.94% /15	87.94% /6	39.64% /7	78.69% /3	88.75% /2
1-NN	Euclid.	87.62% /1	87.62% /1	87.62% /1	78.69% /1	78.69% /1	78.69% /1
	Manha.	87.62% /1	87.62% /1	87.62% /1	78.69% /1	78.69% /1	78.69% /1

Table 3. Mutagenesis and NWE databases: average accuracies obtained with CoTReC, TRANSC and Mr-SBC. CoTReC is run with NB learner as basic classifier, k=5% and Manhattan distance.

System/DB	Mutagenesis	NWE
CoTReC	85.79%	83.33%
TRANSC	82.89%	81.96%
Mr-SBC	75.08%	77.29%

the comparison results in Table 3. Results indicate that the CoTReC method outperformed both the TRANSC and the Mr-SBC algorithms for the two tested databases, in terms of the obtained accuracies. For example, when contrasting CoTReC with the Mr-SBC method, our algorithm improved the accuracy against the Mutagenesis and NWE databases by 10.71% and 6.04%, respectively. Compared to the TRANSC strategy, the CoTReC algorithm was able to reduce the predictive error against the Mutagenesis and NWE databases by 2.90% and 1.37%, respectively. This reduction is mainly due to the co-training approach.

In summary, these results show that, with only a small number of labeled data, CoTReC can successfully construct accurate classification models, through benefiting from the use of co-training and transductive learning paradigms.

5 Conclusions

Numerous real-world applications contain complex and heterogeneous data, which are naturally modeled as several tables in a relational database. Often, such data repositories consist of a small number of labeled data together with a set of unlabeled data. In this paper, we have investigated the combination of transductive inference with co-training for the classification task in order to successfully mine such data. Our method exploits multi-views extracted from a relational database in a co-training schema. Multi-views are extracted by following the foreign key chains from relational databases and these views enable us

to deal with the relational structure of the data. Co-training allows us to boost the classification of examples within an iterative learning process.

The proposed classifier (CoTReC) has been compared to the TRANSC transductive relational classifier and the Mr-SBC inductive relational classifier. Results show that CoTReC outperforms both systems. As future work, we intend to extend the empirical investigation to corroborate our intuition that transductive inference has benefits from co-training when applied to multi-views extracted from a relational database. Our work will further include a thorough investigation of the robustness and efficiency of CoTReC against further databases. In addition, experimental evaluation on the influence of the different confident measures on the CoTReC approach will be conducted. We also plan to compare our method with other state-of-the-art co-training and multi-relational learning strategies. It would also be interesting to extend the CoTReC algorithm to other relational data such as spatial data, deep-web data and social network data.

Acknowledgment. This work is supported in fulfillment of the research objectives of the PRIN 2009 Project "Learning Techniques in Relational Domains and Their Applications" funded by the Italian Ministry of University and Research (MIUR).

References

1. Berka, P.: Guide to the financial data set. In: Siebes, A., Berka, P. (eds.) PKDD 2000 Discovery Challenge (2000)
2. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Workshop on Computational Learning Theory (1998)
3. Ceci, M., Appice, A., Malerba, D.: Mr-SBC: A Multi-relational Naïve Bayes Classifier. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 95–106. Springer, Heidelberg (2003)
4. Cheetham, W., Price, J.: Measures of Solution Accuracy in Case-Based Reasoning Systems. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 106–118. Springer, Heidelberg (2004)
5. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 100–110 (1999)
6. Dasgupta, S., Littman, M.L., McAllester, D.A.: PAC generalization bounds for co-training. In: NIPS, pp. 375–382 (2001)
7. Delany, S.J., Cunningham, P., Doyle, D., Zamolotskikh, A.: Generating Estimates of Classification Confidence for a Case-Based Spam Filter. In: Muñoz-Ávila, H., Ricci, F. (eds.) ICCBR 2005. LNCS (LNAI), vol. 3620, pp. 177–190. Springer, Heidelberg (2005)
8. Gammerman, A., Azoury, K., Vapnik, V.: Learning by transduction. In: Proc. of the 14th Annual Conference on Uncertainty in Artificial Intelligence, UAI 1998, pp. 148–155. Morgan Kaufmann (1998)
9. Guo, H., Viktor, H.L.: Multirelational classification: A multiple view approach. Knowledge and Information Systems: An International Journal 17, 287–312 (2008)

10. Hall, M.: Correlation-based feature selection for machine learning, Ph.D diss., Waikato Uni. (1998)
11. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proc. of the 16th International Conference on Machine Learning, ICML 1999, pp. 200–209. Morgan Kaufmann (1999)
12. Joachims, T.: Transductive learning via spectral graph partitioning. In: Proc. of the 20th International Conference on Machine Learning, ICML 2003 (2003)
13. Keerthi, S.S., Shevade, S.K., Bhattacharyya, C., Murthy, K.R.K.: Improvements to platt's smo algorithm for svm classifier design. Neural Computation 13(3), 637–649 (2001)
14. Kiritchenko, S., Matwin, S.: Email classification with co-training. In: Proceedings of the 2001 Conference of the Centre for Advanced Studies on Collaborative Research, CASCON 2001, p. 8. IBM Press (2001)
15. Krogel, M.-A., Scheffer, T.: Multi-relational learning, text mining, and semi-supervised learning for functional genomics. Machine Learning 57(1-2), 61–81 (2004)
16. Kukar, M., Kononenko, I.: Reliable Classifications with Machine Learning. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) ECML 2002. LNCS (LNAI), vol. 2430, pp. 219–231. Springer, Heidelberg (2002)
17. Levin, A., Viola, P., Freund, Y.: Unsupervised improvement of visual detectors using co-training. In: Proceedings of the Ninth IEEE International Conference on Computer Vision, ICCV 2003, Washington, DC, USA, vol. 2, pp. 626–637. IEEE Computer Society (2003)
18. Li, S.Z., Zhu, L., Zhang, Z., Blake, A., Zhang, H., Shum, H.-Y.: Statistical Learning of Multi-view Face Detection. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 67–81. Springer, Heidelberg (2002)
19. Malerba, D., Ceci, M., Appice, A.: A relational approach to probabilistic classification in a transductive setting. Eng. Appl. Artif. Intell. 22, 109–116 (2009)
20. Mitchell, T.: Machine Learning. McGraw-Hill, New York (1997)
21. Muslea, I., Minton, S., Knoblock, C.A.: Active + semi-supervised learning = robust multi-view learning. In: Proceedings of the Nineteenth International Conference on Machine Learning, ICML 2002, pp. 435–442. Morgan Kaufmann Publishers Inc., San Francisco (2002)
22. Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of co-training. In: CIKM, pp. 86–93. ACM (2000)
23. Pan, S., Kwok, J., Yang, Q., Pan, J.: Adaptive localization in a dynamic wifi environment through multi-view learning. In: AAAI 2007, Menlo Park, CA, pp. 1108–1113 (2007)
24. Srinivasan, A., Muggleton, S., King, R.D., Sternberg, M.J.E.: Mutagenesis: Ilp experiments in a non-determinate biological domain. In: Wrobel, S. (ed.) Proc. of the 4th Inductive Logic Programming Workshop, pp. 217–232. GMD-Studien (1994)
25. Taskar, B., Segal, E., Koller, D.: Probabilistic classification and clustering in relational data. In: Nebel, B. (ed.) IJCAI, pp. 870–878. Morgan Kaufmann (2001)
26. Yin, X., Han, J., Yang, J., Yu, P.S.: Crossmine: Efficient classification across multiple database relations. In: ICDE 2004, Boston, pp. 399–410 (2004)

Generalized Nonlinear Classification Model Based on Cross-Oriented Choquet Integral

Rong Yang[1,*] and Zhenyuan Wang[2]

[1] College of Mechatronics and Control Engineering, Shen Zhen University, Shen Zhen, China
ryang@szu.edu.cn
[2] Department of Mathematics, University of Nebraska at Omaha, USA
zhenyuanwang@unomaha.edu

Abstract. A generalized nonlinear classification model based on cross-oriented Choquet integrals is presented. A couple of Choquet integrals are used in this model to achieve the classification boundaries which can classify data in such situation as one class surrounding another one in a high dimensional space. The values of unknown parameters in the generalized model are optimally determined by a genetic algorithm based on a given training data set. Both artificial experiments and real case studies show that this generalized nonlinear classifier based on cross-oriented Choquet integrals improves and extends the functionality of traditional classifier based on one Choquet integral on solving the classification problems of multi-class multi-dimensional situations.

Keywords: classification, Choquet integral, signed efficiency measure, genetic algorithm, optimization.

1 Introduction

The Choquet Integral [1, 2] with respect to a fuzzy measure or signed fuzzy measure [3, 4], also called efficiency measure or signed efficiency measure respectively in literature, has been performed successfully as a nonlinear aggregation tool [5]. The nonadditivity of the signed fuzzy measure provides an effective representation to describe the interaction among the contributions from the predictive attributes to the objective attribute. These properties endow the Choquet integral with the ability of solving data classification problems. Typical works can be found in [5-7], where encouraging results have been presented.

However, the applicability of the classification model using only one Choquet integral is limited. It cannot solve the classification problem such as one data class surrounding with another data class. In [6], we proposed a new nonlinear classification model based on cross-oriented Choquet integrals, where two Choquet integrals are used to construct a quadrilateral boundary for 2-dimensional 2-class classification problems. In that work, regularization is operated on the original data such that the projection axis

* Corresponding author.

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 26–39, 2012.

passes through the origin, and a rotation transformation is operated to adjust the slope of the projection axis. This model can solve the classification problems of 2-dimensional data sets successfully. But when the problems are about 3-dimensional or high dimensional data, it is difficult to get the explicit expression of rotation transformation. So, in this work, a generalized nonlinear classification model based on cross-oriented Choquet integral is proposed. It has the ability of solving the high dimensional classification problems of multiple classes. Experiments both on artificial and real data sets have been performed and partially satisfactory results are achieved.

This paper is organized as follows. In Section 2, basic mathematical knowledge for constructing the generalized classification model is given. Section 3 and Section 4 present our generalized nonlinear classification model based on cross-oriented Choquet integrals and its relevant optimization algorithm. In Section 5, artificial data and real data from UCI Machine Learning repository [8] are exhibited to illustrate the classification performance of our methods.

2 Signed Efficiency Measure, Choquet Integral, and Classification

2.1 Signed Efficiency Measure

Let $X = \{x_1, x_2, \cdots, x_n\}$ be the set of all considered feature attributes. The collection of all subsets of *X*, called the power set of *X*, is denoted by $\mathscr{P}(X)$.

Definition 1. Set function $\mu: \mathscr{P}(X) \to (-\infty, \infty)$ is called a signed efficiency measure iff $\mu(\varnothing) = 0$. A signed efficiency measure μ is called subadditive iff $\mu(E \cup F) \le \mu(E) + \mu(F)$ whenever $E \cap F = \varnothing$; μ is called superadditive iff $\mu(E \cup F) \ge \mu(E) + \mu(F)$ whenever $E \cap F = \varnothing$; μ is called additive iff $\mu(E \cup F) = \mu(E) + \mu(F)$ whenever $E \cap F = \varnothing$.

The weights used in the weighted average method can be uniquely extended to an additive measure on $\mathscr{P}(X)$ and the weighted average of observation values for all attributes $x_1, x_2, \cdots,$ and x_n is just the classical Lebesgue integral of the observation values that can be regarded as a function defined on *X*. When a nonadditive signed efficiency measure is adopted, as an aggregation tool, the Lebesgue integral fails. It should be replaced by a nonlinear integral, such as the Choquet integral defined below.

2.2 Choquet Integral

Definition 2. Let f be a real-valued function on *X* and μ be a signed efficiency measure on $\mathscr{P}(X)$. The Choquet integral of f with respect to μ is defined by

$$\int f \, d\mu = \int_{-\infty}^{0} [\mu(F_\alpha) - \mu(X)] \, d\alpha + \int_{0}^{\infty} \mu(F_\alpha) \, d\alpha \tag{1}$$

where $F_\alpha = \{x \mid f(x) \geq \alpha\}$ for $\alpha \in (-\infty, \infty)$.

When f and μ are given, the Choquet integral can be calculated by

$$\int f \, d\mu = \sum_{i=1}^{n} [f(x_i^*) - f(x_{i-1}^*)] \cdot \mu(\{x_i^*, x_{i+1}^*, \cdots, x_n^*\}) \tag{2}$$

where $f(x_0^*) = 0$ and $(x_1^*, x_2^*, \cdots, x_n^*)$ is a permutation of $\{x_1, x_2, \cdots, x_n\}$ such that $f(x_1^*) \leq f(x_2^*) \leq \cdots \leq f(x_n^*)$.

2.3 Classification by Choquet Integral Projections

Based on the Choquet integral, an aggregation tool which projected the feature space onto a real axis had been established. Under the projection, each point in the feature space becomes a value of the virtual variable.

In the feature space, a point $(f(x_1), f(x_2), \cdots, f(x_n))$, denoted by $(f_1, f_2, \cdots, f_n)$ simply, can be regarded as a function f defined on X. It represents an observation of all feature attributes. The basic model of 2-classifications based on the Choquet integral can be expressed as:

$$\text{If } \int (a + bf) \, d\mu \geq c \text{, then } f \in A \text{; otherwise } f \in B,$$

where A and B are two classes. In above expression, $a = (a_1, a_2, \cdots, a_n)$ and $b = (b_1, b_2, \cdots, b_n)$, which balance the scales of the different dimensions, satisfy $\min_i a_i = 0$ and $\max_i |b_i| = 1$, μ is a signed efficiency measure with $\mu(X) = 1$, and c is a real number indicating the classifying boundary. The values of all these parameters can be optimally determined by a soft computing technique, such as a genetic algorithm.

Taking 2-dimentional data (with feature space $X = \{x_1, x_2\}$) belonging to 2 classes (class A and class B) as an example. The classifying boundary is just a contour of the function with two variables f_1 and f_2 expressed by the Choquet integral being a constant c, that is,

$$\int (a + bf) \, d\mu = c, \tag{3}$$

where, $f_1 = f(x_1)$ and $f_2 = f(x_2)$. The contour is a broken line showing in Fig.1. Its vertex is on line L that has equation.

$$a_1 + b_1 f(x_1) = a_2 + b_2 f(x_2).$$

The geometric meaning of $b = (b_1, b_2)$ can be considered as the parameters which determine the slope of the projection axis L, where $a = (a_1, a_2)$ controls the intercept of L from the origin. If $a_1 = a_2$, the projection axis will pass through the axis.

This contour can be extended into high-dimensional data, where the contour is a broken hyper plane with a common vertex.

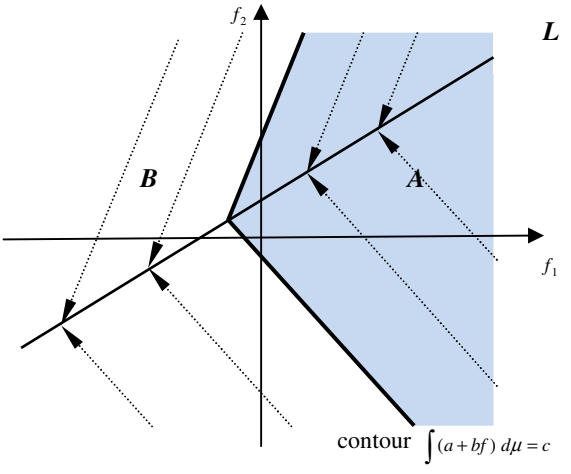


Fig. 1. A contour of the Choquet integral used as the classifying boundary

3 Generalized Classification Model by Cross-Oriented Projection

The applicability of the classification model stated in previous section is limited. It cannot solve the classification problem such as one data class surrounding with another data class. In this case, two Choquet integrals with respect to signed efficiency measure should be used. Relevant works have been published in [6], where the proposed algorithm can solve the classification problems of 2-dimensional data sets successfully. Here, a generalized nonlinear classification model by cross-oriented projection pursuit is introduced. It can solve the classification problems of high-dimensional data sets.

Let $X = \{x_1, x_2, \cdots, x_n\}$ be the set of all considered feature attributes. Let μ and ν be two signed efficiency measures defined on power set $\mathscr{P}(X)$. For convenience, $\mu(\{x_1\})$, $\mu(\{x_2\})$, $\cdots$, $\mu(\{x_n\})$, $\mu(\{x_1, x_2\})$, $\mu(\{x_1, x_3\})$, $\cdots$ are abbreviated by μ_1, μ_2, $\cdots$, μ_n, μ_{12}, μ_{13}, $\cdots$, and $\nu(\{x_1\})$, $\nu(\{x_2\})$, $\cdots$, $\nu(\{x_n\})$, $\nu(\{x_1, x_2\})$, $\nu(\{x_1, x_3\})$, $\cdots$ are abbreviated by ν_1, ν_2, $\cdots$, ν_n, ν_{12}, ν_{13}, $\cdots$, respectively. Assume that, $\mu(X) = \nu(X) = 1$, $-0.5 \le \mu_i \le 1.5$, and $-0.5 \le \nu_i \le 1.5$, where, $i = 1, 2, \cdots, n, 12, 13, \cdots$. Thus, the two Choquet integrals with respect to μ and ν form cross-oriented projections from the feature space onto a real axis. Then a one-dimensional classification can be made on this axis. In such a way, point f satisfying $\int (a+bf)\, d\mu \ge c_\mu$ and $\int (a+bf)\, d\nu \le c_\nu$ will be regarded belonging to one class, say A; while point f satisfying $\int (a+bf)\, d\mu < c_\mu$ or

$\int(a+bf)\,d\nu > c_\nu$ will be regarded belonging to another class, say B. Contour of the classifying boundaries derived by the cross-oriented projection in 2-dimensional case is a quadrilateral, as illustrated in Fig. 2.

The unknown parameters of the generalized classification model by cross-oriented projection are summarized in Table 1.

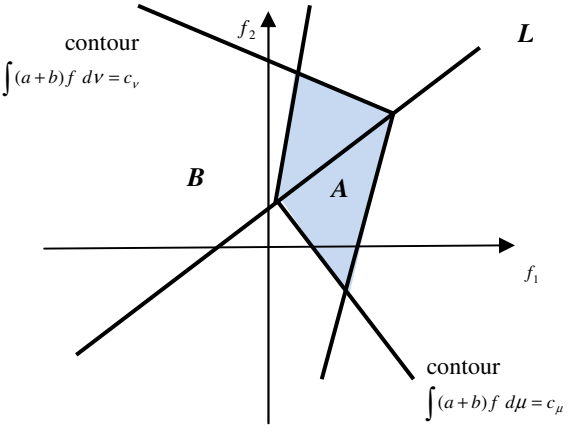


Fig. 2. Nonlinear classification by cross-oriented projection pursuit

Table 1. List of unknown parameters

Parameter name	Meaning	Number
$\boldsymbol{\mu}$	One signed efficiency measure: $\mu_1, \mu_2, ..., \mu_n, \mu_{12}, \mu_{13}, \ldots$	2^n-2
$\boldsymbol{\nu}$	One signed efficiency measure: $\nu_1, \nu_2, ..., \nu_n, \nu_{12}, \nu_{13}, \ldots$	2^n-2
$\boldsymbol{a}$	Parameters to balance the scales $\boldsymbol{a} = (a_1, a_2, ..., a_n)$	n
$\boldsymbol{b}$	Parameters to balance the scales $\boldsymbol{b} = (b_1, b_2, ..., b_n)$	n
$\boldsymbol{C_\mu}$	One dimensional vertex on projection axis L by the Choquet integral with respect to $\boldsymbol{\mu}$	1
$\boldsymbol{C_\nu}$	One dimensional vertex on projection axis L by the Choquet integral with respect to ν	1
Total number		2^{n+1}+2n-2

We called the above-mentioned model as the generalized nonlinear classification model based on cross-oriented Choquet integral. To reduce the number of unknown parameters, a translation transformation and regularization are implemented on original data sets, so that the center of class A coincides with the origin and the projection axis L passes through the origin, as shown in Fig. 3. In this case, the vector $\boldsymbol{a}$ in the model (Eq. (3)) can be released. Then the number of unknown parameters is reduced to $2^{n+1}+n-2$.

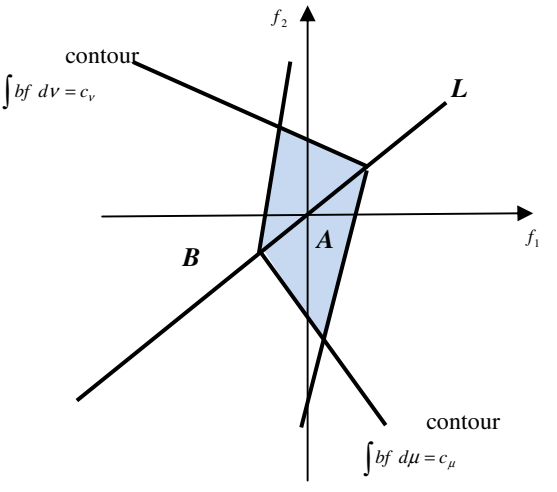


Fig. 3. Contour of classification boundary by $\int bf\ d = c_\mu$ and $\int bf\ d\nu = c_\nu$

4 Algorithms

Since any multi-class classification problem can be decomposed as several consecutive 2-class classification problems, for convenience, we discuss the algorithms of two classes, class *A* and class *B*, in the following statements.

Assume that a rearranged (according to the class) data set $\{f_j \mid j = 1, 2, \cdots, l_1, l_1+1, l_1+2, \cdots, l_1+l_2\}$ is available, where f_1, f_2, $\cdots$, f_{l_1} belong to class *A* while f_{l_1+1}, f_{l_1+2}, $\cdots$, $f_{l_1+l_2}$ belong to class *B*. Positive integers l_1 and l_2 are large enough (not smaller than 20). Each datum f_j is an ordered pair of real numbers $(f_j(x_1), f_j(x_2), \cdots, f_j(x_n))$, $j = 1, 2, \cdots, l_1+l_2$, and can be regarded as a function defined on $X = \{x_1, x_2, \cdots, x_n\}$.

The algorithm consists of the following steps.

1. Finding the data centers.

Calculate the geometric center of the data for class *A*.

$$\begin{aligned}\bar{f}^A &= (\bar{f}_1^A, \bar{f}_2^A, \cdots, \bar{f}_n^A) \\ &= (\frac{1}{l_1}\sum_{j=1}^{l_1} f_j(x_1), \frac{1}{l_1}\sum_{j=1}^{l_1} f_j(x_2), \cdots, \frac{1}{l_1}\sum_{j=1}^{l_1} f_j(x_n))\end{aligned}$$

2. Regularization of the data set.

Let

$$\begin{aligned}g_j &= (g_j(x_1), g_j(x_2), \cdots, g_j(x_n)) \\ &= (\frac{1}{k}(f_j(x_1) - \bar{f}_1^A), \frac{1}{k}(f_j(x_2) - \bar{f}_2^A), \cdots, \frac{1}{k}(f_j(x_n) - \bar{f}_n^A))\end{aligned}$$

for $j = 1, 2, \cdots, l_1 + l_2$, where

$$k = \max_{1 \le j \le l_1} \sqrt{\sum_{i=1}^{n} (f_j(x_i) - \bar{f}_i^A)^2}$$

That is, finding a linear transformation to the data such that class *A* centers at the origin and spreads in a hyper cube $[-1, 1] \times [-1, 1] \times \cdots \times [-1, 1]$.

3. Determining the values of unknown parameters.

A genetic algorithm is designed to optimize the unknown parameters, where each unknown parameter is taken as a gene in chromosome. The ranges of unknown parameters are summaries in Table 2.

Table 2. Ranges of unknown parameters

Parameter name	Ranges
μ	[-0.5, 1.5]
ν	[-0.5, 1.5]
b	[-1, 1]
C_μ	[-1, 1)
C_ν	(-1, 1]

Each chromosome represents a set of the values of these parameters. Using these values, a hyper quadrangular frame expressed by the contours of two Choquet integrals can be obtained as the classifying boundary. Any datum f_j, $j = 1, 2, \cdots, l_1$, is correctly classified if $\int bgd\mu \ge c_\mu$ and $\int bgd\nu \le c_\nu$; otherwise, it is misclassified. Similarly, any datum f_j, $j = l_1 + 1, l_1 + 2, \cdots, l_1 + l_2$, is misclassified if $\int bgd\mu \ge c_\mu$ and $\int bgd\nu \le c_\nu$; otherwise, it is correctly classified. Then, based on the given data, the optimal values of the parameters can be determined by minimizing the misclassification rate.

When there is no learning datum f_j, for some $j = l_1 + 1, l_1 + 2, \cdots, l_1 + l_2$, satisfying $\int bgd\mu < c_\mu$, the part of boundary corresponding to $\int bgd\mu = c_\mu$ can be omitted; similarly, if there is no learning datum f_j, for some $j = l_1 + 1, l_1 + 2, \cdots, l_1 + l_2$, satisfying $\int bgd\nu > c_\nu$, the part of boundary corresponding to $\int bgd\nu = c_\nu$ can be omitted.

4. Exchanging class *A* and *B*

To obtain a better result, exchange the definition of classes *A* and *B*, then redo steps 1 to 3. Compare the result with the previous one. Take the better (with smaller misclassification rate) as the output.

5 Experiments

We have implemented the algorithm shown in Section 4 using Microsoft Visual C++. To verify the classification performance of the algorithms, simulations on artificial data and real data have been conducted.

5.1 Simulation on Artificial Data

The 2-dimensional training data sets are generated by a random number generator in range $[0,1]\times[0,1]$. After regularization, the data set are separated into two classes by two broken lines

$$\int bg\, d\mu = c_\mu \text{ and } \int bg\, d\nu = c_\nu,$$

where $b_1, b_2, \mu_1, \mu_2, \nu_1, \nu_2, c_\mu$, and c_ν are pre-assigned parameters. Each datum $g_j = (g_j(x_1), g_j(x_2))$ is labeled with class *A* if $\int bg_j\, d\mu > c_\mu$ and $\int bg_j\, d\nu < c_\nu$; otherwise, g_j is labeled with *B*. Here, $j = 1, 2, \cdots, l_1 + l_2$ and $l_1 = l_2 = 100$. The original data set after regularization is plotted in Fig. 4.

The data set has 100 points of class *A* and 100 points of class *B*. After regularization, the genetic program runs for searching the classifying boundaries. The population size of this simulation is set as 200. The number of unknown parameters is 8. At 732^{th} generation, zero misclassification rate is achieved. Fig. 5 shows the optimized procedure of the misclassification rate with respect to the genetic generation.

The original data and the classifying boundaries after regularization are drawn in Fig. 6. Here, circles and stars indicate the points in class *A* and *B*, respectively. The solid line is the projection axis to which the 2-dimensional data are projected. It passes through the origin and with slope defined by b_2 / b_1. The dotted broken line is the classifying boundary derived by $\int bgd\mu = c_\mu$, and the dash dot broken line is the classifying boundary derived by $\int bgd\nu = c_\nu$.

The preset parameters and the optimized parameters derived by the genetic algorithm are compared in Table 3. Here, the optimized parameters almost retrieve their preset ones.

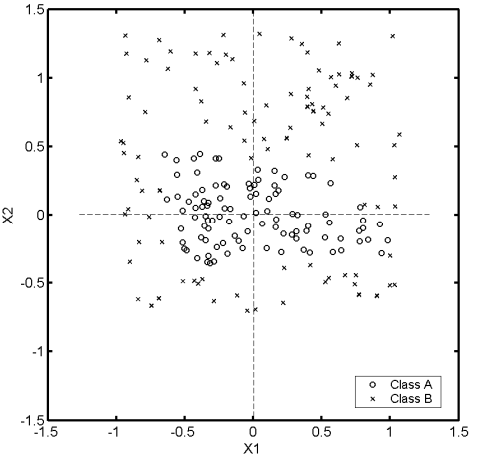


Fig. 4. Original data set after regularization

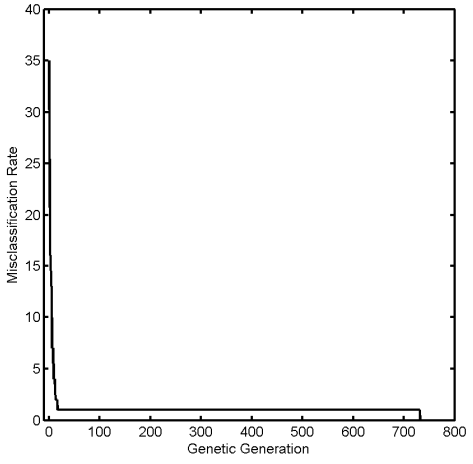


Fig. 5. Misclassification rate with respect to genetic generation

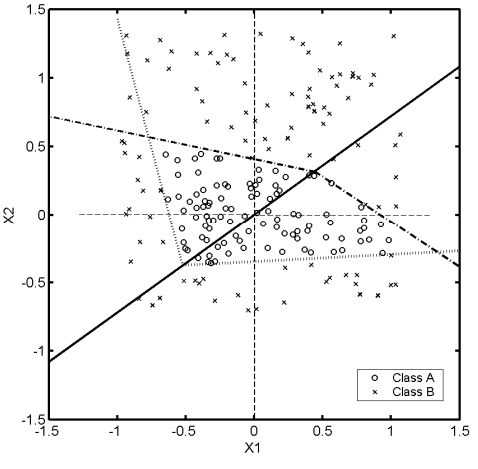


Fig. 6. Classifying boundary derived in simulation on artificial data

Table 3. Comparison between the preset and optimized parameters

Parameter name	Preset value	Optimized value
μ_1	-0.8	-0.796
μ_2	0.2	0.161
ν_1	0.5	0.479
ν_2	0.8	0.776
b_1	0.6	0.596
b_2	0.8	0.828
C_μ	-0.3	-0.309
C_ν	0.3	0.263

5.2 Simulation on Real Data

In case study, considering real multi-class situations, we utilized the IRIS data from UCI [8]. These data include three classes (three IRIS species: Setosa, Versicolor and Virginica) with 50 samples each and 4-dimensional features (the length and the width of the sepal and petal). For convenience, we denote the three classes as class 0, class 1 and class 2. This 3-class classification problem can be decomposed as 3 consecutive 2-class classification problems: class 0 and class 1, class 0 and class 2, class 1 and class2. From Table 1, we know that there are totally $2^{n+1}+n-2=34$, where $n=4$ for 4-dimensional features. They are $\mu_1,\mu_2,\cdots,\mu_{14}$, $v_1,v_2,\cdots,v_{14}$, b_1,b_2,b_3,b_4, c_μ and c_v.

1. Classification of class 0 and class 1.

The genetic program for searching the classifying boundary between class 0 and class 1 runs based on the data set has 50 points of class 0 and 50 points of class 1. The whole data set is used as training data set to retrieve the classifying boundaries.

Consider class 0 as the class *A* in the algorithm stated in Section 4. After regularization, the genetic program runs for searching the classifying boundary. The population size of this simulation is set as 200. In 0^{th} generation, 200 individuals are randomly generated as initial population. In these 200 randomly generated individuals, there exist 2 ones whose corresponding classification boundaries can classify the class 0 and class 1 completely. That means, we get zero misclassification rate at generation 0^{th}. The retrieved unknown parameters of 3 individuals are listed in Table 4.

2. Classification of class 0 and class 2.

The genetic program for searching the classifying boundary between class 0 and class 2 runs based on the data set has 50 points of class 0 and 50 points of class 2. The whole data set is used as training data set to retrieve the classifying boundaries.

Consider class 0 as the class *A* in the algorithm stated in Section 4. After regularization, the genetic program runs for searching the classifying boundary. The population size of this simulation is still set as 200. In 0^{th} generation, there exist 5 ones whose corresponding classification boundaries can classify the class 0 and class 2 completely. That means, we get zero misclassification rate at generation 0^{th} . The retrieved unknown parameters of 7 individuals are listed in Table 5.

3. Classification of class 1 and class 2.

The genetic program for searching the classifying boundary between class 1 and class 2 runs based on the data set has 50 points of class 1 and 50 points of class 2. The whole data set is used as training data set to retrieve the classifying boundaries.

Consider class 1 as the class *A* in the algorithm stated in Section 4. After regularization, the genetic program runs for searching the classifying boundary. The population size of this simulation is set by as 200 as well. The genetic algorithm stops at generation 4000^{th}, which is the preset maximum generation number. At the 4000^{th} generation, the misclassification rate is 1. That means, there is one data which cannot be classified correctly in the training data sets.

Then we consider class 2 as the class A in algorithm stated in Section 4. The genetic algorithm stops at generation 16^{th}, where zero misclassification rate is reached.

There is one individual in generation 16^{th} whose corresponding classification boundaries can classify the class 1 and class 2 completely. The retrieved unknown parameters of this individual are listed in Table 6. Fig. 7 shows the optimized procedure of the misclassification rate with respect to the genetic generation.

Obviously, the performance of the generalized nonlinear classification model based on cross-oriented Choquet integrals and its relevant genetic algorithm is satisfactory by the verification experiments on both artificial and real data set.

Table 4. Retrieved unknown parameters in classifiers of class 0 and class 1

Parameter	Individual 1	Individual 2
μ_1	0.0048232	-0.449075
μ_2	1.19058	1.47242
μ_3	0.398048	0.0688685
μ_4	0.692307	0.164629
μ_5	-0.158026	0.324399
μ_6	-0.184292	-0.271003
μ_7	-0.450792	0.937226
μ_8	0.503475	1.14374
μ_9	0.165395	1.49275
μ_{10}	1.31317	1.49296
μ_{11}	-0.478385	-0.339041
μ_{12}	-0.143717	1.16854
μ_{13}	-0.42883	0.658134
μ_{14}	0.0608117	1.44526
ν_1	-0.222677	1.1743
ν_2	0.253379	0.337968
ν_3	0.652973	-0.493873
ν_4	0.282479	-0.372947
ν_5	-0.349608	1.10865
ν_6	1.23432	-0.45856
ν_7	0.25334	1.24356
ν_8	0.0439466	1.15741
ν_9	0.983515	1.28242
ν_{10}	-0.258399	0.0530433
ν_{11}	0.291705	1.06404
ν_{12}	-0.0326033	-0.287872
ν_{13}	1.45221	-0.288123
ν_{14}	-0.0254469	-0.0602971
b_1	0.711253	-0.0340049
b_2	-0.374262	0.0179182
b_3	-0.724922	-0.935174
b_4	0.138267	-0.113526
C_μ	-0.995321	-0.589644
C_ν	0.431257	0.922721

Table 5. Retrieved unknown parameters in classifiers of class 0 and class 2

Parameter	Individual 1	Individual 2	Individual 3	Individual 4	Individual 5
μ_1	0.0048232	0.679669	1.46229	-0.449075	-0.305349
μ_2	1.19058	1.46651	1.18286	1.47242	1.35059
μ_3	0.398048	-0.445465	0.127911	0.0688685	0.887157
μ_4	0.692307	0.606251	0.773051	0.164629	1.01912
μ_5	-0.158026	-0.196814	0.462173	0.324399	-0.300904
μ_6	-0.184292	0.73977	1.11407	-0.271003	0.327651
μ_7	-0.450792	-0.0619089	0.265001	0.937226	0.0408298
μ_8	0.503475	-0.335183	0.846153	1.14374	-0.202304
μ_9	0.165395	1.45156	0.57368	1.49275	0.219136
μ_{10}	1.31317	0.43558	0.0481065	1.49296	-0.316584
μ_{11}	-0.478385	0.671609	0.44541	-0.339041	0.88649
μ_{12}	-0.143717	0.909113	1.47536	1.16854	0.658876
μ_{13}	-0.42883	1.07041	0.398209	0.658134	0.712251
μ_{14}	0.0608117	0.853584	0.0447845	1.44526	0.64552
ν_1	-0.222677	0.889886	1.35938	1.1743	-0.196435
ν_2	0.253379	-0.474606	1.4578	0.337968	0.771025
ν_3	0.652973	0.564477	0.833294	-0.493873	1.22056
ν_4	0.282479	0.617486	-0.429183	-0.372947	1.16579
ν_5	-0.349608	0.0837916	0.642148	1.10865	1.429
ν_6	1.23432	0.617603	-0.314279	-0.45856	-0.167072
ν_7	0.25334	0.432854	0.845429	1.24356	1.1304
ν_8	0.0439466	0.0456995	0.0412946	1.15741	0.705507
ν_9	0.983515	0.696691	0.788805	1.28242	-0.317401
ν_{10}	-0.258399	0.548281	0.412069	0.0530433	1.16213
ν_{11}	0.291705	-0.0123057	0.511328	1.06404	1.29819
ν_{12}	-0.0326033	0.882966	0.784715	-0.287872	0.0131133
ν_{13}	1.45221	-0.294089	0.62227	-0.288123	-0.215865
ν_{14}	-0.0254469	-0.239225	0.294133	-0.0602971	1.18375
b_1	0.711253	0.867482	-0.673768	-0.0340049	0.417769
b_2	-0.374262	0.445345	0.330747	0.0179182	0.491805
b_3	-0.724922	-0.639646	-0.530299	-0.935174	-0.640386
b_4	0.138267	-0.952714	-0.532176	-0.113526	0.835031
C_μ	-0.995321	-0.31443	-0.355089	-0.589644	-0.151552
C_ν	0.431257	0.977191	0.393777	0.922721	0.665952

Table 6. Retrieved unknown parameters in classifier of class 1 and class 2

Parameter	Value	Parameter	Value	Parameter	Value
μ_1	-0.155727	μ_{13}	1.37527	ν_{11}	1.12347
μ_2	0.698759	μ_{14}	0.858355	ν_{12}	-0.267951
μ_3	0.410291	ν_1	0.939024	ν_{13}	-0.184767
μ_4	0.69709	ν_2	0.867304	ν_{14}	0.662568
μ_5	0.848792	ν_3	0.123789	b_1	-0.108787
μ_6	1.11531	ν_4	-0.14382	b_2	-0.17422
μ_7	1.33937	ν_5	0.989996	b_3	0.478004
μ_8	1.04216	ν_6	0.715143	b_4	0.591394
μ_9	0.187169	ν_7	1.40483	C_μ	-0.0474166
μ_{10}	0.568226	ν_8	0.33452	C_ν	0.813631
μ_{11}	0.994638	ν_9	0.558908		
μ_{12}	0.793781	ν_{10}	-0.0523634		

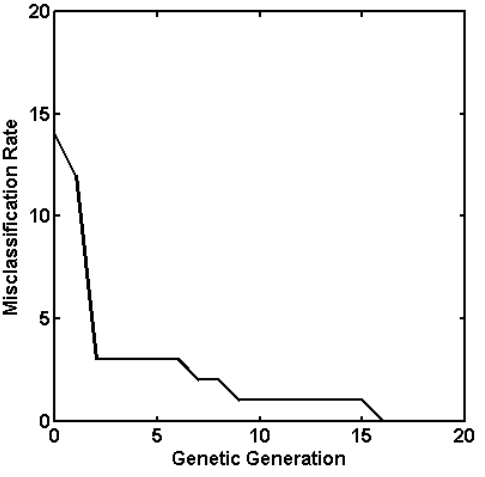


Fig. 7. Misclassification rate with respect to genetic generation in experiments of class 1 and class 2 of IRIS data set

6 Conclusions

Based on our previous work of nonlinear classification model based on cross-oriented Choquet integrals, we present a more generalized model which can classify high-dimensional data set in this paper. We stated this model and its relevant algorithm based on 2-class classification problem for convenience. It can be extended to multi-class multi-dimensional situations since any multi-class classification problem can be decomposed as several consecutive 2-class classification problems. Stage of experiments on both artificial and real data verify the performance of the nonlinear classification model based on cross-oriented Choquet integrals. The classification accuracy of IRIS data (whole data set perform both training and testing data set) is 100%. This result is superior to those of existed classification models. Of course, more rigorous tests, such as ten-fold validation, on other real data sets are expected to be implemented in future works.

Acknowledgement. This work was supported by National Natural Science Foundation of China (Grant No. 61105044).

References

1. Denneberg, D.: Non-additive Measure and Integral. Kluwer, Boston (1994)
2. Wang, Z., Klir, G.K.: Fuzzy Measure Theory. Plenum Press, New York (1992)
3. Murofushi, T., Sugeno, M., Machida, M.: Non monotonic fuzzy measures and the Choquet integral. Fuzzy Sets and Systems 64, 73–86 (1994)
4. Pap, E.: Null-Additive Set Functions. Kluwer Academic Publisher, Boston (1995)

5. Xu, K., Wang, Z., Heng, P.-A., Lueng, K.-S.: Classification by nonlinear integrals projections. IEEE Trans. Fuzzy Syst. 11(2), 187–201 (2003)
6. Wang, Z., Yang, R., Shi, Y.: A new nonlinear classification model based on cross-oriented Choquet integrals. In: Proc. Inter. Conf. Information Science and Technology, Nanjing, China, pp. 176–181 (2011)
7. Fang, H., Rizzo, M.L., Wang, H., Espy, K.A., Wang, Z.: A new nonlinear classifier with a penalized signed fuzzy measure using effective genetic algorithm. Pattern Recognition 43, 1393–1401 (2010)
8. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
9. Wang, Z., Yang, R., Leung, K.S.: Nonlinear Integrals and Their Application on Data Mining. World Scientific, Signapore (2010)

A General Lp-norm Support Vector Machine via Mixed 0-1 Programming

Hai Thanh Nguyen and Katrin Franke

NISlab, Department of Computer Science and Media Technology,
Gjøvik University College, P.O. box 191, N-2802 Gjøvik, Norway
{hai.nguyen,katrin.franke}@hig.no
www.nislab.no

Abstract. Identifying a good feature subset that contributes most to the performance of Lp-norm Support Vector Machines (Lp-SVMs with $p = 1$ or $p = 2$) is an important task. We realize that the Lp-SVMs do not comprehensively consider irrelevant and redundant features, because the Lp-SVMs consider all n full-set features be important for training while skipping other $2^n - 1$ possible feature subsets at the same time. In previous work, we have studied the L1-norm SVM and applied it to the feature selection problem. In this paper, we extend our research to the L2-norm SVM and propose to generalize the Lp-SVMs into one general Lp-norm Support Vector Machine (GLp-SVM) that takes into account all 2^n possible feature subsets. We represent the GLp-SVM as a mixed 0-1 nonlinear programming problem (M01NLP). We prove that solving the new proposed M01NLP optimization problem results in a smaller error penalty and enlarges the margin between two support vector hyperplanes, thus possibly giving a better generalization capability of SVMs than solving the traditional Lp-SVMs. Moreover, by following the new formulation we can easily control the sparsity of the GLp-SVM by adding a linear constraint to the proposed M01NLP optimization problem. In order to reduce the computational complexity of directly solving the M01NLP problem, we propose to equivalently transform it into a mixed 0-1 linear programming (M01LP) problem if $p = 1$ or into a mixed 0-1 quadratic programming (M01QP) problem if $p = 2$. The M01LP and M01QP problems are then solved by using the branch and bound algorithm. Experimental results obtained over the UCI, LIBSVM, UNM and MIT Lincoln Lab datasets show that our new proposed GLp-SVM outperforms the traditional Lp-SVMs by improving the classification accuracy by more than 13.49%.

Keywords: support vector machine, feature selection, mixed 0-1 quadratic programming, branch and bound.

1 Introduction

The Lp-norm Support Vector Machines (Lp-SVMs, with $p = 1$ or $p = 2$) were studied in many previous works and were demonstrated to be efficient in solving

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 40–49, 2012.

a broad range of different practical problems (see, for example, [1–7, 10, 11, 20–22]). As other machine learning classifiers, the performance of Lp-SVMs strongly depends on the quality of features from a dataset. The existence of irrelevant and redundant features in the dataset can reduce the accuracy of Lp-SVMs. We realize that the traditional Lp-SVMs do not comprehensively consider irrelevant and redundant features. In fact, the Lp-SVMs consider all n full-set features be important for training. However, there probably exist irrelevant and redundant features among n full-set features. Furthermore, in many cases L2-SVM was shown not to select any features [1]. The L1-SVM provides some relevant features, but it cannot remove redundant features [1]. In order to improve the performance of the Lp-SVMs by looking at important features, it is necessary to test all 2^n possible combinations of features for training. In previous work [24], we have studied the L1-norm SVM and applied it to the feature selection problem. In this paper, we extend our research to the L2-norm SVM and propose to generalize the Lp-SVMs into one general Lp-norm Support Vector Machine (GLp-SVM) that takes into account all 2^n possible feature subsets.

In the formulation of the GLp-SVM, we encode the data matrix by using the binary variables x_k $(k = \overline{1,n})$ for indicating the appearance of the k^{th} feature $(x_k = 1)$ or the absence of the k^{th} feature $(x_k = 0)$. Following this proposed encoding scheme, our GLp-SVM can be represented as a mixed 0-1 nonlinear programming problem (M01NLP). The objective function of this M01NLP is a sum of the inverse value of margin by means of Lp-norm and the error penalty. We prove that the minimal value of the objective function from the GLp-SVM is not greater than the one from the traditional Lp-SVMs. As a consequence, solving our new proposed M01NLP optimization problem results in a smaller error penalty and enlarges the margin between two support vector hyper-planes, thus possibly giving better generalization capability of SVM than those obtained by solving the traditional Lp-SVMs. Moreover, by following the new general formulation we can easily control the sparsity of the GLp-SVMs by adding the constraint $x_1 + x_2 + .. + x_n = T$, where T is the number of important features, to the proposed M01NLP optimization problem.

In order to reduce the computational complexity of directly solving the M01NLP problem, we apply Chang's method [8, 9] to equivalently transform it into a mixed 0-1 linear programming (M01LP) problem if $p = 1$ or a mixed 0-1 quadratic programming (M01QP) problem if $p = 2$. The obtained M01LP and M01QP problems can then be efficiently solved by using the branch and bound algorithm. In order to validate our theoretical findings, in this paper we have compared our new GLp-SVM with the standard L2-norm SVM [10, 11] and the traditional L1-norm SVM proposed by Bradley and Mangasarian [1] regarding the classification accuracy and the number of selected important features. Experimental results obtained over the UCI [12], LIBSVM [17], UNM [19] and MIT Lincoln Lab [18] data sets show that the new GLp-SVM gives better classification accuracy, while in many cases selecting fewer features than the traditional Lp-SVMs do.

The paper is organized as follows. Section 2 formally defines a new general formulation of Lp-SVMs (GLp-SVM). We show how to represent GLp-SVM as a mixed 0-1 nonlinear programming problem (M01NLP) by the encoding scheme. In this section, we also prove that the minimal value of the objective function from the GLp-SVM is not greater than the one from the traditional Lp-SVMs. Section 3 describes our new search approach for globally solutions of the GLp-SVM. We present experimental results in Section 4. The last section summarizes our findings.

2 A General Lp-norm Support Vector Machine

Our goal is to develop a new SVM-based method capable of identifying the best feature subset for classification. To achieve this goal, our first contribution is a novel general formulation of the Lp-norm Support Vector Machines (Lp-SVMs). In the standard Lp-SVMs, we are given a training data set D with m instances: $D = \{(a_i, c_i)|a_i \in \mathbb{R}^n, c_i \in \{-1, 1\}\}_{i=1}^m$, where a_i is the i^{th} instance that has n features and c_i is a class label. a_i can be represented as a data vector as follows: $a_i = (a_{i1}, a_{i2}, ..., a_{in})$, where a_{ij} is the value of the j^{th} feature in the instance a_i.

For the two-class classification problem, a Support Vector Machine (SVM) learns a separating hyper-plane $w \cdot a_i = b$ that maximizes the margin distance $\frac{2}{||w||_p^p}$, where $w = (w_1, w_2, ..., w_n)$ is the weight vector and b is the bias value. The standard form of the SVM is given below [10, 11]:

$$\min_{w,b,\xi} \frac{1}{p}\|w\|_p^p + C\sum_{i=1}^{m}\xi_i,$$

$$such\,that \quad \begin{cases} c_i(\sum_{j=1}^{n} a_{ij}w_j - b) \geq 1 - \xi_i, \\ \xi_i \geq 0, i = \overline{1, m}. \end{cases} \tag{1}$$

Above, ξ_i is slack variable, which measures the degree of misclassification of the instance a_i, and $C > 0$ is the error penalty parameter; $p = 1$ or $p = 2$. If $p = 1$, then we have the L1-norm support vector machine that was first proposed by Bradley and Mangasarian [1]. If $p = 2$, then we have the traditional L2-norm support vector machine [10].

From (1), we observe that the traditional Lp-norm SVMs consider all n features be important for training. However, there probably exist irrelevant and redundant features among n features of the dataset [14, 15]. The performance of Lp-SVMs might be reduced because of these features. Therefore, it is necessary to test all 2^n possible feature subsets for training the Lp-SVMs.

When the number of features n is small, we can apply the brute force method to scan all 2^n subsets. However if this number of features becomes large, a more computational efficient method that also ensures the best feature subset is required. In the following, we first show how to generalize the problem (1) into a general Lp-norm SVM (GLp-SVM), which is in fact a mixed 0-1 nonlinear programming (M01NLP) problem. We then describe how to solve this M01NLP optimization problem in order to get globally optimal solutions.

Firstly, we use the binary variables x_k $(k = \overline{1,n})$ for indicating the appearance of the k^{th} feature $(x_k = 1)$ or the absence of the k^{th} feature $(x_k = 0)$ to encode the data vector a_i $(i = \overline{1,m})$ as follows:

$$\begin{cases} a_i = (a_{i1}x_1, a_{i2}x_2, ..., a_{in}x_n). \\ x = (x_1, x_2, ..., x_n) \in \{0,1\}^n. \end{cases} \tag{2}$$

With the encoding scheme (2), the problem (1) can be generalized into the following mixed 0-1 nonlinear programming (M01NLP) problem:

$$\min_{w,\xi,b,x} \frac{1}{p}\|w\|_p^p + C\sum_{i=1}^{m}\xi_i,$$

$$\begin{cases} c_i\big(\sum_{j=1}^{n} a_{ij}w_jx_j - b\big) \geq 1 - \xi_i, \\ \xi_i \geq 0, i = \overline{1,m}; C > 0, \\ x_j \in \{0,1\}, j = \overline{1,n}. \end{cases} \tag{3}$$

Proposition 1: Suppose that $S1$ and $S2$ are the minimal values of the objective functions from (1) and (3), respectively. The following inequality is true:

$$S2 \leq S1 \tag{4}$$

Proof. It is obvious, since the problem (1) is a case of the problem (3) when $x = (x_1, x_2, .., x_n) = (1, 1, .., 1)$. □

Remark

1. As a consequence of the Proposition 1, solving the problem (3) results in a smaller error penalty and enlarges the margin between two support vector hyper-planes, thus possibly giving a better generalization capability of SVM than solving the traditional Lp-norm SVMs in (1).
2. We can even control the sparsity of the general Lp-norm SVMs by adding the following linear constraint $x_1 + x_2 + .. + x_n = T$, where $T \leq n$ is the number of important features, to the optimization problem (3).
3. Normally, we can apply popular optimization techniques, such as the branch and bound algorithm, to directly solve a mixed 0-1 non-linear programming problem. However, with non-linear constraints, the problem (3) becomes even harder to solve. In the next section, we propose a new approach to linearize the constraints in (3), thus reducing the computational complexity of solving (3).

3 Optimizing General Lp-norm Support Vector Machine

The main idea of our new proposed method is to linearize mixed $0-1$ terms w_jx_j in (3) applying Chang's method. By this way, we equivalently transform the optimization problem (3) into a mixed 0-1 linear programming (M01LP)

problem if $p = 1$ or into a mixed 0-1 quadratic programming (M01QP) problem if $p = 2$. The obtained M01LP and M01QP problems can then be efficiently solved by using the branch and bound algorithm.

Proposition 2: A mixed $0 - 1$ term $w_j x_j$ from (3) can be represented by a continuous variable z_j, subject to the following linear inequalities [8, 9], where M is a large positive value:

$$\begin{cases} z_j \geq M(x_j - 1) + w_j, \\ z_j \leq M(1 - x_j) + w_j, \\ 0 \leq z_j \leq M x_j. \end{cases} \tag{5}$$

Proof

(a) If $x_j = 0$, then (5) becomes

$$\begin{cases} z_i \geq M(0 - 1) + w_j, \\ z_i \leq M(1 - 0) + w_j, \\ 0 \leq z_i \leq 0, \end{cases}$$

z_i is forced to be zero, because M is a large positive value.

(b) If $x_j = 1$, then (5) becomes

$$\begin{cases} z_i \geq M(1 - 1) + w_j, \\ z_i \leq M(1 - 1) + w_j, \\ 0 \leq z_i \leq M, \end{cases}$$

z_i is forced to be w_j, because M is a large positive value.

Therefore, the constraints on z_i reduce to:

$$z_i = \begin{cases} 0, & if \quad x_j = 0, \\ w_j, & if \quad x_j = 1. \end{cases}$$

which is the same as $w_j x_j = z_i$.

Remark

1. All the constraints in (3) are now linear. We consider the first case when $p = 1$. We define $w = p - q$ with $p = (p_1, p_2, ..., p_n) \geq 0, q = (q_1, q_2, ..., q_n) \geq 0$ and $e_n^T = (1, 1, ..., 1)$. The problem (3) is then equivalent to the following problem [20]:

$$\min_{p,q,\xi,b,x} e_n^T (p + q) + C \sum_{i=1}^{m} \xi_i,$$

$$\begin{cases} c_i \big(\sum_{j=1}^{n} a_{ij}(p_j - q_j)x_j - b \big) \geq 1 - \xi_i, \\ \xi_i \geq 0, i = \overline{1, m}; C > 0, \\ x_j \in \{0, 1\}, \\ p_j, q_j \geq 0, j = \overline{1, n}. \end{cases} \tag{6}$$

By applying the Proposition 2, we substitute terms $p_j x_j$ and $q_j x_j$ in (6) by new variables t_j and v_j, respectively, satisfying linear constraints. By doing this substitution, we in fact transform the problem (6) into a mixed 0-1 linear programming (M01LP) problem. The total number of variables for the M01LP problem will be $6n+m+1$, as they are b, ξ_i, x_j, p_j, q_j, t_j, z_j and $v_j (i = 1, ..., m; j = 1, ..., n)$. Therefore, the number of constraints on these variables will also be a linear function of n. We propose to use the branch and bound algorithm to solve this M01LP problem.

2. If $p = 2$, then it is obvious that the optimization problem (3) is equivalent to the following mixed 0-1 quadratic programming problem, which can be solved by using the branch and bound algorithm:

$$\min_{w,\xi,b,x,z} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{m}\xi_i,$$

$$\begin{cases} c_i\left(\sum_{j=1}^{n} a_{ij} z_j - b\right) \geq 1 - \xi_i, \\ z_j \geq M(x_j - 1) + w_j, \\ z_j \leq M(1 - x_j) + w_j, \\ 0 \leq z_j \leq M x_j, \\ \xi_i \geq 0, i = \overline{1, m}; C > 0, \\ x_j \in \{0, 1\}, j = \overline{1, n}. \end{cases} \tag{7}$$

4 Experiment

This section validates the capability of our new proposed general Lp-norm SVM (GLp-SVM) in dealing with irrelevant and redundant features. In order to do that, in this paper we only consider the linear SVMs for binary classification problem. The goal is to compare our new GLp-SVM method with the L2-norm SVM [11] and the traditional L1-norm SVM proposed by Bradley and Mangasarian [1] regarding the number of selected features and the generalization capability. Note that the number of selected features in context of linear SVM for the binary classification problem is the number of nonzero elements of the weight vector.

4.1 Experimental Settings

We conducted our experiments on various benchmark datasets from the UCI [12] and the LIBSVM repositories [17]. The datasets we tested are "a1a", "a2a", "w1a", "w2a", "Spectf" and "Haberman". Not only we are interested in datasets with large numbers of features, such as "w1a" that has 300 features, for the experiments, but also we tested our algorithms on small datasets, such as "Haberman" that has only 3 features. The reason is to test the ability of our algorithms in finding the most important features in diverse datasets.

We also used several benchmark datasets from different fields to validate our new algorithms. In particular, we selected datasets from *(1)* UNM [19] and *(2)* the MIT Lincoln Lab databases [18].

Ad. 1. The University of New Mexico (UNM) provides a number of system call datasets [19] for testing host-based intrusion detection systems (HIDSs). A HIDS is an agent on a host that identifies intrusions or attacks to computer systems by analyzing system calls, application logs and file system modifications [23]. In the UNM datasets, a feature is defined as the number of occurrence of a system call in an input sequence. Each instance in the UNM datasets, which is labeled as as "normal" or "intrusion", has a large number of feature values, such as Xlock has 200 features. We used five different datasets from the UNM database: "L-inetd","Login", "PS","S-lpr" and "Xlock". More detail on the numbers of features of the datasets is given in Table 1.

Ad. 2. The datasets generated by MIT Lincoln Lab in 1998 [18] were used for benchmarking of different intrusion detection systems. However, in this paper we only considers two datasets for testing host-based intrusion detection systems: "RawFriday" and "RawMonday" datasets. The numbers of features in the MIT Lincoln Lab datasets are less than the ones from the UNM datasets. For example, the "RawFriday" and the "RawMonday" have 53 and 54 features, respectively.

We needed to run four different algorithms on the chosen datasets: L1-norm SVM, L2-norm SVM, GL1-norm SVM and GL2-norm SVM. In order to implement the L2-norm SVM and the traditional L1-norm SVM, we used the Mangasarian's code from [16]. For implementing the new general Lp-norm SVM (GLp-SVM, p=1 and p=2), the TOMLAB tool [13] was used for solving the mixed 0-1 linear programming problem if $p = 1$ and the mixed 0-1 quadratic programming problem if $p = 2$. The values of the error penalty parameter C used for the experiment were: $2^{-7}, 2^{-6}, 2^{-5}, ..., 2, ..., 2^5, 2^6, 2^7$. We applied 10-fold cross validation for estimating the average classification accuracies as well as the average number of selected features. All the best results obtained over those penalty parameters were chosen and are given in the Table 1 and the Table 2.

4.2 Experimental Results

Table 1 shows the number of features selected by our GLp-norm SVMs and those selected by the traditional Lp-norm SVMs. Table 2 summaries the classification accuracies of 10 folds cross-validation of the SVMs performed on 13 datasets.

It can be observed from Table 1 that our new method GL1-norm SVM removes dramatically irrelevant and redundant features from the full-set of features. Surprisingly, in some cases the GL1-norm SVM selected only one important feature, such as in the Xlock and RawMonday datasets. Moreover, in comparison with other methods the GL1-SVM provided a smallest number of important features. The GL2-norm SVM selects smaller numbers of features than the L2-norm SVM, but larger than the L1-norm SVM.

From the Table 2, it can be seen that the GL1-norm SVM provided higher performance than the one given by the traditional L1-norm SVM. In fact, the GL1-norm

Table 1. Number of selected features (on average)

Data Sets	Full-set	L2-norm SVM	GL2-SVM	L1-norm SVM	GL1-SVM
a1a [17]	123	105.3	105.3	64	3.5
a2a [17]	123	107.6	96	74.9	3.8
w1a [17]	300	266.2	266.2	76.7	11.1
w2a [17]	300	270.5	270.5	99.9	10.2
Spectf [12]	44	44	33	31	6
Haberman [12]	3	3	3	2.8	1.6
RawFriday [18]	53	33.9	33.9	8.4	1.9
RawMonday [18]	54	26	26	1	1
L-inetd [19]	164	33.5	23	13.5	2.4
Login [19]	164	46	30	9.6	2
PS [19]	164	22	22	5	2
S-lpr [19]	182	36.9	36.9	3.2	2
Xlock [19]	200	46.8	46.8	13.4	1
Average	**144.1**	**80.1**	**76.35**	**31**	**3.7**

SVM improved the classification accuracy by more than 13.49%. This phenomenon can be explained by the fact that L1-norm SVM is just a case of the GL1-norm SVM, thus solving the GL1-norm SVM results a better classification accuracy than solving the traditional L1-norm SVM. Moreover, the datasets contain many irrelevant and redundant features that negatively affect the performance of SVMs. These explanations can also be applied to the case of the GL2-norm SVM and L2-norm SVM. As shown in Table 2, GL2-norm SVM gave the best classification accuracy on average. In some cases, such as in the Xlock and RawMonday datasets, one feature is enough to identify attacks and normal traffic.

Table 2. Classification Accuracies (on average)

Data Sets	L2-norm SVM	GL2-SVM	L1-norm SVM	GL1-SVM
a1a [17]	83.99	83.99	65.40	75.37
a2a [17]	82.25	90.34	68.34	74.75
w1a [17]	96.76	96.76	88.49	97.09
w2a [17]	96.69	96.69	85.76	96.92
Spectf [12]	72.20	83.00	79.55	79.55
Haberman [12]	73.48	73.48	73.16	73.48
RawFriday [18]	98.40	100	54.02	98.80
RawMonday [18]	100	100	95.65	100
L-inetd [19]	88.33	90.05	85.00	85.83
Login [19]	80.00	85.00	65.00	81.67
PS [19]	100	100	100	100
S-lpr [19]	100	100	70	99.11
Xlock [19]	100	100	56.79	100
Average	**90.16**	**92.25**	**75.93**	**89.42**

5 Conclusions

We have proposed a new general Lp-norm SVM (GLp-SVM) that considers comprehensively irrelevant and redundant features of a dataset. The central idea was to use binary variables for encoding the data matrix. The new GLp-SVM can then be represented as a mixed 0-1 nonlinear programming problem (M01NLP). We proved that the traditional Lp-norm SVMs are a special case of our new GLp-SVM. Therefore, solving the new proposed M01NLP optimization problem results in a smaller error penalty and enlarges the margin between two support vector hyper-planes, thus possibly giving better generalization capability of SVM than solving the traditional Lp-norm SVMs problem. We also proposed to equivalently transform the M01NLP problem into a mixed 0-1 linear programming (M01LP) problem if $p = 1$ or a mixed 0-1 quadratic programming (M01QP) problem if $p = 2$. The M01LP and M01QP problems can then be solved by using the branch and bound algorithm. Experimental results obtained over the UCI, the LIBSVM, the UNM and the MIT Lincoln Lab data sets show that the new general Lp-norm SVMs gives better generalization capability, while in many cases selecting fewer features than the traditional Lp-norm SVMs do.

References

1. Bradley, P., Mangasarian, O.L.: Feature selection via concave minimization and support vector machines. In: Proceedings of the Fifteenth International Conference (ICML), pp. 82–90 (1998)
2. Mangasarian, O.L.: Exact 1-Norm Support Vector Machines Via Unconstrained Convex Differentiable Minimization (Special Topic on Machine Learning and Optimization). Journal of Machine Learning Research 7(2), 1517–1530 (2007)
3. Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., Vapnik, V.: Feature selection for SVMs. In: Advances in Neural Information Processing Systems, pp. 668–674 (2001)
4. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46(1), 389–422 (2002)
5. Guan, W., Gray, A., Leyffer, S.: Mixed-Integer Support Vector Machine. In: NIPS Workshop on Optimization for Machine Learning (2009)
6. Neumann, J., Schnorr, C., Steidl, G.: Combined SVM-based feature selection and classification. Machine Learning 61(1), 129–150 (2005)
7. Rakotomamonjy, A.: Variable selection using SVM based criteria. Journal of Machine Learning Research 3, 1357–1370 (2003)
8. Chang, C.-T.: On the polynomial mixed 0-1 fractional programming problems. European Journal of Operational Research 131(1), 224–227 (2001)
9. Chang, C.-T.: An efficient linearization approach for mixed integer problems. European Journal of Operational Research 123, 652–659 (2000)
10. Vapnik, V.: The Nature of Statistical Learning Theory. Springer (1995)
11. Cortes, C., Vapnik, V.: Support-Vector Networks. In: Machine Learning, pp. 273-297 (1995)
12. Murphy, P.M., Aha, D.W.: UCI repository of machine learning databases. Technical report, Department of Information and Computer Science, University of California, Irvine (1992), http://www.ics.uci.edu/mlearn/MLRepository.html

13. TOMLAB, The optimization environment in MATLAB, http://tomopt.com/tomlab/
14. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.A.: Feature Extraction: Foundations and Applications. STUDFUZZ. Physica-Verlag, Springer (2006)
15. Liu, H., Motoda, H.: Computational Methods of Feature Selection. Chapman & Hall/CRC (2008).
16. DMI Classification Software, http://www.cs.wisc.edu/dmi/
17. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), Data sets and software, http://www.csie.ntu.edu.tw/cjlin/libsvm/
18. Lippmann, R.P., Graf, I., Garfinkel, S.L., Gorton, A.S., Kendall, K.R., McClung, D.J., Weber, D.J., Webster, S.E., Wyschogrod, D., Zissman, M.A.: The 1998 DARPA/AFRL off-line intrusion detection evaluation. Presented to The First Intl. Work Workshop on Recent Advances in Intrusion Detection (RAID 1998) (No Printed Proceedings) Lovain-la-Neuve, Belgium, September 14-16 (1998)
19. UNM (University of New Mexico) audit data, http://www.cs.unm.edu/~immsec/systemcalls.htm
20. Bennett, K.P., Mangasarian, O.L.: Robust linear programming discrimination of two linearly inseparable sets. Optimization Methods and Software 1(1), 23–34 (1992)
21. Zhu, J., Rosset, S., Hastie, T., Tibshirani, R.: 1-norm support vector machines. In: Neural Information Processing Systems (2003)
22. Wang, L., Xiatong, S.: On L1-Norm Multiclass Support Vector Machines: Methodology and Theory. Journal of the American Statistical Association 102, 583–594 (2007)
23. Newman, R.C.: Computer Security: Protecting Digital Resources. Jones & Bartlett Learning (2009) ISBN 0763759945
24. Nguyen, H.T., Franke, K., Petrovi'c, S.: On General Definition of L1-norm Support Vector Machines for Feature Selection. The International Journal of Machine Learning and Computing 1(3), 279–283 (2011)

Reduction of Distance Computations in Selection of Pivot Elements for Balanced GHT Structure

László Kovács

University of Miskolc, Department of Information Technology, Miskolc, Hungary
kovacs@iit.uni-miskolc.hu

Abstract. In general metric spaces, one of the most widely used indexing techniques is the partitioning of the objects using pivot elements. The efficiency of partitioning depends on the selection of the appropriate set of pivot elements. In the paper, some methods are presented to improve the quality of the partitioning in GHT structure from the viewpoint of balancing factor. The main goal of the investigation is to determine the conditions when costs of distance computations can be reduced. We show with different tests that the proposed methods work better than the usual random and incremental pivot search methods.

Keywords: pivot based indexing, general metric space, interval based computation.

1 Introduction

In many application areas the objects can't be represented with appropriate feature vectors, only the distances between the objects are known. If the distance function *d()* is a metric it fulfills the following conditions:

$$\begin{gathered} d(x,y) \geq 0 \\ d(x,y) = 0 \leftrightarrow x = y \\ d(x,y) = d(y,x) \\ d(x,z) + d(z,y) \geq d(x,y) \end{gathered} \tag{1}$$

The different application areas may have very different and complex distance functions. For example the detection and comparison of components for sound data objects, are relatively expensive operations. In the case of applications with huge collection of these objects, the objects are clustered and indexed to reduce the computational costs on the collection. The most widely used indexing methods in general metric spaces use pivot elements. The pivot element p is a distinguished object from the object-set. The distance from an object x to p is used as the indexing key value of x to locate the bucket containing x. Usually more than one single pivot element are used in the algorithms.

It is known that the efficiency of indexing methods depends significantly on the position of the pivot elements [4], thus the appropriate selection of the pivot elements is a crucial optimization component of object management. The usual measure to calculate the fitness of a pivot-set is the mean of distance distribution [1]:

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 50–62, 2012.

$$\mu_{p_1,..,p_M} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} max_{k\in 1..M}\{|d(x_i,p_k) - d(x_j,p_k)|\} \quad (2)$$

where N denotes the number of objects in the set, This measure is tailored to reduction of the pruning operation in the search tree. Having a nearest neighbor query $Q(q,r)$ where q is the query object and r is the threshold distance, a branch of the search tree assigned to pivot element set $\{p_i\}$ can be excluded if for all elements x of the subtree

$$|d(x,p_i) - d(q,p_i)| > r \quad (3)$$

is met for some index i. As the exact calculation of the measure μ is an $O(N)$ operation, only a sampling with $O(1)$ is used to estimate the fitness parameter value.

There are many variants of indexing trees in general metric spaces. The Generalized Hyperplane Tree (GHT) [5] and Bisector Tree [6] are widely used alternatives. These structures are binary trees where each node of the tree is assigned to a pair of pivot elements (p_1,p_2). If the distance of the object to p_1 is smaller than the distance to p_2, then the object is assigned to the left subtree, otherwise it is sent to the right subtree. According to authors, the GHT provides a better indexing structure than the usual vantage point trees [5].

Based on the survey of [1], the following methods are usually used for pivot selection. The most simple solution is the random selection of the pivot elements. In this approach, more tests are run and the pivot set with best parameter is selected. The second method is the incremental selection method. In the first step of this algorithm, a p_1 with optimal fitness is selected. In the next step, the pivot set is extended with p_2, yielded by a new optimization process where p_1 is fixed already. On this way, the pivot set is extended incrementally to the required size. The third way is the local optimization method. In this case, an initial pivot set is generated on some arbitrary way. In the next step, the pivot element with worst contribution is removed from the set and a new pivot element is selected into the set.

The work [1] analyzed the pivot selection methods from the viewpoint of subtree pruning operation. Usually a heuristic approach is used in the applications. The core elements of the heuristics are the following rules: the pivot elements should be far from the other not pivot elements and they should be far from each others too. The paper concluded that the incremental selection method provides the optimal solution of this heuristics.

An improved pivot selection method called Sparse Spatial Selection (SSS) is presented in [2]. The SSS method generates the pivot elements dynamically when a new outlier elements is inserted into the object pool. A new incoming element is selected as a new pivot if it is far enough from the other pivot elements. A loss minimization method was proposed by [8] where the loss is measured as the real distance between the object and its nearest neighbor in the index tree.

A conceptually different approach for object indexing is the family of computation methods based only on the distance matrix. In the AESA [4] algorithm, the distances between every pairs of objects are known and thus every objects can be considered as

a pivot element The method provides the best query results for small object sets but it can't be applied to larger sets because of the $O(N^2)$ number of distance computations.

The main goal of this paper is to analyze the pivot selection methods from a different viewpoint, namely from the viewpoint of balancing factor of the generated index tree. The balancing factor is an important parameter of the traditional search trees too. In the case of well-balanced tree, the cost of search operation is low and stable [9]. The another aspect of the investigation is the cost reduction of distance computations during the selection of pivot set. The proposed method optimizes the process of pivot selection to achieve a balanced GHT tree using a minimum number of distance calculations. The investigation addresses the split operation of a GHT node when the bucket gets full. In this process, two new pivot elements should be selected. As the bucket contains only the objects of a single node, it can be assumed that the size of the object set is limited. Based on these consideration, the presented method is a combination of algorithms on distance matrices and direct pivot selection.

2 Distance Matrices

Let $\mathcal{H} \subset \Re^{N\times N}$ denotes the set of distance matrices meeting the axioms of the distance functions. Let $\hat{d}$ denote the upper triangle part of $\hat{h}$ and $\mathcal{H}^u$ the set of these matrices. With corresponding mapping of indexes the formula (1) can be converted into the following form:

$$\begin{gathered} \forall\, d_{ij}, d_{jk}, d_{ki} \in \hat{d} \;\in\; \mathcal{H}^u \colon d_{ik} + d_{kj} - d_{ij} \geq 0 \\ \forall\, d_{ij} \in \hat{d} \in \mathcal{H}^u \colon d_{ij} \geq 0 \\ \mathcal{H}^u \subset \Re^{\binom{N}{2}}. \end{gathered} \tag{4}$$

As it can be seen the set of valid distance matrices is equal to the solution set of the linear homogenous inequality system (2). In this formula we allow to have a zero distance value between any objects. This difference enables the investigation of degenerate cases where two objects may be overlapped, i.e. they are the same object. It follows from this fact that if

$$\hat{x}, \hat{y} \in \mathcal{H}^u, \alpha, \beta \in \mathcal{R}^+$$

then

$$\alpha\hat{x} + \beta\hat{y} \in \mathcal{H}^u$$

is also met. Thus $\mathcal{H}^u$ is a convex cone in $\Re^{\binom{N}{2}}$ containing the zero element of $\Re^{\binom{N}{2}}$ too.

A ray of $\mathcal{H}^u$ for direction $\hat{d} \in \mathcal{H}^u$ is defined as

$$\alpha\hat{d} \in \mathcal{H}^u, \alpha \in \mathcal{R}^+$$

The direction $\hat{d}$ is an extreme direction of a convex cone if it cannot be expressed as a conic combination of directions of any rays in the cone distinct from it:

$$\forall \hat{a}, \hat{b} \in \mathcal{H}^u, \alpha, \beta, \gamma \in \mathcal{R}^+, \hat{a} \neq \gamma \hat{d}, \hat{b} \neq \gamma \hat{d}: \alpha \hat{a} + \beta \hat{b} \neq \hat{d} \tag{5}$$

According to the theory of Klee [10], any closed convex set containing no lines can be expressed as the convex hull of its extreme points and extreme rays.

Unfortunately, the extreme rays of the metric cone can't be generated directly for larger N values as the number of extreme rays is a $O(2^{N^2})$[11]. Thus, in the investigation, only a subset of the cone is considered, namely the subset of distance matrices corresponding to the points in Euclidean space. Thus the values in the generated matrices correspond to the Euclidean distances between the points.

In the test generations, three main types of distribution were selected: uni-polar, bi-polar and multi-polar distributions with uniform distribution within the clusters. As the experiences show the efficiency of the algorithms are significantly influenced by the characteristics of the distribution.

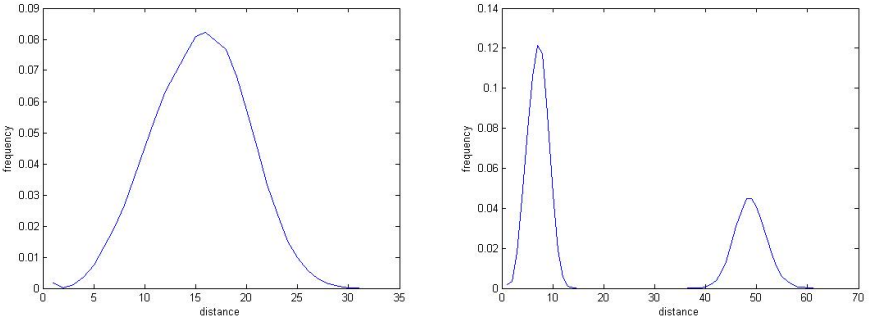


Fig. 1. Distance distributions of uni-polar and bi-polar cases

3 Selection of Pivot Elements

As the main goal of the investigation is to provide a well-balanced distribution of the objects, the efficiency criterion is measured with

$$\mu = 2 \cdot \frac{min\{|B_L|, |B_R|\}}{|B_L| + |B_R|} \tag{6}$$

where B_L and B_R denote the left and right side subtrees. The value domain of μ is [0..1]. In the optimal case, the value of μ is equal to 1. If μ = 0 then all objects are assigned to one of the child branches.

Based on the suggestions of [3], a multi-phase pivot selection method was implemented. In the first phase the usual heuristic step is applied: the object pair with largest intra-distance will be selected. In order to minimize the computation cost, only an approximation is performed in the followings steps:

- random selection of an object p_0
- selection of p_1 with $d(p_1,p_0) \rightarrow$ maximum
- selection of p_2 with $d(p_1,p_2) \rightarrow$ maximum

The object pair (p_1,p_2) is selected as initial pivots. Based on the experiences, the fitness of the random selection largely depends on the object distribution. In the case of uniform distribution it provides a relatively good result but if the distribution is bi-polar a poor results is yielded. To improve the efficiency a local optimization process is performed in the second phase. The main steps of this phase are the followings:

```
1: mumax = mu (p1,p2);
   selection of p3 where mu(p1,p3) is maximum;
   selection of p4 where mu(p4,p2) is maximum;
   mu = max (mu(p1,p3),mu(p4,p2));
   if mu > mumax then
     mumax = mu;
     replace the old pivot pair with the new one;
     go back to step 1
   else
     terminate the procedure;
   end;
```

In the tests, three algorithms were compared. The first algorithm is the brute force search where for every object pair (p_1,p_2) the $\mu(p_1,p_2)$ value is evaluated and the best pair is selected. This method provides a global optimum but it requires $O(N^3)$ operation as the computation of $\mu()$ belongs to the $O(N)$ complexity class. The second method is the random pivot selection method with maximum intra-distance criteria. The third method is the proposed local optimization algorithm. In the tests, two parameters were measured: the efficiency factor μ and the execution costs t.

The test results are summarized in Table 1. The first table (Table 1a) is for the uni-polar case, the second table (Table 1b) shows the bi-polar case with parameter value: 0.2/0.8. For the multi-polar cases, the results always lay between these values. The Fig. 1 shows the comparison of the quality μ values for the random and local search methods for bi-polar distribution.

It can be seen from the results that the random search method work weak in the case of bi-polar distribution and can work well in uni-polar case. The local optimum search method provides always a good result and it requires significant less time than the brute force algorithm.

Table 1. a

sample size	brute force		random		local optimization	
	μ	t(s)	μ	t(s)	μ	t(s)
300	1	0.910	0.89	0.001	1	0.018
400	1	2.312	0.93	0.001	0.99	0.026
500	1	4.321	0.93	0.001	1	0.054
600	1	7.916	0.95	0.001	1	0.081
700	1	12.532	0.94	0.001	1	0.120
800	1	19.166	0.91	0.001	0.98	0.167
900	1	27.331	0.93	0.001	0.99	0.188
1000	1	38.182	0.94	0.001	1	0.221
1200	1	63.529	0.93	0.001	1	0.340
1400	1	105.117	0.94	0.001	0.98	0.442

Table 2. b

sample size	brute force		random		local optimization	
	μ	t(s)	μ	t(s)	μ	t(s)
300	1	0.932	0.40	0.001	1	0.017
400	1	2.212	0.36	0.001	1	0.024
500	1	4.413	0.39	0.001	0.99	0.054
600	1	7.806	0.39	0.001	1	0.077
700	1	12.731	0.38	0.001	1	0.112
800	1	18.463	0.44	0.001	1	0.163
900	1	27.625	0.44	0.001	0.99	0.181
1000	1	37.458	0.42	0.001	1	0.218
1200	1	62.715	0.41	0.001	0.99	0.351
1400	1	103.563	0.41	0.001	1	0.438

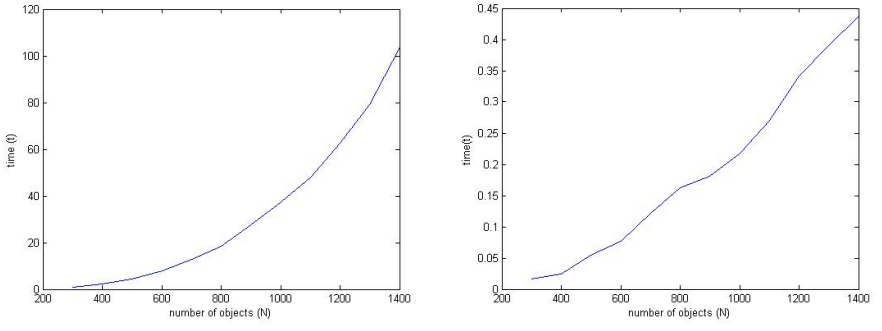


Fig. 2. , Time cost of the brute force and local optimum search methods

4 Improvements of the Local Optimization Method

It can be seen that the execution cost of the local optimization method shows a $O(N^2)$ characteristic. Thus some additional modules were included into the algorithm to reduce the cost value. The implemented reduction methods relates to the calculation of the μ value as this module has the largest cost portion within the pivot search algorithm.

The first optimization step relates to the reduction of candidate set in selection of p_3 and p_4. In the initial version, all objects belong to the candidate set. On the other hand, it follows from the triangle inequality that if

$$min_k\{|d(x_i, x_k) - d(x_j, x_k)|\} > d(x_i, x_j) \tag{7}$$

is satisfied then the partitioning sets of x_i and x_j are the same, i.e. the left (right) subtrees are the same for both elements. Thus if x_i is already tested and x_j meets the condition (7) then the testing of x_j can be omitted.

In the next table, the cost reduction factor of this elimination step is shown for different object distributions. As it can be seen this step is effective only if the

distribution is bi-polar. The reason of this experience is the fact: the smaller is the relative distance d_{ij} the higher is the chance that inequality (7) can be used for test elimination. In the case of bi-polar distribution the chance to have large distance differences is greater than in the case of uni-polar distribution.

Table 3.

distribution	reduction factor (in percent)	
	average	deviation
uni-polar	0.2%	0.04%
bi-polar	36.4%	5.4%

The second method for candidate set reduction is the application of sampling technique instead of full scan of the objects. In this method the μ is calculated with

$$\mu = 2 \cdot \frac{min\{|B'_L|, |B'_R|\}}{|B'_L| + |B'_R|} \tag{8}$$

where B' denotes subset generated by random sampling. The next table (Table 4) summarizes the achieved accuracy at different sample sizes (reduction levels). The table contains the accuracy error values in percentage.

The test data shows that sampling of the object distribution has some similarity with the standard theories of determining the optimal sample size for normal distributions. For example, the Cochran's formula [12] gives the sample size as

$$\gamma = \frac{t^2 \cdot s}{d^2} \tag{9}$$

where

t : value for selected alpha level for each tail
s: estimation for variance
d: acceptable margin of error

The formula of Krejcie [13] provides a different approach:

$$n = \frac{\chi^2 \cdot N}{d^2 \cdot 4 \cdot (N-1) + \chi^2} \tag{10}$$

where χ^2 denotes the Chi-square of the given confidence level. As both formulas show, the optimal sample size depends on many factors and its value changes only very slow for increase of N. For example the optimal sample size for N = 1000 lies between 210 and 270. In our experiment, the optimal sample size is about 140 for N = 1000.

In the computation of the μ fitness value, the distances from a given object x to both pivot objects p_1, p_2 are considered to check which pivot is closer to x. On the other hand, the distance value calculation can be omitted in some situations.

Table 4.

sample size	error for set A (N=200)		error for set B (N=1000)	
	average	deviation	average	deviation
sqrt(N)	42%	27%	19%	16%
2 sqrt(N)	30%	21%	17%	13%
4 sqrt(N)	15%	7%	7%	6%
8 sqrt(N)	13%	8%	7%	5%
16 sqrt(N)	-	-	7%	5%
24 sqrt(N)	-	-	6%	5%

Let p_1, p_2 denote the current pivot candidate objects. Let q denote the current object to be tested. The test returns 1 if q is close to p_1, otherwise it returns 2. It is assumed that exists a r object for which the distances $d(q,r)$ and $d(p_2,r)$ are already known. The distance $d(p_1,q)$ is known also. It follows from the triangle inequalities that

$$|d(p_2,r) - d(q,r)| \leq d(p_2,q) \leq |d(p_2,r) + d(q,r)| \tag{11}$$

Thus if

$$d(p_1,q) < |d(p_2,r) - d(q,r)|$$

then

$$d(p_1,q) < d(p_2,q)$$

If

$$d(p_1,q) > |d(p_2,r) + d(q,r)|$$

then

$$d(p_1,q) > d(p_2,q)$$

Thus in this cases, the object q can be assigned to the corresponding subset without calculating the actual $d(p_2,q)$ distance value.

5 Interval Model of Distance Calculations

The distance matrix contains $\binom{N}{2}$ distance values, thus the generation of the matrix is an $O(N^2)$ cost operation. As the calculation of the distance value for complex object is a high cost operation, the reduction of the redundant distance values is also an important optimizations step.

The first question of this research phase was to investigate how can the single distance values restricted by the other distance values of the same object set. As the distance value between two objects is constrained by the triangle inequalities of the metric function, the values already in the matrix will constrain the values not already

filled in. Every new value entered into the matrix will reduce the uncertainty on the still empty positions.

To measure the uncertainty of the unknown distance values, an interval model is introduced. Every distance value is given here with an interval, where the interval determines the interval of possible values. Thus initially, when no distance is known yet, every value is given with *[0,max_dist]* where *max_dist* denotes the largest possible value. If a distance value is set to value *v*, the corresponding interval contains only one element: *[v,v]*. To indicate the level of value uncertainty, an ϕ measure is introduced on the following way:

$$\phi_{avg} = \frac{\sum_i \phi_i}{N^2} \qquad \phi_i = v_{max} - v_{min} \tag{12}$$

If a new distance value v_{ij} is set for object pair (x_i,x_j) , then the interval values of the matrix elements are updated using the following algorithm:

```
d1 = min(tavok_max(i,j),tavok_max(j,k));
d2 = max(tavok_min(i,j),tavok_min(j,k));
if d1 < d2 then
  tavok_min(i,k) = max(d2 - d1,tavok_min(i,k));
  tavok_min(k,i) = tavok_min(i,k);
end
d1 = tavok_max(i,j)+tavok_max(j,k);
tavok_max(i,k) = min(tavok_max(i,k),d1);
tavok_max(k,i) = tavok_max(i,k);
```

In the algorithm, *tavok_min* denotes the array of low boundary values and *tavok_max* stores the upper boundary values. These rules are based on the following inequality derived from the triangle inequality:

$$|d(x,r) - d(y,r)| \leq d(x,y) \leq |d(x,r) + d(y,r)| \tag{12}$$

In the tests, the average ϕ and the minimum ϕ values were investigated during the generation of the distance matrix. It is clear the more values are set the less is the uncertainty. The figures Fig. 4 - 5 show the average ϕ value for increasing number of set values (x-axis) for both the uni-polar and the bi-polar object distributions. As the figure demonstrates in the case of bi-polar distribution the average uncertainty is less than in the case of uni-polar distribution.

The Fig 3 shows the minimum, not zero ϕ values of the matrix. Based on these results, it can be seen that the smallest value interval are equal to some percents of the average distance value. Thus, if a given level of uncertainty is allowed, some of the distance calculations can be eliminated.

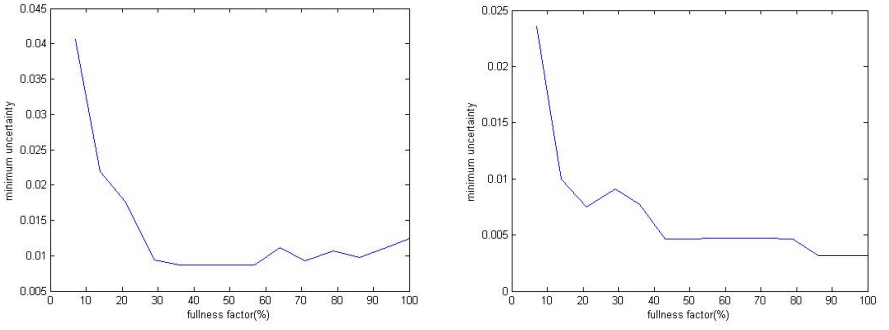


Fig. 3. Minimum uncertainty values for uni-polar and bi-polar distributions

For the case when the distance values are stored with interval values a new definition of the μ fitness measure is introduced. In this approach, an object may belong to both sides with given certainty. Let p_1, p_2 denote the pivot objects and q denotes the current object to be tested. Let *E1* denote the event that q is closer to p_1 than to p_2 and *E2* is the event that q is closer to p_2 than to p_1. If the $d(q,p_1)$ distance has the value $[va_1, vb_1]$ and $d(q,p_2)$ is equal to $[va_2,vb_2]$, then the probability of *E1* and of *E2* can be calculated with the method of geometric probability. The area of valid value pairs is a rectangle with sides corresponding to the intervals. The set of value pairs belonging to *E2* is a half-plane upper the line $y = x$.

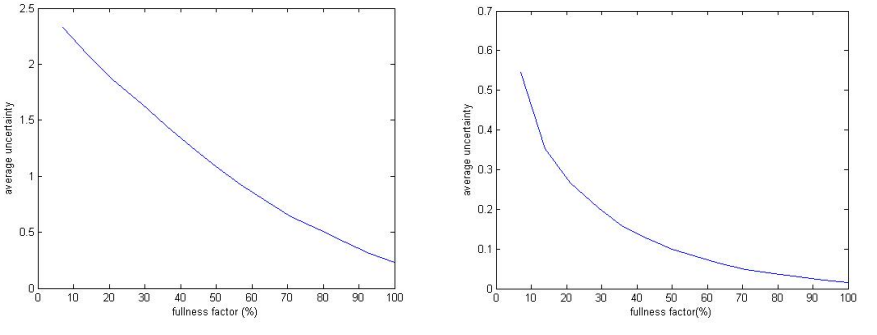


Fig. 4. Average uncertainty values for uni-polar distribution without interval adjustment and with adjustment

Let $p_i(E1)$ denote the probability that the i-th object belongs to the area of p_1. The $p_i(E2)$ is defined on similar way. It can be easily verified that

$$p(E1) + p(E2) = 1$$

for every object.

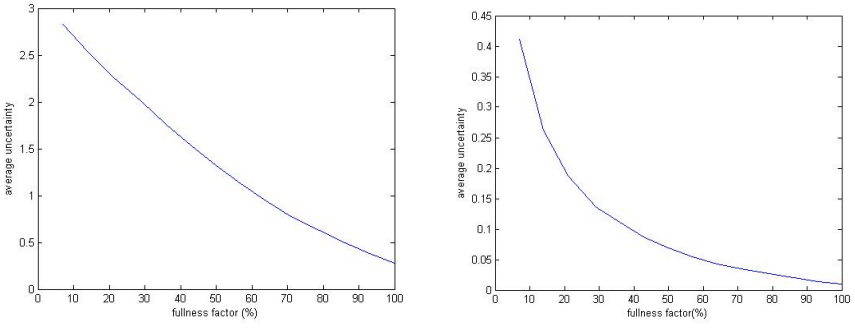


Fig. 5. Average uncertainty values for bi-polar distribution without interval adjustment and with adjustment

Based on the previous definitions, the μ value is calculated with

$$\mu = 2 \cdot \frac{min\{P_1, P_2\}}{P_1 + P_2}$$
$$P_1 = \sum_{i=1}^{N} p_i(E_1) \tag{12}$$
$$P_2 = \sum_{i=1}^{N} p_i(E_2)$$

The redefined fitness function is a generalization of the base fitness function as it yields the same value for the strict cases when $p_i(E1)$ or $p_i(E2)$ is equal to 1. Using the redefined fitness function, the presented pivot selection algorithm can be executed on the interval-based distance matrix too.

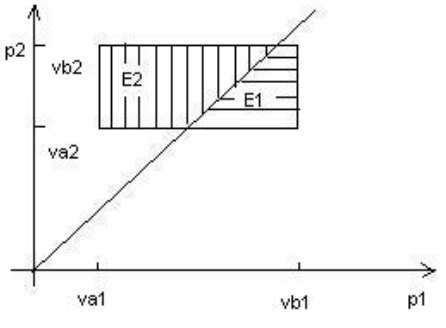


Fig. 6. Geometric probability of subtree assignment

The next table summarizes the test results for the interval-based matrix and it shows the comparison between the strict-valued and the interval-valued matrix approaches.

Table 5.

proportion of known distances	fitness of random selection	fitness of IV selection
0.01%	.65	0.65
3%	.66	0.81
16%	.62	0.96

As the result table shows, the efficiency of the interval-valued (IV) method depends on the proportion of the known distances that means on the uncertainty of the distance matrix. It yields in good fitness value if the uncertainty of the matrix is low. The exact and formal analysis of this relationship is the goal of further investigations. In the tests, a 16% covering rate resulted in a well balanced tree with a 0.96 balancing factor.

6 Conclusions

The paper presented a detailed analysis of optimal pivot selection in general metric space from the viewpoint of index tree balancing. The analysis focused on the GHT index tree assuming that an index tree node contains a moderate number of objects. In the investigation, two main object distributions were tested: the uni-polar and the bi-polar distributions. The paper proposes a combined heuristic and local search optimization method for selection of pivot objects. For reduction of the search algorithm, some novel optimization methods were introduced. One of the cost reduction methods refers to eliminating of object tests within the calculation of balancing factor. Another important goal is to reduce the number of distance calculations in the object set. The dependencies between the distance values are analyzed in order to eliminate the redundant distance values. Another important reduction method is the application of interval values instead of strict values in order to manage the uncertainty of the distance values. The performed analysis and tests show that a the proposed modification improve the efficiency of the standard methods significantly.

Acknowledgements. This research was carried out as part of the TAMOP 4.2.1.B-10/2/KONV-2010-0001 project with support by the European Union, co-financed by the European Social Fund.

References

1. Bustos, B., Navarro, G., Chavez, E.: Pivot selection Techniques for Proximity Searching in Metric Spaces. Journal Pattern Recognition Letters, 2357–2366 (2003)
2. Bustos, B., Pedreira, O., Brisaboa, N.: A Dynamic Pivot Selection Technique for Similarity Search. In: Proceedings of ICDE Workshop, pp. 394–400 (2008)

3. Veltkamp, R., van Leuken, R., Typke, R.: Selecting vantage objects for similarity indexing. ACM Transactions on Multimedia Computing, Communications and Applications 7(3(16)) (2011)
4. Mico, L., Oncina, J., Vidal, E.: A new version of the nearest neighbor approximating and eliminating search with linear preprocessing time and memory requirements. Pattern Recognition Letters 15(1), 9–17 (1994)
5. Uhlmann, K.: Satisfying general proximity similarity queries with metric trees. Information Processing Letters (40), 175–179 (1991)
6. Kalantari, I., McDonald, G.: A data structure and an algorithm for the nearest point problem. IEEE Transactions on Software Engineering 9(5), 631–634 (1983)
7. Batko, M., Gennaro, C., Zezula, P.: Similarity Grid for Searching in Metric Spaces. In: Türker, C., Agosti, M., Schek, H.-J. (eds.) Peer-to-Peer, Grid, and Service-Orientation in Digital Library Architectures. LNCS, vol. 3664, pp. 25–44. Springer, Heidelberg (2005)
8. Henning, C., Latecki, J.: The choice of vantage objects for image retrieval. Pattern Recognition 36(9), 2187–2196 (2003)
9. Sprugnoli, R.: Randomly balanced binary trees. Calcolo 17, 99–117 (1981)
10. Tamura, K.: A Method for Constructing the Polar Cone of a Polyhedral Cone, with Applications to Linear Multicriteria Decision Problems. Journal of Optimization Theory and Applications 19, 547–564 (1976)
11. Versik, A.: Distance matrices, random metrics and Urysohn space, arXiv:math/023008v1, MPI-2002-8 (2002)
12. Bartlett, J., Kotrlik, J., Higgins, C.: Organizational Research: Determining Appropriate Sample Size in Survey Research. Information Technology Learning and Performance 19, 43–51 (2001)
13. Krejcie, R., Morgan, D.: Determining sample size for research activities. Educational and Psychological Measurement 30, 607–610 (1970)

Hot Deck Methods for Imputing Missing Data

The Effects of Limiting Donor Usage

Dieter William Joenssen and Udo Bankhofer

Technische Universität Ilmenau, Fachgebiet für Quantitative Methoden, Ilmenau, Germany
{Dieter.Joenssen,Udo.Bankhofer}@TU-Ilmenau.de

Abstract. Missing data methods, within the data mining context, are limited in computational complexity due to large data amounts. Amongst the computationally simple yet effective imputation methods are the hot deck procedures. Hot deck methods impute missing values within a data matrix by using available values from the same matrix. The object, from which these available values are taken for imputation within another, is called the donor. The replication of values leads to the problem, that a single donor might be selected to accommodate multiple recipients. The inherent risk posed by this is that too many, or even all, missing values may be imputed with the values from a single donor. To mitigate this risk, some hot deck variants limit the amount of times any one donor may be selected for donating its values. This inevitably leads to the question under which conditions such a limitation is sensible. This study aims to answer this question though an extensive simulation. The results show rather clear differences between imputations by hot deck methods in which the donor limit was varied. In addition to these differences, influencing factors are identified that determine whether or not a donor limit is sensible.

Keywords: Hot Deck Imputation, Missing Data, Nonresponse, Imputation, Simulation.

1 Introduction

Missing data is a prevalent problem in many real empirical investigations. Missing data's sources include failures in either manual or automated data collection, where some values are recorded while others are not. Missing data, however, may also be induced through manual or automatic data editing, such as outlier removal [1]. Missing data that cannot be resolved though manual or automatic logic inference, for example when data can be inferred from existing data (e.g. a missing passport number when the respondent has no passport), must be resolved in light of the missingness mechanism.

Rubin [2] first treated missing data indicators as random variables. Based on the indicators' distribution, he defined three basic mechanisms MCAR, MAR, and NMAR. With MCAR (missing completely at random), missingness is independent of any data values, missing or observed. Thus under MCAR, observed data represents a subsample of the intended sample. Under MAR (missing at random), whether or not

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 63–75, 2012.

data is missing depends on some observed data's values. Finally under NMAR (not missing at random), the missing data is dependent on the missing data's values.

With missingness present, conventional methods cannot be simply applied to the data without proxy. Explicit provisions must be made before or within the analysis. The provisions, to deal with the missing data, must be chosen based on the identified missingness mechanism. Principally, two strategies to deal with missing data in the data mining context are appropriate: elimination and imputation. Elimination procedures will eliminate objects or attributes with missing data from the analysis. These only lead to a data set, from which accurate inferences may be made, if the missingness mechanism is correctly identified as MCAR. But even if the mechanism is MCAR, eliminating records with missing values denotes an inferior strategy, especially when many records need to be eliminated due to unfavorable missingness patterns or data collection schemes (e.g. asynchronous sampling). Imputation methods replace missing values with estimates [3-4], and can be suited under the less stringent assumptions of MAR. Some techniques can even lead to correct inferences under the nonignorable NMAR mechanism [5-6]. Replacing missing values with reasonable ones not only assures all information gathered can be used, but also broadens the spectrum of available analyses. Imputation methods differ in how they define these reasonable values. The simplest imputation techniques replace missing values with eligible location parameters. Beyond that, multivariate methods, such as regression or classification methods, may be used to identify imputation values. The interested reader may find more complete descriptions of missingness mechanisms and methods of dealing with missing data in [5-7].

An imputation technique category appropriate for imputation in the context of mining large amounts of data and large surveys, due to its computational simplicity [1], [7-8], is hot deck imputation. Ford [12] defines a hot deck procedure as one where missing items are replaced by using values from one or more similar records within the same classification group. Partitioning records into disjoint, homogeneous groups is done so selected "good" records supplying imputation values (the donors) follow the same distribution as the "bad" records (the recipients). Due to this, and the replication property, all hot deck imputed data sets contain only plausible values, which cannot be guaranteed by most other methods. Traditionally, a donor is chosen at random, but other methods such as ordering by covariate when sequentially imputing records or nearest neighbor techniques utilizing distance metrics are possible to improve estimates at computational simplicities' expense [6],[9].

The replication of values leads to central problem in question here. Any donor may, fundamentally, be chosen to accommodate multiple recipients. This poses the inherent risk that "too many" or even all missing values are imputed with the same value or values from a single donor. Due to this, some variants of hot deck procedures limit the amount of times any one donor may be selected for donating its values. This inevitably leads to question under which conditions a limitation is sensible and whether or not some appropriate limit value exists. This study aims to answer these questions. Chapter 2 discusses current empirical and theoretical research on this topic. Chapter 3 highlights the simulation study design while results are reported and discussed in chapter 4. A conclusion is presented and possibilities for further research are presented in chapter 5.

2 Review of Literature

Limiting the times a donor may be used was first investigated by Kalton and Kish [10]. Based combinatorics, they come to the conclusion that choosing a donor from the pool of available donors without replacement leads to a reduction in the imputation variance, the precision with which any parameter is estimated from the post imputation data matrix. Two possible situations arising from a hot deck imputation also favor the donor limit implementation. First, the risk of exclusively using one donor for all imputations is removed [11]. Second, the probability of using one donor with an extreme value, or extreme values too often, is reduced [5], [12]. Andridge und Little [13] argue against a donor limit implementation. They contend that imposing a donor limit inherently reduces the ability to choose the most similar, and therefore most appropriate, donor for imputation. Not limiting the times a donor can be chosen may increase data quality. Thus, from a theoretical point of view, it is not clear whether or not a donor limit has a positive or negative impact on the post-imputation's data quality.

Literature on this subject provides only studies that compare hot deck imputation methods with other imputation methods. These studies include either only drawing the donor from the donor pool with replacement [14-16] or without replacement [17]. Whether the donor is drawn with or without replacement is also relevant in the context of post imputation variance estimation [9], [18]. These two cases are important or at least considered for deriving the estimation formulas [19]. Literature reviewing the mechanics of hot deck and other imputation methods includes [6-7], [20-21].

Based on this review of literature it becomes apparent, that the consequences of limiting the usage of an object as donor have not been sufficiently examined. Missing are especially recommendations as to under which circumstances a donor limit is sensible.

3 Simulation Design

This work aims to investigate what impact imposing a donor limit, in various forms of hot deck methods, has on the imputed data matrix. To guide this, research questions are formulated in the next section. In succession, factors that are believed to have an impact on the results, and therefore are varied in a factorial design, are described in section 3.2. The chosen quality criterion is outlined in section 3.3 while section 3.4 discusses further details of the simulation.

3.1 Research Questions

Considering possible, theoretical effects, that repeated donor usage might have, the following four research questions will be answered with the simulation study:

1. Is a limitation on donor usage reasonable, or even essential, for successful hot deck imputation?
2. What criteria dictate the necessity to limit donor usage?

3. Is a necessity to limit donor usage independent of the hot deck variant used?
4. Can recommendations be made regarding a maximum possible donor usage based on information available before imputation?

As the research questions show, an initial analysis of whether or not a donor limitation has any influence on the post imputation analysis of the data. Provided that there is an influence, this should be analyzed differentiated by the factors varied, especially the chosen hot deck method. Beyond this, recommendations must be extracted.

3.2 Factorial Design

By considering papers where authors chose similar approaches [15], [12], [22-23] and further deliberations, a series of factors are identified that might have an influence on whether or not a donor limit influences imputation results. These factors are systematically varied in a complete factorial design and their effects are analyzed. The following factors are considered:

- **Data matrix dimension:** (100x9), (350x9), (500x9), and (1750x9) as (*nxm*) data matrices with *n* objects and *m* attributes are considered and the amount of objects as a function of the amount of imputation classes.
- **Variable scale:** Data matrices are of mixed scale variables with each 3 binary, ordinal and quantitative attributes. Binary variables are chosen, as any nominal variable can be represented in a number of binary variables. Ordinal variables are considered separately, either all on a five or seven point scale. The quantitative variables are chosen to be normally distributed with only nonnegative numbers occurring.
- **Imputation class count:** Imputation classes are assumed to be given prior to imputation and data is generated accordingly. Factor levels are two and seven imputation classes.
- **Object count per imputation class:** The amount of objects characterizing each imputation class is varied. Factor levels 50 and 250 objects per class are considered.
- **Class structure:** To differentiate between well- and ill-chosen imputation classes, data is generated with a relatively strong and relatively weak class structure. Strong class structure is achieved by having classes overlap by 5% and inner-class correlation of 0.5. Weak class structure is achieved by an intra-class overlap of 30% and no inner-class correlation.
- **Portion of missing data:** Factor levels include 5, 10, and 20% missing data points. Every object is assured to have at least one data point available (no subject nonresponse).
- **Missingness mechanism:** Two unsystematic mechanisms, MCAR and MAR, are considered as well as NMAR.
- **Hot deck methods:** Three sequential and three simultaneous methods of imputation are considered. Sequential or simultaneous refers to whether or not the variables are imputed simultaneously or sequentially. With simultaneous imputation,

all values missing come from one donor complete in all variables. With sequential imputation, a missing value is imputed from a donor that has a value available for that variable but is not necessarily complete in all other variables. These two groups can each be differentiated into three methods. First is the case that a donor is chosen at random (SeqR and SimR) the other two are distance based methods. The distance based methods differ in the way missing values are treated. The first method weights the pairwise available distances (SeqDW and SimDW), the second computes distances after performing a mean imputation on the data matrix (SeqDM and SimDM). To account for variability and importance, prior to aggregating the Manhattan distances, variables are weighted with the inverse of their range.

Next to the previously mentioned factors, different donor limits are considered. Alongside the two extreme cases, a donor limit of one and no donor usage limitation, two further levels of donor limitation are considered. The four considered factor levels are as follows:

- A donor is only allowed as such once
- A donor is only allowed to be chosen, at most, for 25% of all recipients
- A donor is only allowed to be chosen, at most, for 50% of all recipients
- A donor is allowed to be chosen for all recipients

3.3 Quality Criteria

To evaluate imputation quality, location and/or variability measures for each variable scale type are considered [23]. Following parameters are calculated dependent on the variable's scale:

- **Binary variable:** relative frequency
- **Ordinal variable:** median, quartile distance
- **Quantitative variable:** mean, variance

In order to rank the differences in performance, the mean absolute relative deviation $\Delta\overline{p}$ between the true value of each parameter, $p_{w,}$ and each post imputation estimate, $p_{i,}$ are calculated for each donor limit factor level:

$$\Delta\overline{p} = \sum \left| \frac{p_i - p_w}{p_w} \right| \tag{1}$$

Differences in the $\Delta\overline{p}$'s are compared, but due to the large amounts of data that is generated in this simulation, statistical significance tests are not appropriate. As an alternative to this, Cohen's d measure of effect [24-25] is chosen. It corresponds to the t-statistic without the sample size influence. The calculation of Cohen's d for this case is as follows:

$$d = \frac{\Delta\overline{p}_1 - \Delta\overline{p}_2}{\sqrt{\frac{s_1^2 - s_2^2}{2}}} \tag{2}$$

$\Delta\overline{p}_1$ and $\Delta\overline{p}_2$ are calculated via (1) for two different donor limits. s_1^2 and s_2^2 are the corresponding variances in the relative deviations. The usage of absolute values for $\Delta\overline{p}_1$ and $\Delta\overline{p}_2$ allows the interpretation of the sign of d. A positive sign means that the second case of donor limitation performed better than the first, while a negative sign indicates the first case is superior in estimating the relevant parameter. Cohen [24] does not offer a single critical value above which an effect is nontrivial. However, he denotes effects around 0.2 as being small and presents tables for effect values starting at 0.1, which Fröhlich und Pieter [26] also deem as critical.

3.4 Simulation Details

100 data matrices are simulated for every factor level combination of "imputation class count", "object count per imputation class", "class structure", and "ordinal variable's scale". For every complete data matrix, the true parameters are computed. Each of these 1600 data matrices is then subjugated to each missingness mechanism generating three different amounts of missing data. All of the matrices with missing data are then imputed by all of the six hot deck methods. This creates 3.456 million imputed data matrices for which each parameter is once again is calculated, for which the deviations from the true values are evaluated as stated above.

The missing data is generated as follows: under MCAR a set amount of values are chosen without replacement to be missing. Under MAR, missing data is generated MCAR using two different rates based on one binary variable's value, which is not subject to missingness. The different rates of missingness are either 10% higher or lower than the rates under MCAR. NMAR modifies the MAR mechanism to also allow missingness in the binary variable.

Further, some limits in generating missingness were instituted. To forgo possible problems with the simultaneous imputation methods and the donor limitation to one, it was guaranteed that at least 50% of all objects within one class were complete in all attributes. Further all objects had at minimum one available value.

4 Results

Based on the simulation's results, the formulated research questions are now answered. The first section answers the question on whether a limit on donor usage is reasonable. The following section 4.2 analyzes the factors and how they influence whether or not a donor limit is advantageous. The final section 4.3 reviews the recommendations that can be made pertaining to the levels of donor limitation.

4.1 Donor Limitation Impact

To evaluate whether differences between donor limitation levels are, in principle, possible, effect sizes between the cases of no and most stringent donor limitation are considered. For each parameter, as described in section 3.3, the median effects as well as different variability measures for all effects are available in table 1.

Table 1. Effects' median and variability

	Quantitative variables		Ordinal variables		Binary variables
	Mean	Variance	Median	Quartile difference	Relative frequency
Median	-0,001	-0,009	-0,002	-0,008	-0,007
Range	0,104	2,468	0,171	2,231	3,333
Distance 90%/10%-quantiles	0,031	0,336	0,037	0,262	0,251
Quartile difference	0,013	0,068	0,016	0,050	0,045
Standard deviation	0,013	0,280	0,017	0,249	0,325

The first conclusion that is reached, in light of these values, is that the effect sizes are neither all negative nor all positive. This means that neither a donor usage limitation of one or no donor usage limitation always leads to best results. Second conclusion is that there are no nontrivial effects for the mean and median of the quantitative and the ordinal variables, respectively. Both parameters exhibit small measures of variability and medians near zero. In contrast, some nontrivial effects are expected in all other parameters as the variability is comparatively large. Large effects are expected to be rare as the quartile difference is rather small in comparison to the range and 90%/10% quantile distance. Thus one can conclude that a donor limitation has influence on the quality of imputation, and that cases where the influence is large can be extracted.

The theoretical reduction in imputation variance through donor selection without replacement, as put forth by Kalton and Kish [5], is also investigated empirically at this point. The following table 2 shows the relative frequency (in percent) in how many cases a certain donor limit leads to the smallest variance in the parameter estimate.

Table 2. Frequency distribution of minimum imputation variance

Evaluated parameter		Donor usage limitation			
		Once	25%	50%	Unlimited
Quantitative variable	Mean	68,52%	15,47%	7,95%	8,06%
	Variance	67,25%	15,74%	8,56%	8,45%
Ordinal variable	Median	74,54%	11,38%	7,62%	6,46%
	Quartile distance	85,88%	5,71%	4,96%	3,45%
Binary variable	Relative frequency	78,36%	8,41%	6,96%	6,27%

Clearly, limiting donor usage to once leads to mimimal imputation variance for most cases and thus can be expected to lead to highest precision in parameter estimation. This holds even more so for the binary and ordinal variables. Nonetheless, in certain situations other donor limitations, or no limitation of donor usage, lead to minimum imputation variance.

4.2 Analysis of Donor Limit Influencing Factors

To analyze which factors varied in this study have an influence on whether or not a donor limitation is advantageous, Cohen's *d* is used. Again the effect sizes contrasting the two extreme cases are investigated. Thus negative values mean a maximum donor usage of one is superior to no donor usage limitation, while positive values signify the converse. The following section first highlights possible main effects followed by possible between factor effects on a donor limit advantage.

Analysis of Main Effects

Table 3 (below) shows a cross classification between all factors and factor levels with all parameters that show meaningful effect sizes. Effect sizes larger than the chosen critical effect of 0.1 are in bold.

Upon investigating the results, the first conclusion that is reached, is that, independent of any chosen factors, there are no meaningful differences between using a donor limit and using no donor limit in mean and median estimation. This result is congruent with the previous section's results. In contrast to this, parameters measuring variability are more heavily influenced through the variation of the chosen factors. Especially data matrices with a high proportion missing data, as well as those imputed with SimDM profit significantly from a donor limitation. Also a high amount of imputation classes speaks for a limit on donor usage.

The effects the data matrix's dimensions and the object amount per imputation class have are ambiguous. Class structure and usage any of the random hot deck procedures or SeqDW have no influence on whether a donor limit is advantageous. Fairly conspicuous is the fact that SimDW leads to partially positive effect sizes meaning that leaving donor usage unlimited is advantageous. This might lead to interesting higher order effects.

Analysis of Interactions

Based on the findings in the previous section, all effects for the parameters variance, quartile distance and relative frequency of the quantitative, ordinal and binary variables respectively, stratified by the hot deck methods SimDW, SimDM and SeqDM, are investigated for all other factors' levels. These values are shown in table 4 (below), with again values above 0.1 marked in bold.

As in the analysis of main effects, the table clearly shows that using SimDW in combination with no donor limit is advantageous. All combinations with other factors, with one exception, show positive values, even though only variance and relative frequency exhibit nontrivial effects. Furthermore, the other two methods, SimDM and SeqDM, show only negative values. Thus, the advantage of limiting donor usage is strongly dependent upon the imputation method used.

Table 3. Effect sizes for each factor

		Quantitative variables		Ordinal variables		Binary variables
		Mean	Variance	Median	Quartile difference	Relative frequency
Data matrix dimension	(100x9)	0,000	-0,082	-0,001	-0,030	-0,034
	(350x9)	0,000	**-0,177**	-0,005	**-0,152**	-0,022
	(500x9)	0,000	-0,064	-0,004	-0,030	**-0,130**
	(1750x9)	0,001	**-0,146**	-0,006	-0,065	**-0,162**
Imputation class count	2	0,000	-0,068	-0,001	-0,029	-0,072
	7	0,000	**-0,147**	-0,003	**-0,115**	-0,090
Object count per imputation class	50	0,000	**-0,112**	-0,001	-0,073	-0,028
	250	0,000	-0,090	-0,005	-0,041	**-0,141**
Class structure	Strong	0,000	-0,092	-0,001	-0,072	-0,072
	Weak	0,000	-0,094	-0,001	-0,045	-0,080
Portion of missing data	5%	0,000	-0,025	0,000	-0,013	-0,011
	10%	0,000	-0,071	0,000	-0,037	-0,051
	20%	0,000	**-0,148**	0,000	**-0,100**	**-0,129**
Missingness mechanism	MCAR	0,001	-0,088	-0,001	-0,053	-0,065
	MAR	0,000	**-0,100**	0,000	-0,066	-0,086
	NMAR	0,001	-0,091	0,000	-0,058	-0,077
Hot deck method	SimDW	-0,001	**0,153**	-0,002	0,025	0,075
	SimDM	-0,004	**-0,339**	0,005	**-0,214**	**-0,338**
	SeqDW	0,001	-0,007	-0,003	0,000	-0,005
	SeqDM	0,000	-0,088	0,010	**-0,133**	-0,041
	SimR	0,000	-0,001	-0,001	-0,004	0,000
	SeqR	0,000	-0,001	0,000	-0,001	-0,003

For all three portrayed methods, a high amount of imputation classes and a high percentage of missing data show meaningful effects, indicating an advantage of either selecting the donor with or without replacement. The amount of objects per imputation class show no homogeneous effect on the parameters, rather it seems to strengthen the advantage the donor limitation or non-limitation has, with the parameter evaluated for the binary variable reacting inversely to variance and quartile distance. The other factors seemingly don't influence the effects as their variation does not lead to great differences in the effects sizes, making their absolute level only dependent on the variable's scale or imputation method.

Table 4. Interactions between imputation method and other factors
(Legend: V = variance, Q = quartile distance, R = relative frequency)

		SimDW		SimDM			SeqDM	
		V	R	V	Q	R	V	Q
Data matrix dimension	(100x9)	**0,140**	0,058	**-0,337**	**-0,192**	**-0,216**	-0,089	**-0,139**
	(350x9)	**0,235**	0,055	**-0,473**	**-0,333**	**-0,278**	**-0,120**	**-0,207**
	(500x9)	**0,120**	**0,111**	**-0,283**	**-0,116**	**-0,492**	-0,077	-0,064
	(1750x9)	**0,215**	**0,108**	**-0,420**	**-0,257**	**-0,554**	**-0,109**	**-0,132**
Imputation class count	2	0,097	0,081	**-0,247**	**-0,101**	**-0,300**	-0,066	-0,082
	7	**0,287**	0,075	**-0,521**	**-0,382**	**-0,424**	**-0,130**	**-0,217**
Object count per class	50	**0,182**	0,034	**-0,426**	**-0,284**	**-0,132**	**-0,111**	**-0,196**
	250	**0,143**	**0,140**	**-0,319**	**-0,131**	**-0,684**	-0,088	-0,047
Class structure	Strong	**0,153**	0,078	**-0,339**	**-0,156**	**-0,362**	-0,091	**-0,135**
	Weak	**0,144**	0,071	**-0,338**	**-0,269**	**-0,313**	-0,085	**-0,132**
Portion of missing data	5%	0,065	0,031	-0,084	-0,057	-0,045	-0,013	-0,028
	10%	**0,148**	0,077	**-0,262**	**-0,162**	**-0,213**	-0,039	-0,073
	20%	**0,203**	**0,101**	**-0,558**	**-0,345**	**-0,600**	**-0,168**	**-0,233**
Missingness mechanism	MAR	**0,151**	0,079	**-0,355**	**-0,226**	**-0,372**	**-0,107**	**-0,152**
	MCAR	**0,153**	0,067	**-0,326**	**-0,204**	**-0,296**	-0,075	**-0,119**
	NMAR	**0,154**	0,077	**-0,334**	**-0,213**	**-0,344**	-0,081	**-0,125**

Besides these meaningful effects differentiated by hot deck method, there are also some interactions higher order, that lead to some strikingly large effects. For example, the factor level combination: 20% missing data, high amounts of imputation classes, and a low amount of objects per imputation class lead to effects up to -1.7 in variance and up to -1.9 in the quartile distance for SimDM. While effect sizes up to -3 are calculated for the relative frequency in the binary variable when the amount of imputation classes is large, has many objects in each class and many values are missing. This signifies some large advantage for donor selection without replacement when using SimDM. On the other hand, when using SimDW the largest effects are calculated when the amount of classes is high, but the amount of objects is low while having a high rate of missingness. Even though this only leads to effects of up to 0.6 and 0.34 for variance and quartile difference respectively, the effect is noticeable and relevant for donor selection with replacement. Conspicuous none the less is the fact that especially the combination of hot deck variant, amount of imputation classes, objects per imputation class, and portion of missing data lead to strong effects indicating strong advantages for and against donor limitation. Finally, the analysis of higher order interactions confirm either advantage for donor selection with or without replacement found in the lower order interactions.

4.3 Analysis of Donor Limitation Advantages

So far the investigation was limited to under which circumstances, choosing a donor with or without replacement, was advantageous for parameter estimation. The investigation is now expanded to include the two dynamic donor limitation cases. For this, the frequency with which a certain donor limitation yields the best parameter estimation is calculated and shown in table 5.

Table 5. Frequency distribution: smallest deviation between estimated and true parameter

Evaluated parameter		Donor usage limitation			
		Once	25%	50%	Unlimited
Quantitative variable	Mean	42,71%	20,22%	18,48%	18,60%
	Variance	54,05%	17,79%	13,04%	15,12%
Ordinal variable	Median	46,41%	21,53%	14,47%	17,59%
	Quartile distance	56,83%	16,24%	12,94%	13,99%
Binary variable	Relative frequency	49,42%	18,94%	15,07%	16,57%

The table shows that in most cases donor selection without replacement leads to the best parameter estimation. Again variability measures are more strongly affected. The table shows, for all parameters, that the frequency first decreases with a more lenient donor limit and then increases again with unlimited donor usage. This once again reveals the situation dependent nature of advantages offered by donor limitation.

These findings show, in summary, that there remains the possibility for optimizing the precise donor limit instituted. A further analysis of relationships between factors and to make precise recommendations is not reasonable at this point owed to the fact, that only 4 levels of donor limitation are available. Findings clearly indicate that further research along this path is worthwhile.

5 Conclusions

The simulation conducted show distinct differences between hot deck imputation procedures that make use of donor usage limitations. Limiting donor usage is not advantages under all circumstances, as, under some situations, allowing for unlimited donor usage leads to the best parameter estimates.

Under some situations, donor limitation leads to better parameter estimations. Splitting the data into a low amount of imputation classes leads to better estimation of variance and quartile distance for quantitative and ordinal variables, respectively. For low amounts of objects per imputation class the variance of quantitative variables is estimated better with a donor limitation, while binary variables with many objects per imputation class also profit from a donor limit. This is also the case for data matrices

with high amounts of missingness. Estimation of location, such as mean and median are not influenced by limiting donor usage.

Next to the data's properties, the hot deck variant used to impute missing data plays an important role as to whether or not limiting donor usage is an advantage. Dependent on the imputation method chosen, the limitation of donor usage is either advantageous, disadvantageous or without significant effect on parameter estimation. Both random hot decks and SeqDW are unaffected by any donor limit. Contrary to this, both SimDM and SeqDM perform better with donor limitation, while SimDW performs better without a limit on donor usage.

Even though, in most cases, allowing a donor to be used only once leads to the best parameter estimates, there are situations under which less restrictive donor limits or no donor limit is advantageous to parameter estimation. Thus developing recommendations for specific situation dependent donor limits is reasonable and a detailed investigation of the underlying interactions between the factors is an interesting point for future research.

References

1. Pearson, R.: Mining Imperfect Data. Society for Industrial and Applied Mathematics, Philadelphia (2005)
2. Rubin, D.B.: Inference and Missing Data (with discussion). Biometrika 63, 581–592 (1976)
3. Kim, J.O., Curry, J.: The Treatment of Missing Data in Multivariate Analysis. Sociological Methods and Research 6, 215–240 (1977)
4. Allison, P.D.: Missing Data, Sage University Papers Series on Quantitative Applications in the Social Sciences, Thousand Oaks (2001)
5. Bankhofer, U.: Unvollständige Daten- und Distanzmatrizen in der Multivariaten Datenanalyse, Eul, Bergisch Gladbach (1995)
6. Little, R.J., Rubin, D.B.: Statistical Analysis with Missing Data. Wiley, New York (1987)
7. Kalton, G., Kasprzyk, D.: Imputing for Missing Survey Responses. In: Proceedings of the Section on Survey Research Methods, pp. 22–31. American Statistical Association (1982)
8. Marker, D.A., Judkins, D.R., Winglee, M.: Large-Scale Imputation for Complex Surveys. In: Groves, R.M., Dillman, D.A., Eltinge, J.L., Little, R.J.A. (eds.) Survey Nonresponse, pp. 329–341. John Wiley & Sons, New York (2001)
9. Ford, B.: An Overview of Hot Deck Procedures. In: Madow, W., Nisselson, H., Olkin, I. (eds.) Incomplete Data in Sample Surveys, Theory and Bibliographies, 2, pp. 185–207. Academic Press (1983)
10. Kalton, G., Kish, L.: Two Efficient Random Imputation Procedures. In: Proceedings of the Survey Research Methods Section 1981, pp. 146–151 (1981)
11. Sande, I.: Hot Deck Imputation Procedures. In: Madow, W., Nisselson, H., Olkin, I. (eds.) Incomplete Data in Sample Surveys, Theory and Bibliographies, 3, pp. 339–349. Academic Press (1983)
12. Strike, K., Emam, K.E., Madhavji, N.: Software Cost Estimation with Incomplete Data. IEEE Transactions on Software Engineering 27, 890–908 (2001)
13. Andridge, R.R., Little, R.J.A.: A Review of Hot Deck Imputation for Survey Nonresponse. International Statistical Review 78(1), 40–64 (2010)

14. Barzi, F., Woodward, M.: Imputations of Missing Values in Practice: Results from Imputations of Serum Cholesterol in 28 Cohort Studies. American Journal of Epidemiology 160, 34–45 (2004)
15. Roth, P.L., Switzer III, F.S.: A Monte Carlo Analysis of Missing Data Techniques in a HRM Setting. Journal of Management 21, 1003–1023 (1995)
16. Yenduri, S., Iyengar, S.S.: Performance Evaluation of Imputation Methods for Incomplete Datasets. International Journal of Software Engineering and Knowledge Engineering 17, 127–152 (2007)
17. Kaiser, J.: The Effectiveness of Hot Deck Procedures in Small Samples. In: Proceedings of the Section on Survey Research Methods, American Statistical Association, pp. 523–528 (1983)
18. Kalton, G.: Compensating for Missing Survey Data. Institute for Social Research, University of Michigan, Ann Arbor (1983)
19. Brick, J.M., Kalton, G., Kim, J.K.: Variance Estimation with Hot Deck Imputation Using a Model. Survey Methodology 30, 57–66 (2004)
20. Brick, J.M., Kalton, G.: Handling Missing Data in Survey Research. Statistical Methods in Medical Research 5, 215–238 (1996)
21. Kalton, G., Kasprzyk, D.: The Treatment of Missing Survey Data. Survey Methodology 12, 1–16 (1986)
22. Roth, P.L.: Missing Data in Multiple Item Scales: A Monte Carlo Analysis of Missing Data Techniques. Organizational Research Methods 2, 211–232 (1999)
23. Nordholt, E.S.: Imputation: methods, simulation experiments and practical examples. International Statistical Review 66, 157–180 (1998)
24. Cohen, J.: A Power Primer. Quantitative Methods in Psychology 112, 155–159 (1992)
25. Borz, J., Döring, N.: Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler. Springer, Berlin (2009)
26. Fröhlich, M., Pieter, A.: Cohen's Effektstärken als Mass der Bewertung von praktischer Relevanz – Implikationen für die Praxis. Schweizerische Zeitschrift für Sportmedizin und Sporttraumatologie 57(4), 139–142 (2009)

BINER

BINary Search Based Efficient Regression

Saket Bharambe, Harshit Dubey, and Vikram Pudi

International Institute of Information Technology - Hyderabad
Hyderabad, India
{saket.bharambeug08,harshit.dubeyug08}@students.iiit.ac.in,
vikram@iiit.ac.in

Abstract. Regression is the study of functional dependency of one numeric variable with respect to another. In this paper, we present a novel, efficient, binary search based regression algorithm having the advantage of low computational complexity. These desirable features make BINER a very attractive alternative to existing approaches. The algorithm is interesting because instead of directly predicting the value of response variable, it recursively narrows down the range in which the response variable lies. Our empirical experiments with several real world datasets show that our algorithm, outperforms current state of art approaches and is faster by an order of magnitude.

Keywords: regression, logarithmic performance, binary search, efficient, accurate.

1 Introduction

The problem of regression is to estimate the value of a dependent variable based on the values of one or more independent variables, e.g., predicting price increase based on demand or money supply based on inflation rate etc. Regression analysis is used to understand which among the independent variables are related to the dependent variable and to explore the forms of these relationships. Regression algorithms can be used for prediction (including forecasting of time-series data), inference, hypothesis-testing and modeling of causal relationships.

Statistical approaches try to learn a probability function $P(y \mid x)$ and use it to predict the value of y for a given value of x. Users study the application domain to understand the form of this probability function. The function may have multiple parameters and coefficients in its expansion. Statistical approaches although popular, are not generic in that they require the user to make an intelligent guess, about the form of regression equation, so as to get the best fit for the data.

Regression analysis has been studied extensively in statistics [1], there have been only a few studies from the data mining perspective. The algorithms studied from a data mining perspective fall under the following broad categories

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 76–85, 2012.

- Decision Trees [2], Support Vector Machines [3], Neural Networks [4], Nearest Neighbor Algorithms [5], Ensemble Algorithms [6] among others. It may be noted that most of these studies were originally for classification, but have been later modified for regression [7].

1.1 Motivation and Contribution

The current existing standard algorithms [2,3,4,5] suffer from one or more of high computational complexity, poor results, fine tuning of parameters and extensive memory requirements. KNN [5] provides excellent accuracy, but has linear computational complexity. Finding an optimal decision tree [2] is a NP-Complete problem [8]. Neural networks [4] are highly dependent on the initialization of weight vectors and generally, have large training time. Also, the best fit structure of the neural network has to be intelligently guessed or determined by trial and error method.

The accuracy of KNN highly depends upon the distance metric used. Euclidean distance is a simple and efficient method for computing distance between two reference data points. More complex distance functions may provide better results depending on the dataset and domain. But user may refrain from using a better, generally computationally more complex, distance metric due to high run time of the algorithm. This motivated us to strive for an algorithm which has a significantly low run time and hence can incorporate expensive distance metrics with ease.

In this work, we contribute a new efficient technique for regression. Our algorithm is highly efficient and typically performs with logarithmic computational complexity on standard datasets. This is in contrast to the linear computational complexity of existing standard algorithms. It takes a single parameter K, the same as in KNN. The algorithm instead of directly predicting the response variable, narrows down the range in which the response variable has the maximum likelihood of occurrence and then interpolates to give the output. It more than often outperforms the conventional state of art methods on a wide variety of datasets as illustrated in the Experimental Section.

1.2 Organization of Paper

The organization of rest of the paper is as follows. Section 2 provides a mathematical model for the problem of regression. We throw light on related, and recent, work done in the field of regression in Section 3. In Section 4, intuition and methodology behind the algorithm is described. We explain the BINER algorithm in Section 5. In Section 6, experimental results are presented together with a thorough comparison with existing methods. Finally, in Section 7, conclusions are drawn.

2 Problem Formulation

In this section, we present the problem of regression and notation used to model the dataset.

The problem of regression is to estimate the value of a dependent variable (known as response variable) based on the values of one or more independent variables (known as feature variables). We model the tuple as $\{X, y\}$ where X is an ordered set of attribute values like $\{x_1, x_2, \ldots, x_d\}$ and y is the numeric variable to be predicted. Here x_i is the value of the i^{th} attribute and there are d attributes overall corresponding to a d-dimensional space.

Formally, the problem has the following inputs:

- An ordered set of feature variables Q i.e. $\{q_1, q_2, \ldots, q_d\}$
- A set of n tuples called the training dataset, D, $= \{(X_1, y_1), (X_2, y_2), \ldots, (X_n, y_n)\}$.

The output is an estimated value of y for the given query Q. Mathematically, it can be represented as

$$y = f(X, D, parameters), \tag{1}$$

where *parameters* are the arguments which the function $f()$ takes. These are generally set by user and are learned by trial and error method.

3 Related Work

Before presenting our algorithms, we would like to throw light on related work done in the recent past.

Traditional Statistical Approaches. Most existing approaches [1] follow a curve fitting methodology that requires the form of the curve to be specified in advance. This requires regression problems in each special application domain be studied and solved optimally for that domain. Another problem with these approaches is outlier (extreme cases) sensitivity.

Neural Networks and Support Vector Machines. Neural Networks [4] is a class of data mining approaches that has been used for regression and dimensionality reduction. However, Neural Networks are complex and an in-depth analysis of results obtained is not possible. Support Vector Machine [3] is a new data mining paradigm applied for regression. However, these techniques involve complex abstract mathematics thus resulting in techniques that are more difficult to implement, maintain, embed and modify as situation demands. Ensemble [6] based learning is a new approach to regression. A major problem associated with ensemble based learning is to determine the relative importance of each individual learner.

Indexing based approaches. Roussopoulus et. al. presented a branch and bound R-tree traversal algorithm [13] to find nearest neighbors of a query point. The algorithm required creating and sorting an Active Branch List of Nodes [13] at each of the node and then pruning the list. Another drawback of the approach is the depth first traversal of the index that incurs unnecessary disk IOs. Berchtold et. al. [12] suggest precalculating, approximating and indexing the solution space for the nearest neighbor problem in d dimensional spaces. Precalculating the solution space means determining the Voronoi diagram of the data points.

The exact Voronoi cells in d space are usually very complex, hence, the authors propose indexing approximation of the Voronoi cells. This approach is only appropriate for first nearest neighbor problem in high dimensional spaces.

Decision Tree. One of the first data mining approaches to regression were regression trees [2], a variation of decision trees where the predicted output values are stored at leaf node. These nodes are finite and hence the predicted output is limited to finite set of values in contrast with the problem of predicting a continuous variable as required in regression.

Nearest Neighbor. One of the oldest, accurate and simplest method for pattern classification and regression is K-Nearest-Neighbor (KNN)[5]. It has been studied at length over the past few decades and is widely applied in many fields. The KNN rule classifies each unlabeled example by the majority label of its k nearest neighbors in the training dataset. Despite its simplicity, the KNN rule often yields competitive results. A strong point of KNN is that, for all distributions, its probability of error is bounded above by twice the Bayes probability of error [11].

A recent work on prototype reduction, called Weighted Distance Nearest Neighbor (WDNN) [14] is based on retaining the informative instances and learning their weights for classification. The algorithm assigns a non negative weight to each training instance tuple at the training phase. Only the training instances with positive weight are retained (as the prototypes) in the test phase. Although the WDNN algorithm is well formulated and shows encouraging performance in practice, it can only work with K = 1. A more recent approach WDKNN [15] tries to reduce the time complexity of WDNN and extend it to work for $K > 1$.

Our work shares resemblance with segmented or piecewise regression [9]. However upon analysis, the techniques are entirely different. In segmented regression the independent variables are partitioned into segments. In our method, the response variable is partitioned into three groups to facilitate a binary search based methodology. Also, our work seems to share a resemblance with Binary Logistic Regression [10]. However the technique is again entirely different. In Binary logistic regression the response variable is assumed to follow a binomial logit model and the parameters of this model are learned from training data.

4 Intuition and Methodology of BINER

BINER follows a similar methodology to KNN [20]. In a nutshell, KNN follows this approach:

1. It finds the K nearest neighbors to the given query.
2. Weighted mean of response variables in K nearest neighbors is given as output. The weights are kept inversely proportional to distance from the query.

The intuition of BINER is that the query Q is expected to be similar to tuples whose response variable values are close to that of Q. Thus it is more beneficial to find nearest neighbors in a *locality* where tuples have nearby response variable

values rather than the whole dataset. This guarantees that even if the tuples in the considered locality are not the global nearest neighbors (nearest neighbors of the query in the complete dataset), the value of predicted response variable will be more appropriate.

Thus, the approach boils down to determining the locality (of nearby response variable value) in which to conduct the nearest neighbor search. Once the locality is determined, response variable can be estimated as a weighted mean of responses of K nearest neighbors in this locality where weight is inversely proportional to the distance from the query.

Like other KNN based approaches, BINER has the following core assumption - tuples with similar X-values have similar response variable values. This assumption is almost always borne out in practice and is justified also by our experiments.

5 The BINER Algorithm

The algorithm proceeds in two steps.

1. It first finds the range of tuples where the query Q has the maximum likelihood of occurrence. The term range (or locality), here, refers to consecutively indexed tuples in the dataset D and thus is characterized by two integers namely, start index and end index.
2. KNN is applied to these few (compared to D) tuples, and weighted mean of the K nearest neighbors in these ranges is quoted as output.

To find the range in which the query has the maximum likelihood of occurrence, the dataset is sorted in, say, non-decreasing manner of response variable values and then the function *biner* described below is invoked with Q as query, and range $(0, n)$ where n is the number of tuples in D.

The function, *biner*, iteratively bisects the current range until a range with size less than or equal to $2 * K$ is obtained (line 1) or a confident decision of bisecting a range cannot be taken (line 9-10), explained below. For each range, it makes three choices (lines 2-5) of half sized subranges namely, the lower half subrange (s_1, e_1), the center floating half (s_2, e_2) and the upper half (s_3, e_3), and computes distance of the query from these ranges (lines 6-9). The second subrange i.e. (s_2, e_2) is made to overlap with the other two ranges so as to ensure that tuples at end of first range and at start of third range get their due importance.

The distance of query Q from a range is calculated as

$$\sqrt{\sum \frac{(\mu_i - q_i)^2}{\sigma_i^2}} \tag{2}$$

where q_i is the ith attribute of the query, μ_i is the mean of ith attribute values in all tuples in the range and σ_i is the standard deviation of values of the ith attribute in the *whole* dataset, D. Standard deviation shows how much variation

there is from the mean, μ. A low standard deviation indicates that the data points tend to be very close to the mean whereas high standard deviation indicates that the data points are spread out over a large range of values. Thus standard deviation helps in understanding how good a representational point the mean is for the range. Hence, the term σ^2 occurs in the distance metric.

Algorithm 1. BINER Algorithm

```
Input: Query Q, Range (start, end)
Output: Range (s, e)
while end − start > 2 ∗ K do
    r = e − s
    s2, e2 = start + r/4, start + 3r/4
    s3, e3 = start + r/2, end
    d1 = getDistance(RangeMean(s1, e1), Q)
    d2 = getDistance(RangeMean(s2, e2), Q)
    d3 = getDistance(RangeMean(s3, e3), Q)
    if similar(d1, d2, d3) then
        return (start, end)
    else if min(d1, d2, d3) == di then
        start, end = si, ei
    end if
end while
return (start, end)
```

In order to find the subrange in which the query has the maximum likelihood of occurrence, the distance of subranges from the query are compared. Among the three subranges, the query has the maximum likelihood of occurrence in the subrange which has minimum distance from the query. Thus the subrange with minimum distance from the query is considered and the complete process is repeated again.

When the size of range becomes small, the distances of subranges from the query tend to have same or close by values. In such situations, subranges have similar tuples and thus their distances become *similar*. Hence, a confident decision to select a subrange cannot be made and the current range is returned (lines 9-10). It may be noted that only the two minimum distances are checked for similarity. If the two larger distances are similar, a confident decision of selecting the subrange with minimum distance can be made.

We say that two distances, d_i and d_j are *similar* if $\min(d_i/d_j, d_j/d_i)$ is greater than 0.95. The value of 0.95 was selected by experimentations and it works well on most of the datasets as shown in the experimental section. The limiting size of $2 * K$ was chosen in order to keep a margin for selection of K nearest neighbors.

The above process obtains a range (or locality) where the query point has maximum likelihood of occurrence. Then KNN is applied on the range (locality) returned by *biner* and weighted mean of response variables in K nearest neighbors is quoted as output. The weights are taken as $1/dist$, where $dist$ is the distance of the query and tuple.

5.1 Complexity Analysis

Before presenting the complexity analysis, we would like to first mention the preprocessing done here.

1. The first step is to sort the dataset D in, say, non-decreasing manner of the response variable.
2. Standard deviation, σ_i, of each attribute x_i is calculated and stored.
3. We store the mean, of all data points in all the feasible ranges (or localities) that may be encountered in our algorithm, in a Hash Table. Hence, calculation of mean becomes a constant order step. The key for the Hash Table, thus, will be two integers denoting the start and end index of tuples of in the range.

All feasible subranges are formed by recursively dividing a range into 3 ranges and stopping the recursion, when the size of range becomes less than or equal to $2 * K$. The value of K, is a parameter and is the same the one that would had been set for KNN. The range with start and end indices s and e respectively, is divided into 3 ranges namely $[\ s,\ s + (e - s)/2\ ]$, $[\ s + (e - s)/4,\ s + 3 * (e - s)/4\ ]$ and $[s + (e - s)/2,\ e\ ]$.

The algorithm at each stage divides the current range into 3 subranges each of half size of current range and considers one of them for subsequent processing. It can be observed that the function iterates $O(\log n)$ time. The function returns a range of size, say, R which is significantly smaller than n as confirmed by our experimentations. Thus computational complexity of the algorithm becomes $O(\log n + R)$ and when $R << n$ it becomes logarithmic. We illustrate our run time analysis on 7 datasets in Fig. 1

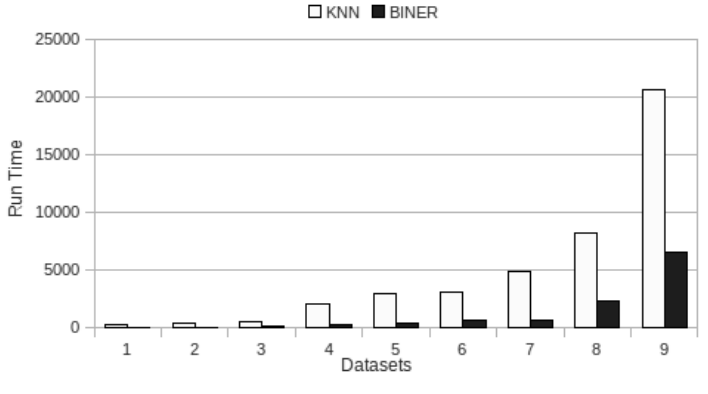


Fig. 1. Comparison of run times of KNN and BINER

6 Experimental Study

6.1 Performance Model

In this section, we demonstrate our experimental results. The experiments were done on a wide variety of datasets obtained from UCI data repository [17], Weka Datasets [18] and ML data repository [19]. We have evaluated our results against the standard existing state of art approaches. The algorithms used were available in Weka toolkit [16]. All the results have been obtained using 10-fold cross validation technique.

We have used two metrics for quantifying our results, namely, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). MAE is mean of the absolute errors (actual output - predicted output). RMSE is square root of mean of squared errors.

We compared our performance against the following approaches: K Nearest Neighbor, Isotonic, Linear Regression, Least Mean Square (LMS) algorithm, Radial Basis Function Network (RBF Network), Regression Tree (RepTree) and Decision Stump.

6.2 Results

Table 1. and Table 2. compare the result of our algorithm with other existing state of art approaches.

Table 1. Comparison of results of BINER and other standard approaches

Dataset	Autompg		Bodyfat		Flow		Housing		Space		Synfriedman2	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
BINER	**1.67**	**2.28**	**0.41**	**0.50**	**10.54**	**14.43**	**2.33**	**2.92**	**0.10**	**0.17**	52.55	64.35
KNN	**2.40**	**3.59**	0.49	0.62	11.89	16.42	4.14	5.52	**0.10**	**0.17**	**48.64**	**60.65**
Isotonic	3.16	4.30	0.52	0.67	11.55	14.18	3.80	5.32	0.12	0.16	219.09	284.46
Linear Reg	2.56	3.40	**0.43**	**0.56**	**10.99**	**13.26**	**3.39**	**4.91**	0.11	0.16	108.44	145.69
LMS	**2.50**	**2.59**	0.45	0.58	13.28	18.56	3.42	5.55	0.11	0.15	109.62	153.43
RBF Network	3.90	5.07	0.61	0.77	14.76	17.42	6.13	8.42	0.14	0.19	246.71	317.76
Rep Tree	**2.30**	**3.31**	0.52	0.67	12.11	15.57	3.18	4.84	**0.10**	**0.14**	**45.34**	**61.10**
Decision Stump	4.20	5.18	0.63	0.80	12.48	15.41	5.61	7.50	0.13	0.18	256.93	320.71

6.3 Discussion

We discuss our results in this section. It can be seen that our algorithm outperforms other algorithms in almost all the datasets. Also, BINER provides competitive results in typically logarithmic computational complexity which is very efficient.

Table 2. Comparison of results of BINER and other standard approaches

Dataset	Bank		Concrete		Forestfire		Slump		Synfriedman1		Synfriedman	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
BINER	**0.02**	**0.03**	**5.41**	**8.10**	**13.67**	**25.74**	6.01	9.79	**1.77**	**2.23**	**1.58**	**2.05**
KNN	**0.02**	**0.03**	**4.15**	**6.10**	14.69	24.37	**5.89**	**9.83**	**1.75**	**2.16**	**1.53**	**2.03**
Isotonic	0.03	0.46	10.81	13.45	19.18	63.50	**5.86**	**7.52**	3.59	4.32	3.69	4.44
Linear Reg	**0.02**	**0.03**	8.30	10.45	19.92	64.28	6.67	7.82	2.76	4.65	2.25	2.83
LMS	0.03	0.05	9.52	17.53	**12.88**	**64.91**	6.68	10.56	2.45	4.71	2.24	2.82
RBF Network	0.04	0.06	13.38	16.56	18.86	63.86	6.98	8.73	3.92	4.91	3.86	4.81
Rep Tree	**0.02**	**0.03**	**5.43**	**7.38**	19.24	64.56	6.14	8.19	2.57	3.24	2.69	3.45
Decision Stump	0.04	0.05	11.54	14.46	18.93	64.68	7.05	8.86	3.69	4.40	3.73	4.63

7 Conclusions

In this paper we have presented a new regression algorithm and evaluated it against existing standard algorithms. Our work is focused on finding a small locality in which the K nearest neighbors have the maximum probability of occurrence. In addition to this, it allows users to use a (computationally) complex distance metric without significant increase in run time. The algorithm finds this locality in nearly logarithmic computational time. We showed that the algorithm, more than often, outperforms existing standard state of art approaches on a wide variety of datasets and is faster by an order of magnitude.

We are planning for better methods of splitting ranges. Instead of simply, bisecting at some fixed points, if we divide at points where there is some abrupt change in y values, probably our algorithm will work better. We plan to incorporate this feature so as to enhance our algorithm.

References

1. Wang, Y., Witten, I.H.: Modeling for optimal probability prediction. In: ICML (2002)
2. Brieman, L., Friedman, J., Olshen, R., Stone, C.: Classfication and Regression Trees. Wadsworth Inc. (1984)
3. Chu, W., Keerthi, S.S.: New approaches to support vector ordinal regression. In: ICML (2005)
4. Haykin, S.: Neural Networks-A comprehensive foundation. Prentice-Hall (1999)
5. Cover, T.M., Hart, P.E.: Nearest Neighbor Pattern Classfication. IEEE Transactions on Information Theory (1967)
6. Schapire, R.E.: A Brief Introduction to Boosting. In: ICJAI (1999)
7. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Technique. Morgan Kaufmann (2005)
8. Hyafil, L., Rivest, R.: Constructing Optimal Binary Search Trees is NP Complete. Information Processing Letters (1976)
9. Hilbe, Joseph, M.: Logistic Regression Models. Chapman and Hall/CRC Press (2009)

10. Ritzema, H.P.: Frequency and Regression Analysis. In: Drainage Principles and Applications, ch.6, ILRI (1994)
11. Stone, C.: Consistent nonparametric regression. Annals of Statistics (1977)
12. Berchtold, S., Ertl, B., Keim, D.A., Kriegel, H.P., Seidl, T.: Fast Nearest Neighbor Search in High Dimensional Space. In: ICDE (1998)
13. Roussopoulus, N., Kelley, S., Vincent, F.: Nearest Neighbor Queries. In: SIGMOD (1995)
14. Jahromi, M.Z., Parvinnia, E., John, R.: A method of learning weighted similarity function to improve the performance of nearest neighbor. Information Sciences 179(17) (2009)
15. Yang, T., Cao, L., Zhang, C.: A Novel Prototype Reduction Method for the K-Nearest Neighbor Algorithm with $K \geq 1$. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010. LNCS, vol. 6119, pp. 89–100. Springer, Heidelberg (2010)
16. Hall, M., Ian, H.: The Weka Data Mining Software: An update. SIGKDD Explorations (2009)
17. UCI Data Repository, http://archive.ics.uci.edu/ml/datasets.html
18. Weka Dataset Repository, http://www.cs.waikato.ac.nz/ml/weka/index_datasets.html
19. ML Dataset Repository, http://mldata.org/repository/data/
20. Todeschini, R.: Weighted k-Nearest Neighbor Method for the Calculation of Missing Values. Chemometrics and Intelligent Laboratory Systems (1990)

A New Approach for Association Rule Mining and Bi-clustering Using Formal Concept Analysis

Kartick Chandra Mondal[1], Nicolas Pasquier[1], Anirban Mukhopadhyay[2], Ujjwal Maulik[3], and Sanghamitra Bandhopadyay[4]

[1] Laboratoire I3S (CNRS UMR-7271), Université de Nice Sophia-Antipolis, France
[2] Department of Computer Science and Engineering, University of Kalyani, India
[3] Department of Comupter Science and Engineering, University of Jadavpur, India
[4] Machine Intelligent Unit, Indian Statistical Institute, Kolkata, India
{mondal,pasquier}@i3s.unice.fr, anirban@klyuniv.ac.in, umaulik@cse.jdvu.ac.in, sanghami@isical.in

Abstract. Association rule mining and bi-clustering are data mining tasks that have become very popular in many application domains, particularly in bioinformatics. However, to our knowledge, no algorithm was introduced for performing these two tasks in one process. We propose a new approach called FIST for extracting bases of extended association rules and conceptual bi-clusters conjointly. This approach is based on the frequent closed itemsets framework and requires a unique scan of the database. It uses a new suffix tree based data structure to reduce memory usage and improve the extraction efficiency, allowing parallel processing of the tree branches. Experiments conducted to assess its applicability to very large datasets show that FIST memory requirements and execution times are in most cases equivalent to frequent closed itemsets based algorithms and lower than frequent itemsets based algorithms.

Keywords: Association Rules, Bi-clustering, Closure Lattice, Frequent Closed Itemsets, Suffix Tree Data Structures.

1 Introduction

Data mining, also known as knowledge discovery from database (KDD), is the task of finding unknown and potentially important information from large databases. The most prominent data mining tasks, gaining actually much importance in many application domains, are association rule mining, classification, clustering and regression [7]. Bi-clustering, that is a special case of clustering, is also gaining much popularity, specially in bioinformatics [9].

Association rule mining (ARM) aims at finding significant relationships between data values, called items, in a database. ARM is a very popular and important, but expensive, task in data mining. Since the ARM problem definition, several approaches have been proposed in the literature to improve ARM efficiency. See [2] for a complete survey on ARM principles and algorithms.

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 86–101, 2012.

Let the database D be a set of data rows, called transactions, $D = \{t_1, t_2, \cdots, t_n\}$ where each transaction t_i is a set of items from the item list $L = \{i_1, i_2, \cdots, i_m\}$, i.e. $t_i \subseteq L$. Literally, an association rule for a *frequent itemset* I, i.e. a set of items that is contained in at least *minsup* number of transactions where *minsup* is an user defined minimum support threshold, is denoted as $I_1 \Rightarrow I_2$, where $I_1, I_2 \subset L$, $I_1 \cup I_2 = I$ and $I_1 \cap I_2 = \emptyset$. The problem of ARM is generally divided into two sub-problems:

1. Find all frequent itemsets with their support from the database D.
2. Generate all association rules with confidence greater than or equal to the minimum confidence threshold.

The second sub-problem is straightforward and the problem of ARM is usually reduced to the problem of finding frequent itemsets. Almost all algorithms based on the frequent itemsets approach use the subset lattice (or itemset lattice) framework and the two following properties:

i. All subsets of a frequent itemset are frequent.
ii. All supersets of an infrequent itemset are infrequent.

For big databases, the list of all frequent itemsets is very large and even larger in the case of dense data where transactions are long. Later, this FIs mining approach was extended to the mining of maximal frequent itemsets (MFIs) from which all frequent itemsets can be derived, but not their support.

In 1998, a condensed representation of frequent itemsets called *frequent closed itemsets* [11,18] was introduced. The frequent closed itemsets, defined using the *Galois connection closure* used in Formal Concept Analysis [5], form the closed itemset lattice [11] that is a sub-order of the subset lattice. Informally, an itemset is closed if none of its supersets is contained in the same number of transactions as it. Frequent closed itemsets (FCIs) constitute a lossless condensed representation of all FIs: All FIs and their support can be obtained in a straightforward manner from the FCIs and their support. Since the number of FCIs is in most cases much lower than the number of FIs, their computation improves ARM execution times and memory usage. ARM extraction using the FCIs framework is based on the three following properties:

i. All subsets of a frequent closed itemset are frequent itemsets.
ii. All supersets of an infrequent closed itemset are infrequent itemsets.
iii. The support of a frequent itemset is equal to the support of the smallest closed itemset containing it.

Many algorithms have been proposed in the literature in recent years for finding FCIs [16,17]. Almost all of them use either the prefix tree [1] or the FP-Tree [8] as an internal data structure for compressed representation of the dataset in main memory. Their efficiency depends mainly on the properties of the database (number of items, density, size of transactions, etc.). In several cases, such as biological data, FIs and MFIs mining pose efficiency problems since the number of FIs is very large and the set of MFIs does not contain all information required

to directly generate the association rules. In such cases, the FCIs framework is a good alternative for the ARM problem as the set of all FCIs is sufficient for finding the association rules and is much smaller than the set of all FIs.

Bi-clustering aims at finding sub-matrices that associate a set of rows and a set of columns such that all these rows have the same value for each of these columns in the matrix. The bi-clusters extracted with FIST are maximal sets of related rows and columns defined as above. These bi-clusters, called *conceptual clusters*, constitute a hierarchical lattice structure defined according to inclusion relation. They are overlapping: each row and each column can be member of several bi-clusters. Contrarily to recent works on gene expression time series [10], where columns represent the evolution of gene expressions during time for one biological experiment, FIST do not restrict bi-clusters to contiguous columns. Extracting such bi-clusters is known to be an NP-Complete problem [14].

Here, we propose a new algorithm for mining conjointly conceptual clusters, or bi-clusters, and bases, or minimal covers, of extended association rules. This algorithm, called *FIST* (Frequent Itemsets Suffix Tree), is based on the FCIs framework and uses a new suffix tree based data structure for computations in main memory. This data structure does not require complex operations such as maintaining transverse chained lists of items and can be implemented using standard data structures like Java collections. It was designed to balance memory usage and computation efficiency and can easily be adapted to specific requirements of particular application cases. It also allows parallel processing of the suffix tree branches in multi-threated environments. FIST finds the *frequent closed patterns*, each associating a FCI and its corresponding object identifier list, i.e., the identifiers of database objects containing the FCI. Its size, that corresponds to the number of occurrences in the database, gives the support of the FCI. The CHARM algorithm [19] also uses object id space to find frequent closed itemsets. However, it discards object ids when they are no more needed whereas FIST keeps them in main memory for generating bi-clusters. Moreover, the list of supporting objects of each association rule is also generated by FIST, instead of only its support value as in classical ARM approaches. The user can then examine the list of objects concerned by each rule. This can be particularly useful in a certain number of applications such as genomics or proteomics where identifying specific genes or proteins concerned by a rule is important, particularly if rules contain semantic information such as biological annotations.

FIST proceeds in three steps: In the first step, the Frequent Generalized Itemset Suffix Tree (fGIST) is created from the database and is stored in main memory for the second step. In the second step, frequent closed patterns are extracted from the fGIST by performing inclusion and intersection operations. Then, during the third step, bases of association rules and conceptual clusters are generated in a straightforward manner.

The rest of the paper is organized as follows. A brief terminology is given in section 2. In section 3, we present the FIST algorithm. Section 4 shows experimental results and concluding remarks are given in section 5.

2 Terminology

In this section, we define the most relevant data mining terms used to help the understanding of the problems of finding association rules and conceptual clusters in a data mining context.

Definition 1 (Database). *A database D is a triplet (O, L, R) where O is a finite set of objects (rows), L is a finite set of items (values of attributes or variables) represented as columns and R is a binary relation showing relationships between rows and columns: $R \subseteq O \times L$. Every couple $(o, i) \in R$, where $o \in O$ and $i \in L$, means that the item i belongs to the object o: $i \in o$.*

Definition 2 (Itemsets). *A non-empty finite set of items $I \subseteq L$ in D is called an itemset. An itemset containing k items is called a k-itemset.*

Definition 3 (Support). *The support of an itemset I, denoted $supp(I)$, is the frequency of occurrence of I in D:*

$$supp(I) = \frac{|\{o \in O \mid I \subseteq o\}|}{|\{o \in O\}|} \tag{1}$$

Definition 4 (Frequent itemsets). *An itemset I with support at least equals to the user-defined threshold $minsupp$ is called a frequent itemset: $I \subseteq L$ is frequent iff $supp(I) \geq minsupp$.*

Definition 5 (Maximal frequent itemsets). *Let F be the set of all frequent itemsets. A frequent itemset in the set F is called a maximal frequent itemset if none of its proper supersets is frequent, i.e. present in the set F.*

Definition 6 (Galois closure operator [5]). *The Galois closure operator γ holds the following properties for I, I_1, $I_2 \subseteq L$ in the power set of L of size 2^L:*

- *Extension:* $I \subseteq \gamma(I)$
- *Idempotency:* $\gamma(\gamma(I)) = \gamma(I)$
- *Monotonicity:* $I_1 \subseteq I_2 \Rightarrow \gamma(I_1) \subseteq \gamma(I_2)$

Definition 7 (Closed itemsets). *An itemset C is said to be closed in the database D if the application of the Galois closure operator γ to C gives C. If C is a closed itemset, none of its proper supersets present in D has support equal to the support of C.*

Definition 8 (Frequent closed itemsets). *A closed itemset which support is greater than or equal to the user defined minimum threshold support is called a frequent closed itemset.*

Definition 9 (Generators [6]). *The generators of a frequent closed itemset C are the minimal itemsets, according to inclusion, which closure is C.*

Definition 10 (Association rules). *The implication relationship between two itemsets I_1 and I_2 with the form $r : I_1 \longrightarrow I_2$ where $I_1, I_2 \subset L$ and $I_1 \cap I_2 = \emptyset$ is called an association rule. I_1 and I_2 are respectively called the antecedent and the consequent of the rule.*

Definition 11 (Confidence). *The confidence of an association rule $r : I_1 \longrightarrow I_2$ is the ratio of the support of the itemset $I_1 \cup I_2$ to the support of the antecedent in the rule:*

$$conf(r) = \frac{supp(I_1 \cup I_2)}{supp(I_1)} \tag{2}$$

Definition 12 (Valid association rules). *An association rule $r : I_1 \longrightarrow I_2$ where $I_1 \cup I_2$ is a frequent itemset and which confidence $conf(r)$ is greater than or equal to the user defined minimum confidence threshold is called a valid association rule.*

Definition 13 (Exact and approximate association rules). *Association rules with confidence equals to 1 are called exact association rules and association rules with confidence less than 1 are called approximate association rules.*

Definition 14 (Clusters). *A cluster is a subset of rows that are similar according to a distance metric defined on variable values. For the above database D, clusters are defined as $C_k = \{o \subset O \mid \forall\ o_i,\ o_j \in o,\ d(o_i,\ o_j) < \sigma\}$ where σ is a user-defined threshold.*

Definition 15 (Bi-clusters). *A bi-cluster is a sub-matrix associating a subset of rows and a subset of columns such that all these rows have a similar value for each of these columns. For the above database D, the bi-clusters have the form $B_k = (o,\ i)$ where $o \subset O$ and $i \subset L$.*

3 FIST Algorithm

In this section, we present the FIST algorithm for extracting bases of association rules and conceptual clusters without extra processing time or database scan. The algorithm is divided into three main phases: Database preprocessing, extracting frequent closed patterns and generating knowledge patterns.

3.1 Database Preprocessing

This preprocessing aims at minimizing execution times and memory usage of subsequent phases. It is divided into two steps: Generating the *Item Table* (IT) and generating the *Sorted Frequent Database* (SFD). This preprocessing phase is required for each different *minsupp* value used for experiments; it is not required if only the *minconf* value is modified.

During the first step, the support (number of occurrences) of each item in the database is counted and all infrequent items, according to the minimum support threshold value supplied by the user, are deleted. Then, the remaining frequent

items are sorted in ascending order of their support. In the second step, data values are mapped to discrete numbers. Each item is then a unique discrete number representing a data, that is a pair *attribute = value*, in the database. This representation optimizes the memory space required for storing itemsets and improves the efficiency of comparison operations. Then, for each row of the database containing frequent items, one line is created in the SFD. If a row contains only infrequent items, it is thus not represented in the SFD.

An exemple database D, in binary format, is given in figure 1 with the corresponding *Item Table* and *SFD* for *minsupp*=40%. O_i rows and P_j columns represent proteins and A_k columns represent biological annotations. A "1" in a cell (O_i, P_j) means that proteins O_i and P_j interact and in a cell (O_i, A_k) means that protein O_i is annotated with A_k. Values "?" mean that there is no relation between the O_i protein and the corresponding protein or annotation in column. We can see that data values P_4 and A_1 that are infrequent do not generate items in the *Item Table* and that row O_4 that contains only infrequent data values is not represented in the *SFD*.

(A) Data Matrix

OID	P_1	P_2	P_3	P_4	P_5	A_1	A_2	A_3	A_4
O_1	1	?	1	?	?	?	1	?	1
O_2	?	1	?	?	1	?	?	1	?
O_3	1	?	1	?	?	1	1	?	?
O_4	?	?	?	1	?	1	?	?	?
O_5	?	1	?	1	1	1	?	1	?
O_6	1	?	1	?	?	?	1	?	1
O_7	?	1	?	?	1	?	?	1	?
O_8	1	?	1	1	?	?	1	?	1
O_9	?	1	?	?	1	?	?	1	?
O_{10}	1	1	1	?	1	?	1	?	1

(B) Item Table

Data	Support	Item
P_4	3	-
A_1	3	-
A_4	4	1
A_3	4	2
P_1	5	3
P_3	5	4
A_2	5	5
P_2	5	6
P_5	5	7

(C) SFD

1 3 4 5
2 6 7
3 4 5
2 6 7
1 3 4 5
2 6 7
1 3 4 5
2 6 7
1 3 4 5 6 7

Fig. 1. Example Database D

3.2 Mining Frequent Closed Patterns

The mining of frequent closed patterns is the core of the algorithm. This phase is divided into two steps: generating the *frequent Generalised Itemset Suffix Tree* (fGIST) and finding frequent closed itemsets with their object identifier (OID) list, each constituting a *frequent closed pattern* (FCP), from the fGIST using inclusion and intersection operations on the itemset and OIDs spaces.

The fGIST data structure is a compressed representation of the database that is stored in main memory for further processing. Each node of the fGIST represents an item and a branch from the root to a leaf represents a frequent itemset of the SFD. Each leaf of the fGIST contains the list of object identifiers containing the itemset in the SFD. The combination of the suffix decomposition of frequent itemsets of the SFD and of the item ordering in increasing support values optimizes the size of the fGIST as the smallest suffixes are the most

frequent itemsets in SFD and are thus most likely closed itemsets. The *fGIST* for the *SFD* in figure 1 is given in figure 2 and the corresponding FCPs are shown in figure 3 with their hierarchical relations defined according to inclusion: A sup-cluster contains a subset of the rows of its sub-clusters and a superset of the columns of its sub-clusters. These bi-clusters show that proteins O_2, O_5, O_7, O_9, O_{10} all interact with proteins P_2 and P_5 but only O_{10} is not annotated with A_3 suggesting new possible tracks of studies for example.

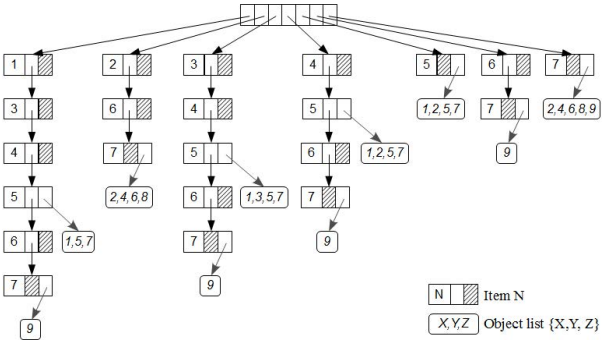


Fig. 2. frequent Generalised Itemset Suffix Tree

Building frequent Generalised Itemset Suffix Tree: The pseudo-code for building the fGIST from the SFD is given in algorithm 1. During this process, rows of the SFD are accessed one by one (lines 3 to 20). Each row read is represented as a vector V_i of numbers (line 4) and a *suffix terminator* T_i containing the OID of the row in the SFD is created (line 5). Then, the list R of suffixes of the string V_i is generated (lines 6 to 14). These suffixes are the subsets of the itemset V_i obtained by deleting successively one item to V_i from the first to the penultimate. For instance, suffixes of itemset {1, 2, 3} are itemsets {1, 2, 3}, {2, 3} and {3}. The k^{th} item of V_i is denoted V_i^k in the algorithms. Suffixes are then inserted in the fGIST (lines 16 to 18). During this process, the SFD is accessed only once; this minimizes disk accesses that are time expensive operations.

Function InsertSuffix($node(I)$, S_j): The pseudo-code of this function is given in algorithm 2. It takes a node $node(I)$ and a suffix S_j as arguments. It recursively creates (line 4), or updates, sub-nodes starting from the *ROOT* node to represent S_j as a branch in the fGIST. If the S_j suffix was already inserted in the fGIST, then only the leaf node of the branch representing S_j is updated by adding its suffix terminator (last element of S_j) to the list of OIDs of the leaf (line 11).

Extracting Frequent Closed Patterns: The second step of this phase consists in retrieving the FCP, i.e. FCIs with the list of identifiers of objects containing each in the SFD, from the fGIST. Algorithm 3 gives the pseudo-code of this step. The output of this step is the *FCP* set containing the list of frequent closed itemsets with their associated OID list. During this step, each itemset

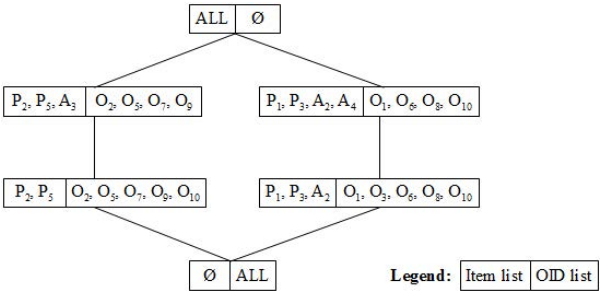


Fig. 3. Frequent Closed Patterns

corresponding to a branch of the fGIST from root to leaf is traversed and the algorithm tests if this itemset is closed or not as follows (lines 2 to 9). At first, each branch of the fGIST is traversed from root to leaf and a new entry is created in the *FCP* set for the collected items L_i and the corresponding OID list $L_i.OIDs$ in the leaf node (line 3). The non-closed itemsets are then identified and deleted from the *FCP* set: If an itemset is included in another itemset and both have identical OID lists, then the included itemset is not closed and is deleted from *FCP* (line 5). The second operation consists to identify the remaining few frequent closed itemsets not already identified (lines 10 to 23). These FCIs are those that can be obtained only by intersecting two FCIs and that do not correspond to suffixes of itemsets in the SFD. For this, intersection operations are performed between two FCIs and if the resulting set is frequent and not present in the *FCP* set, a new FCP is generated (lines 16 to 18). The OID list of this new FCP is the union of the OID lists of the two intersected FCIs (line 14). This procedure for identifying new FCPs continues till no new FCP is found this way (line 11 to 23). However, all pairs of FCIs don't have to be tested and only newly created FCIs L_k are intersected with other FCIs in the *FCP* set. For this, new FCPs are stored in the *NFCP* reference set (lines 12 and 19). For the first iteration, the *NFCP* reference set is initialized with *FCP* members (line 10). At the end, the final *FCP* set contains all frequent closed itemsets with associated OID list.

3.3 Generating Conceptual Clusters and Bases of Association Rules

During the third phase of the algorithm, conceptual clusters, generators and bases of association rules are generated using the *FCP* set. Compared with traditional ARM approaches, FIST association rules provide more information to the end-user as the list of objects supporting each rule is genererated instead of only the support of the rule.

Extracting Bi-clusters and Generators: Algorithm 4 gives the pseudo-code of the conceptual cluster creation and generator identification. First, frequent closed itemsets in the *FCP* set are sorted in ascending order of their size (line 2). Then, for

Algorithm 1. Building fGIST

```
1: begin
2: l ← |SFD|
3: for i = 1 to l do
4:    map the i^th row to a vector V_i
5:    create suffix terminator T_i for V_i
6:    k ← length(V_i)
7:    for m = 1 to k do
8:       S_m ← ∅
9:       for n = m to k do
10:         S_m ← S_m ∪ V_i^n
11:      end for
12:      S_m ← S_m ∪ T_i
13:   end for
14:   R = ⋃ {S_m}
15:   destroy(V_i)
16:   for all S_m ∈ R do
17:      InsertSuffix(ROOT, S_m)
18:   end for
19:   destroy(R)
20: end for
21: end
```

Algorithm 2. InsertSuffix($node(I), S_j$)

```
1: begin
2: if length(S_j) ≠ 1 then
3:    if node(I).children = ∅ or
      S_j^1 ∉ node(I).children then
4:       create node(I).child ← S_j^1
5:    else
6:       identify node(I).child = S_j^1
7:    end if
8:    delete S_j^1 from S_j
9:    InsertSuffix(node(I).child, S_j)
10: else
11:   add S_j^1 to node(I).OIDs
12: end if
13: end
```

each frequent closed pattern in the *FCP* set (lines 4 to 27), the generators of the FCI are identified in a levelwise manner (lines 5 to 25). First, the subsets of the frequent closed itemset are created in increasing order of their size (line 7). Then, we test for each subset if it is present among generators already found (lines 9 to 12) or among the FCIs (lines 13 to 17). If these tests were false, a new entry for generators of the FCI is created in the *GEN* set containing generators and their closure (lines 18 to 21). If no generator was identified, this process is repeated for subsets of a size increased by one (line 23) until the first iteration that founds at least one generator. If all subsets were proceeded and no generator was found, then the frequent closed itemset is itself its only generator (line 25). Finally, a bi-cluster is created in the *BIC* set representing maximal sets of related rows and columns respectively (line 26). As a final operation (line 28), items in generators and bi-clusters are mapped to their original names in the source database to simplify interpretation by the end-user. To limit the number of extracted patterns, objective or subjective measures for selecting patterns according to the application objectives can easily be integrated in the process.

Generating Bases of Association Rules: Algorithm 5 shows the pseudo-code for finding bases of association rules using the *FCP* and *GEN* sets. These bases are extracted into three sets, one for exact association rules and two different bases for approximate association rules. The base of min-max exact association rules contains valid association rules between a generator (minimal set) and the frequent closed itemsets that is its closure (maximal set) [13]. The base of min-max approximate association rules contains valid association rules between

Algorithm 3. Extracting Frequent Closed Patterns

```
 1: begin
 2: for each itemset L_i in fGIST
 3:    insert pattern {L_i, L_i.OIDs} into FCP
 4:    for each itemset L_j successor of L_i in fGIST with length(L_j) > length(L_i)
 5:       if L_i ⊂ L_j and L_i.OIDs = L_j.OIDs then
 6:          delete pattern {L_i, L_i.OIDs} from FCP
 7:       end if
 8:    end for
 9: end for
10: NFCP ← FCP
11: while NFCP ≠ ∅ do
12:    NFCP ← ∅
13:    for each itemset L_k in NFCP
14:       for each itemset L_i in FCP
15:          L_m ← L_i ∩ L_k
16:          if L_m ≠ ∅ and L_m ∉ FCP then
17:             L_m.OIDs ← L_i.OIDs ∪ L_k.OIDs
18:             insert pattern {L_m, L_m.OIDs} into FCP
19:             NFCP ← NFCP ∪ {L_m, L_m.OIDs}
20:          end if
21:       end for
22:    end for
23: end while
24: end
```

a generator and frequent closed itemsets that are supersets of its closure [13]. The proper base of approximate association rules contains rules between two frequent closed itemsets related by inclusion [12]. First, the bases of min-max association rules are created by considering each generator successively (lines 3 to 17). For each generator, a min-max exact association rule is created in the *AR_E* set if the generator is different from its closure (lines 5 to 9). This rule as a support equals to the size of the object id list *F.OIDs* of the closure *F.Itemset* and a confidence equal to 1, and its associated supporting OID list is *F.OIDs* (line 7). Then a min-max approximate association rule is created in the *AR_SB* set for each FCI in the *FCP* set that is a superset of the generator closure (lines 10 to 15). This rule as a support equals to the size of the closure OID list *F.OIDs* and a confidence equal to the division of the size of the generator OID list *G.OIDs* by the size of *F.OIDs*, and its supporting OID list is *F.OIDs* (line 12). Then, the proper base of approximate association rules, stored in the *AR_PB* set, is created in a straightforward manner (lines 18 to 23). This base contains rules between a FCI and FCIs that are its supersets (line 20). The last step of the algorithm consists to map the items in the association rules to their original names in the source database (line 24).

Algorithm 4. Finding Bi-clusters and Generators

```
1: begin
2: sort FCP in increasing size of itemsets
3: GEN, BIC ← ∅
4: for all F in FCP do
5:    found_gen ← false and gen_size ← 1
6:    while found_gen = false and gen_size < |F.Itemset| do
7:       SUB ← subsets of F.Itemset of size gen_size
8:       for all subset S in SUB do
9:          not_gen ← false
10:         for all G.Generator in GEN do
11:            if S = G.Generator then not_gen ← true
12:         end for
13:         if not_gen = false then
14:            for all C ∈ FCP preceding F ∈ FCP do
15:               if S ⊆ C.Itemset then not_gen ← true
16:            end for
17:         end if
18:         if not_gen = false then
19:            insert{S, F} into GEN
20:            found_gen ← true
21:         end if
22:      end for
23:      gen_size = gen_size + 1
24:   end while
25:   if found_gen = false then insert{F.Itemset, F.Itemset} into GEN
26:   insert{F.Itemset, F.OIDs} into BIC
27: end for
28: map items in BIC, GEN to database values
29: return(BIC, GEN)
30: end
```

4 Performance Analysis

For portability, the FIST algorithm was implemented in Java. A PC with an Intel Core 2 Duo (T5670 Series) processor at 1.80 GHz and 4 GB DDR2 of RAM running under the 32 bits Windows 7 Professional Edition operating system was used for experiments. For comparison of performances, three state-of-art algorithms were used: Apriori, Zart and DCI-Closed. Apriori and ZART are two frequent itemsets based algorithms for mining association rules and DCI-Closed is actually, to the best of our knowledge, the most efficient frequent closed itemsets based algorithm for mining association rules. It should however be noted that FIST generates more information than these three algorithms. We could not find a Java implementation of bi-clustering or formal concept mining able to process the datasets used for these experiments. The optimized Java implementations of the Apriori, Zart and DCI-Closed algorithms used for these experiments are available at `http://www.philippe-fournier-viger.com/spmf/`. We

Algorithm 5. Generating Bases of Association Rules

```
1: begin
2: AR_E, AR_SB, AR_PB ← ∅
3: for all G.Generator in GEN do
4:    for all F.Itemset in FCP do
5:       if G.Closure = F.Itemset then
6:          if G.Generator ≠ F.Itemset then
7:             create rule r: {G.Generator ⇒ F.Itemset\G.Generator, sup(r) =
               |F.OIDs|, conf(r) = 1, F.OIDs}
8:             insert r into AR_E
9:          end if
10:      else
11:         if G.Closure ⊂ F.Itemset then
12:            create rule r: {G.Generator ⇒ F.Itemset\G.Generator, sup(r) =
               |F.OIDs|, conf(r) = |F.OIDs|/|G.OIDs|, F.OIDs}
13:            insert r into AR_SB
14:         end if
15:      end if
16:   end for
17: end for
18: for all F_i.Itemset in FCP do
19:    for all F_j.Itemset in FCP where F_j.Itemset ⊃ F_i.Itemset do
20:       create rule r: {F_i.Itemset ⇒ F_j.Itemset\F_i.Itemset, sup(r) = |F_j.OIDs|,
          conf(r) = |F_j.OIDs|/|F_i.OIDs|, F_j.OIDs}
21:       insert r into AR_PB
22:    end for
23: end for
24: map items in AR_E, AR_SB, AR_PB to database values
25: return(AR_E, AR_SB, AR_PB)
26: end
```

present experimental results in terms of execution times and memory usage for four bioinformatics datasets. These datasets and their descriptions are available at http://keia.i3s.unice.fr/?Datasets. The reason for using these datasets is that knowledge patterns generated by FIST are particularly relevant in the field of bioinformatics and biological analysis.

The first dataset was constructed from the HIV-1–Human Protein Protein Interaction Database of the NIAID [4,15] available at http://www.ncbi.nlm.nih.gov/RefSeq/HIVInteractions/. It contains 1433 rows corresponding to human proteins and 19 columns corresponding to the HIV-1 proteins. Each cell of this matrix contains a 1 if a positive interaction between the corresponding pair of proteins was reported and a question mark if no interaction was reported. The second dataset was constructed by integrating biological knowledge and bibliographical annotations of human proteins from the UniProtKB-GOA database (http://www.geneontology.org/GO.downloads.annotations.shtml) with interaction data. 1149 distinct Gene Ontology (GO) annotations, describing function and characteristics of human proteins, and 2670 distinct bibliographic

annotations from Pubmed and Reactome related publications were integrated in this dataset. This dataset contains up to 40 GO annotations and 88 publication annotations for each gene. The third dataset is the well known Eisen Yeast Dataset [3] with gene expression values in terms of over expressed, under expressed, and not expressed. This dataset contains 2464 rows corresponding to yeast gene and 79 columns corresponding to different experimental biological conditions. The fourth dataset results from the integration with the Eisen Yeast gene expressions of the following gene annotations as columns: 76 GO, 97 different pathways, 109 transcriptional regulators, 1776 phenotypes and 7623 Pubmed IDs. It contains up to 25 GO, 14 pathways, 25 transcriptional regulators, 14 phenotypes and 581 Pubmed ID annotations for each gene.

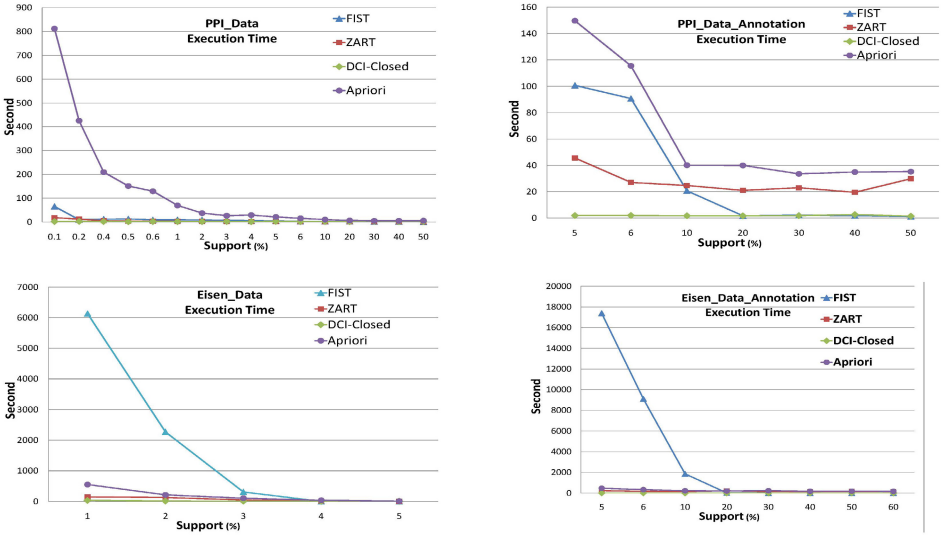


Fig. 4. Comparison of execution times

Execution times of the algorithms for the four datasets are presented in Figure 4. We can see that for the Eisen Yeast Datasets, execution times of FIST are higher for very low *minsupport* threshold values. This is because of the exponential number of bi-clusters generated when *minsupport* is lowered. However, execution times remain acceptable in all cases, ranging from seconds to minutes. Except for these challenging cases, execution times of FIST are equivalent to those of the three other algorithms, even if it generates much more patterns. For the two PPI datasets, we can see that Apriori is the the worst performer and that execution times of ZART are lower than those of FIST only for the integrated dataset and for the two smallest *minsupport* values. Except for these two cases, execution times of FIST are equivalent to those of DCI-Closed for the two PPI datasets.

Memory usage of the four algorithms are shown in Figure 5. We can see that, even if FIST generates more information as OID lists are generated instead of

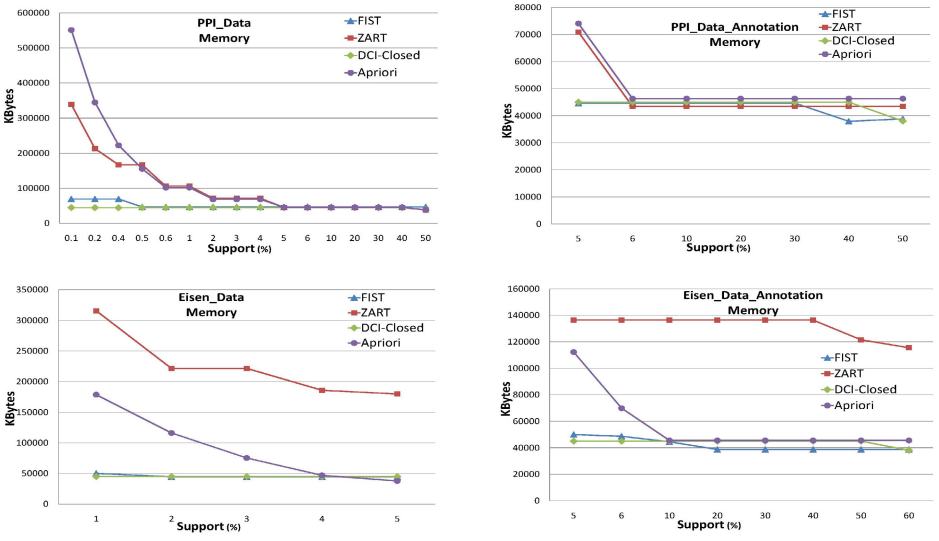


Fig. 5. Comparison of memory consumption

only support values for the other three algorithms, the reduction of memory requirement remains important compared to the frequent itemsets based approach used by Apriori and ZART. FIST memory usage is identical, and sometimes even lower, than those of DCI-Closed as both are based on the frequent closed itemsets framework. This is due to the suffix tree based data structure that, together with the re-ordering of items according to their support, optimizes the number of itemsets stored in main memory for computations of FCIs and OID lists.

5 Conclusions

We present the FIST approach for mining bases of extended association rules and bi-clusters conjointly, without extra database access. It uses a suffix tree based data structure and items re-ordering according to supports to optimize the extraction of frequent closed itemsets with lists of object identifiers containing each. These frequent closed patterns are used to generate bases, or minimal covers, of association rules and conceptual clusters, or bi-clusters. This suffix tree based data structure does not require complex procedures, like maintaining a transverse chained list of items, and permits parallel processing of the tree branches in multi-threated environments. Another important feature of FIST is that the lists of objects supporting each association rule are generated, instead of only their support value as in classical association rule mining approaches.

Experiments were conducted on four bioinformatics datasets, two with Yeast gene expressions and HIV-1–Human protein interactions and two integrating with these data both biological and publication annotations. These experiments

show that FIST efficiently extracts bases association rules and bi-clusters even for a very large number of items such as for datasets integrating annotations that contain several thousands of variables (data matrix columns). They also show that FIST execution times are always lower than those of the state-of-the-art Apriori association rule mining algorithm, even for the optimized implementation of Apriori used for these experiments. Regarding memory usage, these experiments show that FIST requirements are always lower than frequent itemsets based approaches and are equivalent to those of the most efficient frequent closed itemsets based approaches, even if FIST generates not only association rules but also bi-clusters. In the future, we plan to apply FIST to different domains of application and integrate subjective and objective measures of interestingness to filter mined patterns according to the end-user's interests.

References

1. Agrawal, R., Srikant, R.: Fast algorithm for mining association rules in large databases. In: Proc. VLDB, pp. 487–499 (1994)
2. Ceglar, A., Roddick, J.: Association mining. ACM Computing Surveys 38 (2006)
3. Eisen, M., Spellman, P., Brown, P.O., Botstein, D.: Cluster analysis and display of genome wide expression patterns. Proc. Natl. Acad. Sci. USA 95(25), 14863–14868 (1998)
4. Fu, W., Sanders-Beer, B., Katz, K., Maglott, D., Pruitt, K., Ptak, R.: Human immunodeficiency virus type 1, human protein interaction database at NCBI. Nucleic Acids Research 37, 417–422 (2009)
5. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer (1999)
6. Hamrouni, T., Ben Yahia, S., Mephu Nguifo, E.: Succinct System of Minimal Generators: A Thorough Study, Limitations and New Definitions. In: Yahia, S.B., Nguifo, E.M., Belohlavek, R. (eds.) CLA 2006. LNCS (LNAI), vol. 4923, pp. 80–95. Springer, Heidelberg (2008)
7. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques, 3rd edn. Morgan Kaufmann Series in Data Management Systems (2011)
8. Han, J., Pei, J.: Mining frequent patterns by pattern-growth: Methodology and implications. SIGKDD Explor. Newsl. 2(2), 14–20 (2000)
9. Madeira, S., Oliveira, A.: Biclustering algorithms for biological data analysis: A survey. IEEE/ACM Trans. Comput. Biol. Bioinformatics 1, 24–45 (2004)
10. Madeira, S., Oliveira, A.: A polynomial time biclustering algorithm for finding approximate expression patterns in gene expression time series. Algorithms for Molecular Biology 4(8) (2009)
11. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Efficient mining of association rules using closed itemset lattices. Inf. Systems 24(1), 25–46 (1999)
12. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Closed sets based discovery of small covers for association rules. Network. and Inf. Systems 3(2), 349–377 (2001)
13. Pasquier, N., Taouil, R., Bastide, Y., Stumme, G., Lakhal, L.: Generating a condensed representation for association rules. Journal of Intelligent Information Systems 24(1), 29–60 (2005)
14. Peeters, R.: The maximum edge biclique problem is NP-complete. Discrete Applied Mathematics 131(3), 651–654 (2003)

15. Ptak, R., Fu, W., Sanders-Beer, B., Dickerson, J., Pinney, J., Robertson, D., Rozanov, M., Katz, K., Maglott, D., Pruitt, K., Dieffenbach, C.: Cataloguing the HIV-1 human protein interaction network. AIDS Research and Human Retroviruses 4(12), 1497–1502 (2008)
16. Shekofteh, M.: A survey of algorithms in FCIM. In: Proc. DSDE, pp. 29–33 (2010)
17. Yahia, S.B., Hamrouni, T., Nguifo, E.M.: Frequent closed itemset based algorithms: A thorough structural and analytical survey. SIGKDD Explorations 8, 93–104 (2006)
18. Zaki, M.J.: Generating non-redundant association rules. In: Proc. SIGKDD, pp. 34–43 (2000)
19. Zaki, M.J., Hsiao, C.J.: CHARM: An efficient algorithm for closed itemset mining. In: Proc. SIAM, pp. 457–473 (2002)

Top-N Minimization Approach for Indicative Correlation Change Mining

Aixiang Li, Makoto Haraguchi, and Yoshiaki Okubo

Graduate School of Information Science and Technology
Hokkaido University
N-14 W-9, Sapporo 060-0814, Japan
{aixiang,makoto,yoshiaki}@kb.ist.hokudai.ac.jp

Abstract. Given a family of transaction databases, various data mining methods for extracting patterns distinguishing one database from another have been extensively studied. This paper particularly focuses on a problem of finding patterns that are more uncorrelated in one database, called a base, and begin to be correlated to some extent in another database, called a target. The detected patterns are not highly correlated at the target. In spite of less correlatedness at the target, the detected patterns are regarded as indicative based on a fact that they are uncorrelated in the base.

We design our search procedure for those patterns by applying optimization strategy under some constraints. More precisely, the objective is to minimize the correlation of patterns at the base under the constraint using upper bound of correlations at the target and the lower bound for the correlation changes over two databases. As there exist many potential solutions, we apply top N control that attains the bottom N correlation values at the base for all the patterns satisfying the constraint.

As we measure the degree of correlation by k-way mutual information, that is monotonically increasing with respect to item addition, we can design a dynamic pruning method for disregarding useless items under the top N control. This contributes for much reducing the computational cost, in whole search process, needed to calculate correlation values over several items as random variables. As a result, we can present a complete search procedure producing only top N solution patterns from a set of all patterns satisfying the constraint, and show its effectiveness and efficiency through experiments.

Keywords: Information-theoretic correlation, Correlation change mining, Top-N minimization for correlation change

1 Introduction

"*Compare and Contrast*" is a general heuristic for finding interesting things that are hard to be realized when observing only single domain or context. The contexts designated by time stamps, topics, categories, and so on, are here assumed given in the form of transaction databases as in *association rule mining* [1]. In

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 102–116, 2012.

case of transaction databases, the standard approaches for contrasting several databases are known as *emerging pattern* [6,18] and *contrast set* [7,12], where the actual contrast observed by changing databases is measured by difference or ratio of supports of itemsets before and after the change. These studies are successful namely for finding itemsets that are frequent after the change and less frequent before the change. On the other hand, in this paper, we still show respect for even non-frequent itemsets, provided they become to have some necessity of composition as sets of items. Those itemsets with increasing necessity may not appear as emerging patterns in the standard sense.

As an itemset is made of elemental items, an itemset of correlated elements can be said to have some necessity of composition as a set. Some studies as [2,5] have discussed changes of positive correlations among items, where each item is considered to represent positive event. However, statistical correlation among items as random variables, deriving a notion known as *minimally correlated itemset* in [3], is much more interesting and realistic. This is simply because it can handle partial correlation which may be positive or negative given some conditioning.

From this viewpoint, the authors have discussed in [19] the problem of finding itemsets as variable set whose correlations increase to some extent after change of databases, where we used *k-way mutual information* [8] instead of χ^2 statistics in [3] to measure the degree of correlations, taking the difference of database sizes into account. In that study, given a pair of databases D_B and D_T, called a *base* before some change and a *target* after the change, respectively, an itemset X is required to satisfy the following constraints under the three parameters δ_B, δ_T and d :

$$\text{(C1)}\ \ cor_{D_B}(X) < \delta_B, \quad \text{(C2)}\ \ cor_{D_B}(X) + d < cor_{D_T}(X) < \delta_T,$$

where $cor_{D_\Sigma}(X)$ denotes the k-way mutual information of X at a database designated by Σ. The constraints intuitively mean that X is not highly correlated at both of the two databases and that the degree of correlation at D_B must increase at least d at D_T after the change. The actual meaning of solution itemsets depends on parameter setting. For instance, it is a natural idea to pay particular attention to

> itemsets which are uncorrelated before the change and whose correlations just begin to increase after the change. In other words, the information brought by them may not be high at the target, but may indicate some beginning of change as they are uncorrelated at the base.

In order to restrict solution itemsets to those in the above sense, δ_B must be small and d must not be high. According to our experiments, even when we assign these values to those parameters, there still remain many possible solutions whose correlation degrees at D_B are just near to the upper bound δ_B, as is shown in Section 6. As long as we prefer more uncorrelated pattern at D_B, any itemset X whose cor_{D_B} value is near to the upper bound δ_B can be disregarded.

From this observation, we propose in this paper not to use δ_B and to apply minimization of $cor_{D_B}(X)$ instead of the constraint $cor_{D_B}(X) < \delta_B$, while we

still hold the constraint $cor_{D_B}(X) + d < cor_{D_T}(X) < \delta_T$. Needless to say, this new specification more suits the problem of finding changes from uncorrelatedness to correlatedness to some extent.

The strategy for applying optimization under some constraints is often used to find plausible and potential solution sets (e.g. [16]). In the studies of emerging pattern mining, a procedure to find top K *jumping emerging patterns* has been presented in [21]. A jumping emerging pattern (JEP for short) is a pattern X satisfying $sup_{D_B}(X) = 0$ and $sup_{D_T}(X) \neq 0$, where $sup_{D_\Sigma}(X)$ denotes the support of itemset X at a database D_Σ. Then, a top-K JEP is defined as a JEP whose support $sup_{D_T}(X)$ is in the top-K largest among possible JEPs. From a strategic point of view, ours and the top-K JEP are dual by corresponding minimization of monotonically increasing information measure at D_B to maximization of anti-monotonic support measure at D_T.

Applying this strategy under the above correspondence, we present in this paper a complete procedure to output only top N itemsets in the sense that whose correlations are in bottom N cor_{D_B}-values among all the itemsets satisfying $cor_{D_B}(X) + d < cor_{D_T}(X) < \delta_T$. Particularly, "*dynamic pruning technique*" for cutting off useless combinations of items is a key to improve computational performance. The uselessness of items is judged based on a tentative itemset Z in a search tree. More precisely, an item i is useless given Z iff some combination of $z \in Z$ and i exceeds any of bottom N cor_{D_B} values v found before the tentative Z. In fact, we have $cor_{D_B}(Z \cup \{i\}) \geq cor_{D_B}(\{z, i\}) > v$ by monotonicity of cor_{D_B}. Therefore, any combination including $Z \cup \{i\}$ never achieves smaller value than the present bottom N values. As the computation of $cor_{D_B}(Z \cup \{i\})$ needs $2^{|Z|+1}$ frequencies, the dynamic pruning for our case plays an essential role for reducing computational resources.

In addition to the above, the search process can be performed without computing itemsets satisfying the constraints beforehand. Both the dynamic pruning and the constraint satisfaction checking are carried out dynamically in the process of building search tree. More precise technical description of these processes will be given in Section 5 using a graph representation.

For two kinds of datasets, collections of Japanese news articles and web documents, effectiveness of our algorithm is verified from the viewpoints of quality of extracted patterns and computational performance. We present several interesting patterns actually extracted by our algorithm but never obtained by contrasting supports or bond-based correlations. Moreover, we show considerable reductions of examined patterns in our search achieved by our top-N minimization approach. Particularly, many of patterns undesirably obtained by the previous method can be excluded with the help of dynamic pruning.

The remainder of this paper is organized as follows. In the next section, we introduce some terminologies used throughout this paper. The correlation based on k-way mutual information and its property are presented in Section 3. In Section 4, we define our problem of mining patterns with top-N minimized correlations. Section 5 describes our depth-first algorithm for extracting top-N patterns based on double-clique search with dynamic update of graph. Our experimental results

are reported in Section 6. In Section 7, we conclude this paper with a summary and future work.

2 Preliminaries

Let $\mathcal{I} = \{i_1, \ldots, i_n\}$ be a set of *items*. An *itemset* X is a subset of $\mathcal{I}$. If $|X| = k$, then X is called a k-itemset. A *transaction* T is a set of items $T \subseteq \mathcal{I}$. A *database* $\mathcal{D}$ is a collection of transactions.

In order to calculate the correlation of an itemset (defined later), we take "presence" and "absence" of an item into account by regarding the set of items $\mathcal{I} = \{i_1, \ldots, i_n\}$ as a set of n *boolean random variables*. Thus, a transaction T is an n-tuple in $\{0, 1\}^n$.

Similar to that in [3], for a k-itemset, we can obtain a *contingency table* with 2^k cell values. These cell values are considered as supports of the itemset, i.e., a k-itemset has 2^k supports. More precisely, given a database $\mathcal{D}$, for an item (1-itemset) a, the number of transactions with a in $\mathcal{D}$ is denoted as $T(a)$ and the number of those without a by $T(\overline{a})$. The probability of a, denoted as $p(a) = p(a = 1)$, is estimated as $p(a) = T(a)/|\mathcal{D}|$. Similarly, we define $p(\overline{a})$ as $p(\overline{a}) = p(a = 0) = T(\overline{a})/|\mathcal{D}| = 1 - p(a)$. Thus, the database $\mathcal{D}$ is partitioned into 2 cells by a with the cell values (supports) $p(a)$ and $p(\overline{a})$ in its contingency table.

Similarly, for a 2-itemset $\{a, b\}$, we have 4 supports in probabilities, $p(ab)$, $p(a\overline{b})$, $p(\overline{a}b)$ and $p(\overline{a}\overline{b})$. The probability $p(ab)$ is estimated as $T(ab)/|\mathcal{D}|$, where $T(ab)$ is the number of transactions with both a and b in $\mathcal{D}$. The probabilities $p(a\overline{b})$, $p(\overline{a}b)$ and $p(\overline{a}\overline{b})$ are estimated in the same way. The definitions can be extended for a k-itemset such that $k \geq 3$. For a k-itemset, thus, $\mathcal{D}$ is partitioned into 2^k cells corresponding to 2^k cell values in its contingency table. Based on the contingency table of an itemset, we can calculate the information-theoretic correlation of an itemset defined in the next section.

3 Correlation Based on k-way Mutual Information

Correlation mining has been developed to reveal the relationships between itemsets. To identify linear functional dependence between numerical random variables, *Pearson's correlation coefficient* [14] is usually used in statistics. However, to handle a dependence between categorical variables, (particularly, the dependence between Boolean variables corresponding to items in this paper), it is poor as stated in [3]. *NMI* [13] has also been proposed to measure a correlation between a pair of quantitative variables. On the other hand, in this paper, we consider a correlation among items - Boolean variables. In order to measure a positive correlation(i.e., co-occurrence) between a pair of itemsets, a notion of *lift* has been proposed [4]. Furthermore, a Jaccard coefficient, *bond* [9,10] is popularly used to measure a correlation based on co-occurrence of items. When the items in patterns are regarded as random Boolean variables, their negative

correlation (i.e., anti-co-occurrence) can also be considered as well as positive one. It is possible to use the bond as a measure.

However, we often observe that some items are *conditionally* correlated actually. The degree of correlation among items is heavily affected by other items or conditions. For example, for three items a, b and c, in some case, we might observe $cor(a; b|c) \gg cor(a; b)$. We consider that there exists a *partial correlation* between a, b under c. It should be noted that the meaning of this partial correlation is different from that in statistics. In this paper, it means a correlation caused by conditional factors. This kind of correlation is hard to be detected by correlation in the sense of co-occurrence as bond.

In order to calculate our *extended correlation*, we regard an item as a Boolean random variable with presence and absence values. Based on information theory, for a pair of items, their correlation is measured by standard *mutual information*. That is, for two items a and b, the correlation between them is calculated as $I(a; b)$. For three or more items, the correlation among them is calculated by an extended mutual information, called *k-way mutual information*.

Definition 1. (Itemset Correlation)
Let $\mathcal{D}$ be a database, $X = \{x_1, \ldots, x_k\}$ a k-itemset. The correlation of X in $\mathcal{D}$, denoted as $cor_{\mathcal{D}}(X)$, is measured by *k-way mutual information* $I(x_1; \cdots ; x_k)$ which is defined as

$$cor_{\mathcal{D}}(X) = I(x_1; \cdots ; x_k)$$
$$= \sum_{x_1 \in \{0,1\}} \cdots \sum_{x_k \in \{0,1\}} p(x_1 x_2 \ldots x_k) \log_2 \frac{p(x_1 x_2 \cdots x_k)}{p(x_1)p(x_2) \cdots p(x_k)},$$

where x_i is an item regarded as a Boolean random variable. ∎

Since the extended correlation of an itemset X measured by k-way mutual information is calculated on probabilities, it is not affected by the size of databases and smaller cell values in the contingency table(affecting factors of χ^2 values). Therefore, we can compare the correlations of an itemset across two contrasted databases.

It is easily proved that for a pair of itemsets X and X' such that $X \subseteq X'$, $cor_{\mathcal{D}}(X) \leq cor_{\mathcal{D}}(X')$ holds. This monotonicity of the extended correlation based on k-way mutual information is applied as one of the pruning mechanisms in our algorithm for mining patterns with top-N minimized correlations, as will be discussed in the following sections.

4 Problem of Mining Top-N Correlation Contrast Sets

In this section, we define our optimization problem of *mining top-N correlation contrast sets* given two contrasted databases. It is noted that although the original notion of "contrast sets" in [7] has been discussed for *two or more* databases, our "contrast sets" in this paper particularly assumes we are given just two databases.

As is similar to [19], our goal is to detect a potential correlation increase from one database $\mathcal{D}_B$ (called the *base*) to the other $\mathcal{D}_T$ (called the *target*) by contrasting the correlations of itemsets. In general, since visibly correlated itemsets are not so interesting, in [19], we have imposed correlation constraints in the base and the target by providing correlation upper bounds δ_B and δ_T, respectively. However, from our experimentation, it has been found that many outputted patterns X with correlations close to δ_B tend to be uninteresting in the sense that they seem to show accidental or coincidental changes. In order to exclude those patterns, we prefer to detect patterns that satisfy the correlation constraint in the target but have less correlations in the base. This computation task is formalized as an optimization problem whose solutions are patterns with top-N minimized correlations in the base still satisfying the correlation constraint and showing sufficient correlation increase in the target. Moreover, to detect more implicit information, we prefer the patterns with *not too much lower entropy* in the target. In the framework of *subspace clustering* [11], a pattern (corresponding to a subspace) with higher correlation and lower entropy is preferred as a distinguished cluster. However, it would be difficult to obtain some potential information from such an explicit pattern. Therefore, we impose an entropy-based constraint on our target patterns. In order to exclude the patterns that involve more rare or common items, we also impose a support-based constraint as in [19], because considering the correlations between these items would not be interesting.

Our problem is now formalized as follows:

Definition 2. (Top-N Correlation Contrast Sets)
Let $\mathcal{D}_B$ and $\mathcal{D}_T$ be a pair of databases to be contrasted[1], δ a threshold for maximum correlation in $\mathcal{D}_T$, d a threshold for minimum correlation increase, ε a threshold for minimum entropy and s a minimum support at p% level.

A *problem of mining top-N correlation contrast sets* is to find every pattern X such that

Constraints on

Correlation: $cor_{\mathcal{D}_T}(X) \leq \delta$,

Correlation Increase: $cor_{\mathcal{D}_T}(X) - cor_{\mathcal{D}_B}(X) \geq d$,

Entropy: $H_{\mathcal{D}_T}(X) \geq \varepsilon$,

Support Constraint: X has support s at p% level in both $\mathcal{D}_B$ and $\mathcal{D}_T$,

Objective Function: $cor_{\mathcal{D}_B}(X)$ is in top-N minimum values among those patterns satisfying the constraints. ∎

5 Extracting Top-N Correlation Contrast Sets with Extended Double-Clique Search

Given the base $\mathcal{D}_B$ and the target $\mathcal{D}_T$ to be contrasted, a threshold δ for maximum correlation in $\mathcal{D}_T$, a threshold d for minimum correlation increase and ε

[1] We assume $\cup_{T \in \mathcal{D}_B} T = \cup_{T \in \mathcal{D}_T} T$, that is, the set of items appeared in $\mathcal{D}_B$ is the same as that in $\mathcal{D}_T$.

a threshold for minimum entropy, in order to find out top-N contrast sets (patterns), we can simply use our previous double-clique search method proposed in [19]. More concretely, we can first enumerate all double-cliques satisfying all the constraints by setting correlation upper bounds for the base $\delta - d$ and the target δ, respectively, and then select those actually ranked in top-N in increasing order of correlations in the base. However, such a method would not be so efficient because the computation of correlations based on k-way mutual information is a bit costly and the number of satisfying double-cliques is large. If we can reduce the number of candidate double-cliques for which we need to actually compute their correlations in the base, we can detect top-N contrast sets more efficiently. Fortunately, our top-N minimization of correlations brings us some effective pruning mechanism.

5.1 Double-Clique Search with Dynamic Update on Base Graph

From Definition 2, a target pattern X must satisfy $cor_{\mathcal{D}_T}(X) \leq \delta$ in the target and have a top-N minimum correlation, say α, in the base. Note that the correlation based on k-way mutual information is monotonically increasing as an itemset is expanded to its supersets. This monotonicity property implies that if $cor_{\mathcal{D}_T}(X) \leq \delta$, then any pair of items in X, a and b, always satisfy $cor_{\mathcal{D}_T}(\{a, b\}) \leq \delta$. It is also implied that in the base for any a and b in X, $cor_{\mathcal{D}_B}(\{a, b\}) \leq \alpha$ holds. Regarding these observations as a necessary condition, we can obtain a target pattern X as a clique in both of the two graphs $G_{\mathcal{D}_T} = (\mathcal{I}, E_{\mathcal{D}_T})$ (the *target graph*) and $G_{\mathcal{D}_B} = (\mathcal{I}, E_{\mathcal{D}_B})$ (the *base graph*), called a *double-clique*, where

$$E_{\mathcal{D}_T} = \{(a, b) \mid a, b \in \mathcal{I} \wedge cor_{\mathcal{D}_T}(\{a, b\}) \leq \delta\} \quad \text{and}$$

$$E_{\mathcal{D}_B} = \{(a, b) \mid a, b \in \mathcal{I} \wedge cor_{\mathcal{D}_B}(\{a, b\}) \leq \alpha\}.$$

It should be noted that while the former graph can be *statically* constructed based on the predefined parameter δ, the latter not, because we have no idea to get an adequate value of α beforehand. Instead of α, however, we can make use of the correlation value of patterns actually found so far as *tentative* top-N.

More precisely speaking, we initially construct the graph $G_{\mathcal{D}_B} = (\mathcal{I}, E_{\mathcal{D}_B})$ such that $E_{\mathcal{D}_B} = \{(a, b) \mid a, b \in \mathcal{I} \wedge cor_{\mathcal{D}_B}(\{a, b\}) \leq \delta - d\}$, where $\delta - d$ gives at most value of correlations in the base which our target patterns can have. Then, we try to extract double-cliques in both $G_{\mathcal{D}_T}$ and $G_{\mathcal{D}_B}$. If a double-clique X as a pattern is found to satisfy all of the the constraints, then X is stored in a list $\mathcal{L}$ which keeps patterns with top-N minimum correlations in the base extracted so far. Once the list $\mathcal{L}$ is filled with patterns having *tentative* top-N minimum correlations, our task is now to detect patterns (cliques) with the correlations no more than $maxcol(\mathcal{L})$, where $maxcol(\mathcal{L})$ is the maximum $cor_{\mathcal{D}_B}$ value of the patterns in $\mathcal{L}$. In other words, we can update the initial graph $G_{\mathcal{D}_B}$ for the base into $G'_{\mathcal{D}_B} = (\mathcal{I}, E'_{\mathcal{D}_B})$ such that $E'_{\mathcal{D}_B} = \{(a, b) \mid a, b \in \mathcal{I} \wedge cor_{\mathcal{D}_B}(\{a, b\}) \leq maxcol(\mathcal{L})\}$. Since $maxcol(\mathcal{L}) \leq \delta - d$, it is easy to see that $G'_{\mathcal{D}_B}$ is sparser than $G_{\mathcal{D}_B}$. Moreover, it should be emphasized that as we find more tentative

target patterns, the value of $maxcol(\mathcal{L})$ decreases monotonically and the graph for the base can be accordingly updated into a sparser one more and more. Such a *dynamic update* on the base graph makes the task of finding double-cliques more efficient, as our computation proceeds.

In the previous method [19], the primary task is to extract double-cliques in $G_{\mathcal{D}_T}$ and $G_{\mathcal{D}_B}$, where the latter graph is just *static* and never updated. Although the double-cliques in the previous method completely covers all of our target patterns with top-N minimum correlations in the base, they also include many hopeless patterns which can never be our targets. With the help of dynamic update on the base graph, we can exclude a lot of those useless double-cliques. It is the main advantage of our new method to be emphasized.

Technically speaking, for the base and target graphs, $G_{\mathcal{D}_B} = (\mathcal{I}, E_{\mathcal{D}_B})$ and $G_{\mathcal{D}_T} = (\mathcal{I}, E_{\mathcal{D}_T})$, a double-clique in both $G_{\mathcal{D}_B}$ and $G_{\mathcal{D}_T}$ is a clique in the graph simply defined as $G = (\mathcal{I}, E_{\mathcal{D}_B} \cap E_{\mathcal{D}_T})$. Every clique in G can be enumerated *systematically*, e.g., based on the basic procedure in [17].

For a graph $G = (V, E)$, let us assume a *total ordering* on $V = \{x_1, \ldots, x_n\}$, where $x_i \prec x_{i+1}$. For a clique $Q \subseteq V$ in G, Q can be expanded into a larger clique by adding a certain vertex, called an *extensible candidate*, to Q. A vertex $x \in V$ is called an extensible candidate of Q if x is adjacent to any vertex in Q. The set of extensible candidates is denoted as $cand(Q)$, that is, $cand(Q) = \{x \in V \mid \forall y \in Q, (x, y) \in E\} = \bigcap_{y \in Q} N_G(y)$, where $N_G(y)$ is the set of vertices adjacent to y in G.

Since for any extensible candidate $x \in cand(Q)$, $Q \cup \{x\}$ always becomes a clique, we can easily generate a larger clique of Q by adding $x \in cand(Q)$ such that $tail(Q) \prec x$, where $tail(Q)$ is the last (maximum) element in Q under the ordering $\prec$. Starting with the initial $Q = \phi$ and the initial set of extensible candidates $cand(Q) = cand(\phi) = V$, we can enumerate all cliques in G by expanding Q with $cand(Q)$ step by step. Although such an expansion process can be done in depth-first or breadth-first manner, we prefer to take the former so that we can apply our method to large-scale datasets with a lot of items.

5.2 Pruning Mechanism

Based on the monotonicity of correlations, if a pattern (clique) X cannot satisfy $cor_{\mathcal{D}_T}(X) \leq \delta$, then we can safely discard X and its expansion (supersets) as useless ones. Whenever such a pattern X is found in our depth-first search of cliques, we can immediately backtrack to another candidate.

In addition to that, a similar pruning based on a tentative value of $maxcol(\mathcal{L})$ is also available in our search. For a tentative $maxcol(\mathcal{L})$, any of our target pattern must show the correlation value no more than $maxcol(\mathcal{L})$ in the base. For a pattern X, therefore, if we find $cor_{\mathcal{D}_B}(X) \leq maxcol(\mathcal{L})$ does not hold, then X and its supersets can never be our target and hence we can prune any expansions of X. Since $maxcol(\mathcal{L})$ is monotonically decreasing, the effect of the pruning based on $maxcol(\mathcal{L})$ becomes powerful more and more as the computation proceeds.

Since considering the correlation of an itemset X that involves rare (or common) items is never interesting, and expanding X cannot cause any significant gain of information (correlation), such itemsets should be excluded. Therefore, we set support constraints on searched itemsets as has been described in [19]. More concretely, given $p\%$ and a minimum support s, an itemset X must have support s at $p\%$-level. If X does not satisfy the condition, then we do not need to expand X.

5.3 Algorithm for Extracting Top-N Correlation Contrast Sets

Based on the above discussion, we can design a complete depth-first double-clique search algorithm for finding top-N correlation contrast sets.

Given a pair of databases, $\mathcal{D}_B$ and $\mathcal{D}_T$, to be contrasted, a correlation upper bound in the target δ, a minimum correlation increase d and an entropy lower bound ε in target, we first construct the base and the target graphs $G_{\mathcal{D}_B}$ and $G_{\mathcal{D}_T}$. Then, we enumerate double-cliques in $G_{\mathcal{D}_B}$ and $G_{\mathcal{D}_T}$ in depth-first manner with the pruning rules.

During the process, we manage a *tentative* list $\mathcal{L}$ of patterns with top-N minimum correlations in the base found so far. For a double-clique X, if all of the imposed constraints are satisfied, then the list $\mathcal{L}$ is adequately updated with X so that it can correctly keep patterns with top-N minimum correlations at that point. Thus, the tentative top-N list is iteratively updated, monotonically decreasing the maximum correlation $maxcol(\mathcal{L})$ in the list. Furthermore, the base graph $G_{\mathcal{D}_B}$ is also updated into a more sparse one based on $maxcol(\mathcal{L})$. This procedure is recursively iterated until no double-clique remains to be examined.

Our algorithm for detecting top-N correlation contrast sets is now summarized in Figure 1. In the pseudo-code, we assume $tail(\emptyset) = \perp$, where the symbol $\perp$ is a (virtual) minimum element in any ordering. Moreover, for a list of patterns $\mathcal{L}$, the function $maxcor(\mathcal{L})$ returns ∞ if the list does not yet contain patterns with tentative top-N minimum correlations.

One might claim that updating the base graph seems to be costly especially in case we have to update it *frequently*. In actual computation, however, we do not have to update the base graph *explicitly*. It is enough to have an $n \times n$-table, $M_{\mathcal{D}_B} = (b_{ij})$, where n is the number of items we are concerned with, that is, $n = |\mathcal{I}|$. For $\mathcal{I} = \{x_1, \ldots, x_n\}$, b_{ij} is just defined as $b_{ij} = cor_{\mathcal{D}_B}(\{x_i, x_j\})$. For a clique X and its extensible candidates $cand(X)$, if we expand X with $x_p \in cand(X)$, $cand(X \cup \{x_p\})$ can be identified by just checking whether for each $x_q \in cand(X)$ such that $p \neq q$, $b_{pq} \leq maxcol(\mathcal{L})$ or not. If it is true, x_q is included in $cand(X \cup \{x_p\})$. Thus, we can enjoy pruning based on dynamic update on the graph without any additional cost.

6 Experimental Results

Our algorithm has been implemented in JAVA and evaluated on two types of databases, *Mainichi News Articles* and *BankSearch*.

```
procedure Main():
  [Output]: the set of top-N correlation contrast sets.
  [Global Variables] L: a list of tentative top-N patterns and G: a graph.
  G ← (I, E_{D_B} ∩ E_{D_T}), where
    E_{D_B} = {(a,b) | a,b ∈ I ∧ cor_{D_B}({a,b}) ≤ δ − d} and
    E_{D_T} = {(a,b) | a,b ∈ I ∧ cor_{D_T}({a,b}) ≤ δ};
  L ← ∅;
  FindTopNCCS(∅, I);
  return L;
```

```
procedure FindTopNCCS(Q, Cand):
  for each x ∈ Cand such that tail(Q) ≺ x do
    NewQ ← Q ∪ {x};
    if cor_{D_T}(NewQ) ≤ δ ∧ cor_{D_B}(NewQ) ≤ maxcor(L) ∧
          NewQ has support s at level of p% in both D_B and D_T then
      if cor_{D_T}(NewQ) − cor_{D_B}(NewQ) ≥ d ∧ H_{D_T}(NewQ) ≥ ε then // NewQ might be a target
        prev ← maxcor(L);
        UpdateTopNList(NewQ);
        if prev ≠ maxcor(L) then // maxcor(L) has been updated
          E_{D_B} ← {(a,b) | a,b ∈ I ∧ cor_{D_B}({a,b}) ≤ maxcor(L)}; // Updating the base graph
        end if
      else
        NewCand ← Cand ∩ N_G(x);
        FindTopNCCS(NewQ, NewCand):
      end if
    end if
  end for
```

```
procedure UpdateTopNList(Q):
  L ← L ∪ {Q};
  if | { |cor_{D_B}(P)| | P ∈ L} | ≥ N then // L contains patterns with tentatively top-N min. cor.
    remove all patterns with M-th minimum correlations from L such that N < M;
  end if
```

Fig. 1. An extended double-clique search algorithm for detecting Top-N correlation contrast sets

Mainichi News Articles is a collection of articles appeared in a Japanese newspaper "*Mainichi*" in 1994 and 1995. Since at the beginning of 1995, there happened "*Hanshin earthquake*" (The South Hyogo Prefecture Earthquake) in Japan, we try to discover potential change before and after the earthquake. After a morphological analysis and removing too rare and too frequent words, we have extracted 406 words as items. Then, we divide the articles into ones in 1994 and those in 1995 as contrasted databases. The former, referred to as $\mathcal{D}_{1994}$, consists of 2343 articles and the latter, referred to as $\mathcal{D}_{1995}$, 9331 articles.

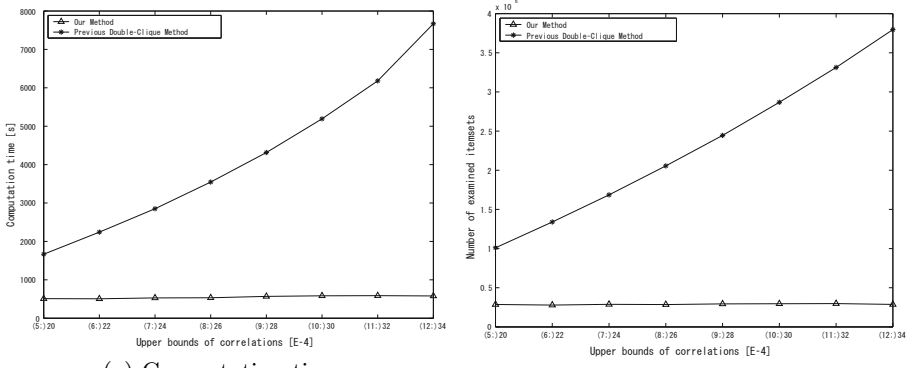
BankSearch is a collection of web documents [15]. From the dataset, we selected two themes "*Banking and Finance*" and "*Sports*" to obtain a pair of databases to be contrasted. The former is referred to as $\mathcal{D}_{Bank}$ and the latter $\mathcal{D}_{Sports}$. Each of them consists of 3000 web documents. After a standard preprocessing (stemming, removing stop-words, too frequent and too infrequent words), we have extracted 585 words.

For these databases, we report the computational performance compared with our previous method on a PC with Core2 Duo E8500 and 4GB main memory. Then, we show some interesting contrast sets actually extracted.

6.1 Computational Performance

For the contrasted databases $\mathcal{D}_{1994}$ and $\mathcal{D}_{1995}$, we compared the efficiency of our method with that of the previous method [19] at 8-pairs of correlation upper bounds $[(\delta_B;)\delta_T]$ in increasing order. These concrete values were determined based on the investigation of correlations between every pair of items. It is noted that in our extended double-clique method, only δ_T is needed.

At the parameter setting $s = 0.005$, $p = 0.25$, $d = 0.001$ and $\varepsilon = 0.8$, computation times for extracting top-100 contrast sets and the numbers of examined itemsets during the search are shown in Figure 2.



(a) Computation times (b) Numbers of examined itemsets

Fig. 2. Computational Performance for *Mainichi News Articles*

In addition, Table 1 shows the numbers of outputted itemsets by our method and the previous method.

Table 1. Numbers of outputted correlation contrast sets on news data

$(\delta_B;)\delta_T$[E-4]	(5;)20	(6;)22	(7;)24	(8;)26	(9;)28	(10;)30	(11;)32	(12;)34
our method	101	102	103	103	102	102	102	101
previous double-clique method	2085	3284	4826	6851	9067	11616	14445	17710

In Figure 2, at higher upper bounds (that is, the double-clique constraint becomes weaker), the performance curves by the previous method tends to increase significantly. On the other hand, the curves by our extended double-clique method keeps at a lower level stably. That is, a considerable number of patterns out of our targets can be excluded in our top-N method. Particularly, from Table 1, it would be expected that such excluded patterns also include many of those with correlations close to δ_B undesirably obtained by the previous method. As has been mentioned before, although top-N patterns can be found by the previous method with an adequate setting of δ_B, our top-N approach can efficiently

detect them without such a difficult parameter adjustment, excluding a significant number of useless patterns.

In our correlation contrast set mining, to avoid the itemsets involving many rare (or common) items, we give a support constraint on itemsets with minimum support s at p% level. Therefore, one might claim that our top-N correlation contrast sets can be found by some fast frequent itemset miner like LCM [20]. That is, we can enumerate frequent itemsets by the miner and then calculate their correlations in both contrasted databases to extract top-N patterns. However, it would be an unpromising approach. For the dataset *Mainichi News Articles*, we have applied LCM under the minimum support equal to the smallest co-occurrence support of our top-N itemsets so that the frequent patterns extracted by LCM can minimally cover all of our top-N correlation contrast sets. As the result, we obtain over 10 billion frequent itemsets even though the computation time is less than 5 minutes. Computing their correlations based on k-way mutual information is clearly expensive.

Although we cannot go into the details due to space limitation, a similar computational performance of our algorithm can also be observed for the dataset of *BankSearch*.

6.2 Extracted Correlation Contrast Sets

For the dataset *Mainichi News Articles*, at the parameter setting $\delta = 0.003$, $s = 0.005$, $p = 0.25$, $d = 0.001$ and $\varepsilon = 0.8$, some of extracted top-100 correlation contrast sets are shown in Table 2.

Table 2. Examples of extracted Top-100 correlation contrast sets for $\mathcal{D}_{1994}$ and $\mathcal{D}_{1995}$

contrast sets	$cor_{\mathcal{D}_{1994}}$	$cor_{\mathcal{D}_{1995}}$	increase
{*recovery, consultation, Itami*}	1.88943e-5	2.94553e-3	2.92663e-3
{*devotion, doctor, Takarazuka*}	2.14402e-5	1.18469e-3	1.16325e-3
{*subway, contact, experience* }	2.29102e-5	1.59180e-3	1.56889e-3
{*restart, unhappy*}	5.62339e-7	1.32056e-3	1.31999e-3
{*photo, consultation*}	2.08342e-6	1.02178e-3	1.01970e-3

It should be noted that most of the top-100 contrast sets are concerned with the disaster and reveal the changes from 1994 to 1995 in Japan. In the table, *Itami* and *Takarazuka* are the damaged cities in the earthquake. Most of the news articles related to those itemsets in the table in $\mathcal{D}_{1995}$ report the rescuing activities and re-construction after the earthquake. Thus, the quality of the output is improved much.

From the first three 3-itemsets, we can observe that there is mainly a *partial correlation* increase in 1995 between the first two component items given the third item. That is, in 1995, the conditional correlation between the first two items x_1 and x_2 given the third item x_3, $cor(x_1; x_2|x_3)$, becomes more greater than $cor(x_1; x_2)$ without the conditioning by x_3. For example, for the itemset {*subway, contact, experience*}, cor(*subway;contact*) is 0.00005 bit actually in

1995. On the other hand, $cor(subway;contact|experience)$ increases to 0.0002 bit in 1995, four times of $cor(subway;contact)$. This means that items *subway* and *contact* are almost not correlated, but get correlated under the conditioning by *experience* after the earthquake. The increase of the partial correlation mainly results in the increase of the correlation of the itemsets $\{subway,\ contact,\ experience\}$ in 1995 since $cor(x_1; x_2; x_3) = cor(x_1; x_3) + cor(x_2; x_3) + cor(x_1; x_2|x_3)$. The news articles related to those terms revealed the fact that the subway administration departments learned much experience from the disaster, and then they determined to improve their contact system in urgent situation during subway re-construction after the earthquake.

In addition to partially correlated itemsets, we also extracted *negatively correlated* contrast sets. For example, $\{restart, unhappy\}$, we checked the *interest* [3], $p(restart, unhappy)/p(restart)p(unhappy) = 0.62$ in 1995. It shows that the two items become more negatively correlated in 1995 with an increase of negative correlation. Similarly, $\{photo, consultation\}$ shows the same change.

By contrasting $\mathcal{D}_{Bank}$ and $\mathcal{D}_{Sports}$, at the parameter setting, $\delta = 0.004$, $s = 0.001$, $p = 0.25$, $d = 0.002$ and $\varepsilon = 1.2$, some examples of top-100 correlation contrast sets are shown in Table 3.

Table 3. Examples of Top-100 correlation contrast sets for $\mathcal{D}_{Bank}$ and $\mathcal{D}_{Sports}$

contrast sets	$cor_{\mathcal{D}_{Bank}}$	$cor_{\mathcal{D}_{Sports}}$	increase
{*employ, commercial, goal*}	7.13674e-5	3.24924e-3	3.17787e-3
{*income, motor, host*}	4.90993e-5	3.10324e-3	3.05414e-3
{*account, race*}	1.06733e-7	2.85277e-3	2.85266e-3
{*stock, league*}	4.35987e-6	2.30343e-3	2.29907e-3
{*winner, power*}	4.65251e-6	3.74283e-3	3.73818e-3

From the first two itemsets, we can observe the partial correlation between the first two component items given the third item, and just the potential increase of their partial correlations brings the increase of the correlation of the itemsets in $\mathcal{D}_{Sports}$. Those documents related to the first itemset mainly discuss the employments of staff and players in soccer clubs and their commercial values. Those related to the second are mainly about the hosted motor games and the incomes of players. The next three itemsets are negatively correlated patterns in $\mathcal{D}_{Bank}$, but become positively correlated ones in $\mathcal{D}_{Sports}$.

Additionally, it should be emphasized that there are about 30% of our extracted contrast sets whose supports (in usual concept) or bonds [9] change little or decrease on the contrary. This means that many of our contrast sets can not be extracted by the emerging pattern mining based on contrasting-supports or contrasting-bonds. This is also a remarkable advantage of our method.

7 Concluding Remark

In this paper, we have discussed an optimization problem of detecting top-N correlation contrast sets. In the problem, we only extracted top-N patterns with

minimized correlations in the base and a minimum correlation increase in the target under the double-clique conditions. We consider that such correlation changes possibly caused by some event and therefore are worth to be detected. From our experimental results, we can find that the qualities of the outputted patterns are improved clearly than that by the previous proposal.

To extract the top-N contrast sets, we developed an extended double-clique algorithm based on the previous static double-clique search with dynamically updating the graph of the base. Our experimental results show that the extended double-clique algorithm works more efficiently than the previous static double-clique method. Especially at higher correlation constraints, the computation times by the previous method increase significantly or even out of memory on the current machine. In comparison, the computation times by our extended double-clique algorithm for extracting top-N patterns keeps at a lower and stable level.

It should be noted that even though we have extracted top-N correlation contrast sets efficiently from the tested databases, the computation of correlation measure, extended mutual information, is still expensive, especially, when the number of items becomes larger. As important future work, it would be worth further investigating correlation measures with monotone properties. Improving efficiency of the algorithm would also be required for a larger scale dataset with many items. Since our method is a general framework, we need to apply it to several actual domains in order to make the method more useful.

References

1. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques, 3rd edn. Morgan Kaufmann (2011)
2. Taniguchi, T., Haraguchi, M.: Discovery of Hidden Correlations in a Local Transaction Database Based on Differences of Correlations. Engineering Application of Artificial Intelligence 19(4), 419–428 (2006)
3. Silverstein, C., Brin, S., Motwani, R.: Beyond Market Baskets: Generalizing Association Rules to Dependence Rules. Data Mining and Knowledge Discovery 2(1), 39–68 (1998)
4. Brin, S., Motwani, R., Ullman, J.D., Tsur, S.: Dynamic Itemset Counting and Implication Rules for Market Basket Data. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data, pp. 265–276 (1997)
5. Yoshioka, M.: NSContrast: An Exploratory News Article Analysis System that Characterizes the Differences between News Sites. In: Proc. of SIGIR 2009, Workshop on Information Access in a Multilingual World, pp. 25–29 (2009)
6. Dong, G., Li, J.: Mining Border Descriptions of Emerging Patterns from Dataset Pairs. Knowledge and Information Systems 8(2), 178–202 (2005)
7. Bay, S.D., Pazzani, M.J.: Detecting Group Differences: Mining Contrast Sets. Data Mining and Knowledge Discovery 5(3), 213–246 (2001)
8. Zhang, X., Pan, F., Wang, W., Nobel, A.: Mining Non-Redundant High Order Correlations in Binary Data. In: Proc. of VLDB, pp. 1178–1188 (2008)
9. Omiecinski, E.: Alternative Interest Measures for Mining Associations in Databases. IEEE Transactions on Knowledge and Data Engineering 15(1), 57–69 (2003)

10. Younes, N.B., Hamrouni, T., Yahia, S.B.: Bridging Conjunctive and Disjunctive Search Spaces for Mining a New Concise and Exact Representation of Correlated Patterns. In: Pfahringer, B., Holmes, G., Hoffmann, A. (eds.) DS 2010. LNCS (LNAI), vol. 6332, pp. 189–204. Springer, Heidelberg (2010)
11. Cheng, C., Fu, A., Zhang, Y.: Entropy-Based Subspace Clustering for Mining Numerical Data. In: Proc. of the 5th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, pp. 84–93 (1999)
12. Novak, P.K., Lavrac, N., Webb, G.I.: Supervised Descriptive Rule Discovery: A Unifying Survey of Contrast Set, Emerging Pattern and Subgroup Mining. Journal of Machine Learning Research 10, 377–403 (2009)
13. Ke, Y.P., Cheng, J., Ng, W.: Mining Quantitative Correlated Patterns Using an Information-Theoretic Approach. In: Proc. of the 12th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, pp. 227–236 (2006)
14. Ke, Y.P., Cheng, J., Ng, W.: Correlation Search in Graph Databases. In: Proc. of the 13th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, pp. 390–399 (2007)
15. Sinka, M.P., Corne, D.W.: A Large Benchmark Dataset for Web Document Clustering. In: Soft Computing Systems: Design, Management and Applications. Frontiers in Artificial Intelligence and Applications, vol. 87, pp. 881–890. IOS Press (2002)
16. Haraguchi, M., Okubo, Y.: Pinpoint Clustering of Web Pages and Mining Implicit Crossover Concepts. In: Usmani, Z.(ed.) Web Intelligence and Intelligent Agents, pp. 391–410. InTech (2010)
17. Tomita, E., Seki, T.: An Efficient Branch-and-Bound Algorithm for Finding a Maximum Clique with Computational Experiments. Journal of Global Optimization 37(1), 95–111 (2007)
18. Li, J., Dong, G., Ramamohanarao, K.: Making Use of the Most Expressive Jumping Emerging Patterns for Classification. Knowledge and Information Systems 3(2), 131–145 (2001)
19. Li, A., Haraguchi, M., Okubo, Y.: Contrasting Correlations by an Efficient Double-Clique Condition. In: Perner, P. (ed.) MLDM 2011. LNCS (LNAI), vol. 6871, pp. 469–483. Springer, Heidelberg (2011)
20. Uno, T., Kiyomi, M., Arimura, H.: LCM ver.3: Collaboration of Array, Bitmap and Prefix Tree for Frequent Itemset Mining. In: Proc. of the 1st Int'l Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations, pp. 77–86. ACM (2005)
21. Terlecki, P., Walczak, K.: Efficient Discovery of Top-K Minimal Jumping Emerging Patterns. In: Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) RSCTC 2008. LNCS (LNAI), vol. 5306, pp. 438–447. Springer, Heidelberg (2008)

Selecting Classification Algorithms with Active Testing

Rui Leite[1], Pavel Brazdil[1], and Joaquin Vanschoren[2]

[1] LIAAD-INESC Porto L.A./Faculty of Economics, University of Porto, Portugal
rleite@fep.up.pt, pbrazdil@inescporto.pt
[2] LIACS - Leiden Institute of Advanced Computer Science,
University of Leiden, Nederlands
joaquin@liacs.nl

Abstract. Given the large amount of data mining algorithms, their combinations (e.g. ensembles) and possible parameter settings, finding the most adequate method to analyze a new dataset becomes an ever more challenging task. This is because in many cases *testing* all possibly useful alternatives quickly becomes prohibitively expensive. In this paper we propose a novel technique, called *active testing*, that intelligently selects the most useful cross-validation tests. It proceeds in a tournament-style fashion, in each round selecting and testing the algorithm that is *most likely to outperform the best algorithm of the previous round* on the new dataset. This 'most promising' competitor is chosen based on a history of prior duels between both algorithms on *similar* datasets. Each new cross-validation test will contribute information to a better estimate of dataset similarity, and thus better predict which algorithms are most promising on the new dataset. We have evaluated this approach using a set of 292 algorithm-parameter combinations on 76 UCI datasets for classification. The results show that active testing will quickly yield an algorithm whose performance is very close to the optimum, after relatively few tests. It also provides a better solution than previously proposed methods.

1 Background and Motivation

In many data mining applications, an important problem is selecting the best algorithm for a specific problem. Especially in classification, there are hundreds of algorithms to choose from. Moreover, these algorithms can be combined into composite learning systems (e.g. ensembles) and often have many parameters that greatly influence their performance. This yields a whole spectrum of methods and their variations, so that *testing* all possible candidates on the given problem, e.g., using cross-validation, quickly becomes prohibitively expensive.

The issue of selecting the right algorithm has been the subject of many studies over the past 20 years [17,3,23,20,19]. Most approaches rely on the concept of *metalearning*. This approach exploits characterizations of datasets and past performance results of algorithms to recommend the best algorithm on the current

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 117–131, 2012.

dataset. The term *metalearning* stems from the fact that we try to learn the function that maps *dataset characterizations* (meta-data) to *algorithm performance estimates* (the target variable).

The earliest techniques considered only the dataset itself and calculated an array of various simple, statistical or information-theoretic properties of the data (e.g., dataset size, class skewness and signal-noise ratio) [17,3]. Another approach, called *landmarking* [2,12], ran simple and fast versions of algorithms (e.g. decision stumps instead of decision trees) on the new dataset and used their performance results to characterize the new dataset. Alternatively, in *sampling landmarks* [21,8,14], the complete (non-simplified) algorithms are run on small samples of the data. A series of sampling landmarks on increasingly large samples represents a partial learning curve which characterizes datasets and which can be used to predict the performance of algorithms significantly more accurately than with classical dataset characteristics [13,14]. Finally, an 'active testing strategy' for sampling landmarks [14] was proposed that actively selects the most informative sample sizes while building these partial learning curves, thus reducing the time needed to compute them.

Motivation. All these approaches have focused on dozens of algorithms at most and usually considered only default parameter settings. Dealing with hundreds, perhaps thousands of algorithm-parameter combinations[1], provides a new challenge that requires a new approach. First, distinguishing between hundreds of subtly different algorithms is significantly harder than distinguishing between a handful of very different ones. We would need many more data characterizations that relate the effects of certain parameters on performance. On the other hand, the latter method [14] has a scalability issue: it requires that pairwise comparisons be conducted between algorithms. This would be rather impractical when faced with hundreds of algorithm-parameter combinations.

To address these issues, we propose a quite different way to characterize datasets, namely through the *effect that the dataset has on the relative performance of algorithms run on them*. As in landmarking, we use the fact that each algorithm has its own learning bias, making certain assumptions about the data distribution. If the learning bias 'matches' the underlying data distribution of a particular dataset, it is likely to perform well (e.g., achieve high predictive accuracy). If it does not, it will likely under- or overfit the data, resulting in a lower performance.

As such, we *characterize a dataset based on the pairwise performance differences between algorithms run on them*: if the same algorithms win, tie or lose against each other on two datasets, then the data distributions of these datasets are likely to be similar as well, at least in terms of their effect on learning performance. It is clear that the more algorithms are used, the more accurate the characterization will be. While we cannot run all algorithms on each new dataset

[1] In the remainder of this text, when we speak of *algorithms*, we mean *fully-defined algorithm instances* with fixed components (e.g., base-learners, kernel functions) and parameter settings.

because of the computational cost, we can run a fair amount of CV tests to get a reasonably good idea of which prior datasets are most similar to the new one.

Moreover, we can use these same performance results to establish which (yet untested) algorithms are likely to perform well on the new dataset, i.e., those algorithms that outperformed or rivaled the currently best algorithm on similar datasets in the past. As such, we can intelligently select the most promising algorithms for the new dataset, run them, and then use their performance results to gain increasingly better estimates of the most similar datasets and the most promising algorithms.

Key concepts. There are two key concepts used in this work. The first one is that of the *current best candidate algorithm* which may be challenged by other algorithms in the process of finding an even better candidate.

The second is the pairwise performance difference between two algorithm run on the same dataset, which we call *relative landmark*. A collection of such relative landmarks represents a history of previous 'duels' between two algorithms on prior datasets. The term itself originates from the study of landmarking algorithms: since absolute values for the performance of landmarkers vary a lot depending on the dataset, several types of *relative* landmarks have been proposed, which basically capture the relative performance difference between two algorithms [12]. In this paper, we extend the notion of relative landmarks to *all* (including non-simplified) classification algorithms.

The history of previous algorithm duels is used to select the most promising challenger for the current best candidate algorithm, namely the method that most convincingly outperformed or rivaled the current champion on prior datasets *similar* to the new dataset.

Approach. Given the current best algorithm and a history of relative landmarks (duels), we can start a tournament game in which, in each round, the current best algorithm is compared to the next, most promising contender. We select the most promising challenger as discussed above, and run a CV test with this algorithm. The winner becomes the new current best candidate, the loser is removed from consideration. We will discuss the exact procedure in Section 3.

We call this approach *active testing* (AT)[2], as it actively selects the most interesting CV tests instead of passively performing them one by one: in each iteration the *best competitor* is identified, which determines a new CV test to be carried out. Moreover, the same result will be used to further characterize the new dataset and more accurately estimate the similarity between the new dataset and all prior datasets.

Evaluation. By intelligently selecting the most promising algorithms the test on the new dataset, we can more quickly discover an algorithm that performs very well. Note that running a selection of algorithms is typically done anyway

[2] Note that while the term 'active testing' is also used in the context of actively selected sampling landmarks [14], there is little or no relationship to the approach described here.

to find a suitable algorithm. Here, we optimize and automate this process using historical performance results of the candidate algorithms on prior datasets.

While we cannot possibly guarantee to return the absolute best algorithm without performing all possible CV tests, we can return an algorithm whose performance is either identical or very close to the truly best one. The difference between the two can be expressed in terms of a *loss*. Our aim is thus to *minimize* this loss using a *minimal number of tests*, and we will evaluate our technique as such.

In all, the research hypothesis that we intend to prove in this paper is: *Relative landmarks provide useful information on the similarity of datasets and can be used to efficiently predict the most promising algorithms to test on new datasets.* We will test this hypothesis by running our active testing approach in a leave-one-out fashion on a large set of CV evaluations testing 292 algorithms on 76 datasets. The results show that our AT approach is indeed effective in finding very accurate algorithms in a very limited number of tests.

Roadmap. The remainder of this paper is organized as follows. First, we formulate the concepts of relative landmarks in Section 2 and active testing in Section 3. Next, Section 4 presents the empirical evaluation and Section 5 presents an overview of some work in other related areas. The final section presents conclusions and future work.

2 Relative Landmarks

In this section we formalize our definition of relative landmarks, and explain how are used to identify the most promising competitor for a currently best algorithm.

Given a set of classification algorithms and some new classification dataset d_{new}, the aim is to identify the potentially best algorithm for this task with respect to some given performance measure M (e.g., accuracy, AUC or rank). Let us represent the performance of algorithm a_i on dataset d_{new} as $M(a_i, d_{new})$. As such, we need to identify an algorithm $a*$, for which the performance measure is maximal, or $\forall a_i M(a*, d_{new}) \geq M(a_i, d_{new})$. The decision concerning $\geq$ (i.e. whether $a*$ is at least as good as a_i) may be established using either a statistical significance test or a simple comparison.

However, instead of searching exhaustively for $a*$, we aim to find a near-optimal algorithm, $\hat{a*}$, which has a high probability $P(M(\hat{a*}, d_{new}) \geq M(a_i, d_{new}))$ to be optimal, ideally close to 1.

As in other work that exploits metalearning, we assume that $\hat{a*}$ is likely better than a_i on dataset d_{new} if it was found to be better on a similar dataset d_j (for which we have performance estimates):

$$P(M(\hat{a*}, d_{new}) \geq M(a_i, d_{new})) \sim P(M(\hat{a*}, d_j) \geq M(a_i, d_j)) \quad (1)$$

The latter estimate can be maximized by going through all algorithms and identifying the algorithm $\hat{a*}$ that satisfies the $\geq$ constraint in a maximum number of cases. However, this requires that we know which datasets d_j are most similar to

d_{new}. Since our definition of similarity requires CV tests to be run on d_{new}, but we cannot run all possible CV tests, we use an iterative approach in which we repeat this scan for $\hat{a*}$ in every round, using only the datasets d_j that seem most similar at that point, as dataset similarities are recalculated after every CV test.

Initially, having no information, we deem all datasets to be similar to d_{new}, so that $\hat{a*}$ will be the globally best algorithm over all prior datasets. We then call this algorithm the *current best algorithm* a_{best} and run a CV test to calculate its performance on d_{new}. Based on this, the dataset similarities are recalculated (see Section 3), yielding a possibly different set of datasets d_j. The best algorithm on this new set becomes the *best competitor* a_k (different from a_{best}), calculated by counting the number of times that $M(a_k, d_j) > M(a_{best}, d_j)$, over all datasets d_j.

We can further refine this method by taking into account how large the performance differences are: the larger a difference was in the past, the higher chances are to obtain a large gain on the new dataset. This leads to the notion of relative landmarks RL, defined as:

$$RL(a_k, a_{best}, d_j) = i(M(a_k, d_j) > M(a_{best}, d_j)) * (M(a_k, d_j) - M(a_{best}, d_j)) \quad (2)$$

The function $i(test)$ returns 1 if the $test$ is true and 0 otherwise. As stated before, this can be a simple comparison or a statistical significance test that only returns 1 if a_k performs significantly better than a_{best} on d_j. The term RL thus expresses how much better a_k is, relative to a_{best}, on a dataset d_j. Experimental tests have shown that this approach yields much better results than simply counting the number of wins.

Up to now, we are assuming a dataset d_j to be either similar to d_{new} or not. A second refinement is to use a gradual (non-binary) measure of similarity $Sim(d_{new}, d_j)$ between datasets d_{new} and d_j. As such, we can weigh the performance difference between a_k and a_{best} on d_j by how similar d_j is to d_{new}. Indeed, the more similar the datasets, the more informative the performance difference is. As such, we aim to optimize the following criterion:

$$a_k = \arg\max_{a_i} \sum_{dj \in D} RL(a_i, a_{best}, d_j) * Sim(d_{new}, d_j)) \quad (3)$$

in which D is the set of all prior datasets d_j.

To calculate the similarity $Sim()$, we use the outcome of each CV test on d_{new} and compare it to the outcomes on d_j.

In each iteration, with each CV test, we obtain a new evaluation $M(a_i, d_{new})$, which is used to recalculate all similarities $Sim(d_{new}, d_j)$. In fact, we will compare four variants of $Sim()$, which are discussed in the next section. With this, we can recalculate equation 3 to find the next best competitor a_k.

3 Active Testing

In this section we describe the active testing (AT) approach, which proceeds according to the following steps:

1. Construct a global ranking of a given set of algorithms using performance information from past experiments (metadata).
2. Initiate the iterative process by assigning the top-ranked algorithm as a_{best} and obtain the performance of this algorithm on d_{new} using a CV test.
3. Find the most promising competitor a_k for a_{best} using relative landmarks and all previous CV tests on d_{new}.
4. Obtain the performance of a_k on d_{new} using a CV test and compare with a_{best}. Use the winner as the current best algorithm, and eliminate the losing algorithm.
5. Repeat the whole process starting with step 3 until a stopping criterium has been reached. Finally, output the current a_{best} as the overall winner.

Step 1 - Establish a Global Ranking of Algorithms. Before having run any CV tests, we have no information on the new dataset d_{new} to define which prior datasets are similar to it. As such, we naively assume that all prior datasets are similar. As such, we generate a global ranking of all algorithms using the performance results of all algorithms on all previous datasets, and choose the top-ranked algorithm as our initial candidate a_{best}. To illustrate this, we use a toy example involving 6 classification algorithms, with default parameter settings, from Weka [10] evaluated on 40 UCI datasets [1], a portion of which is shown in Table 1.

As said before, our approach is entirely independent from the exact evaluation measure used: the most appropriate measure can be chosen by the user in function of the specific data domain. In this example, we use *success rate (accuracy)*, but any other suitable measure of classifier performance, e.g. AUC (area under the ROC curve), precision, recall or F1 can be used as well.

Each accuracy figure shown in Table 1 represents the mean of 10 values obtained in 10-fold cross-validation. The ranks of the algorithms on each dataset are shown in parentheses next to the accuracy value. For instance, if we consider dataset *abalone*, algorithm MLP is attributed rank 1 as its accuracy is highest on this problem. The second rank is occupied by $LogD$, etc.

The last row in the table shows the *mean rank* of each algorithm, obtained by averaging over the ranks of each dataset: $R_{a_i} = \frac{1}{n}\sum_{d_j=1}^{n} R_{a_i,d_j}$, where R_{a_i,d_j} represents the rank of algorithm a_i on dataset d_j and n the number of datasets. This is a quite common procedure, often used in machine learning to assess how a particular algorithm compares to others [5].

The mean ranks permit us to obtain a global ranking of candidate algorithms, CA. In our case, $CA = \langle MLP, J48, JRip, LogD, IB1, NB\rangle$. It must be noted that such a ranking is not very informative in itself. For instance, statistical significance tests are needed to obtain a truthful ranking. Here, we only use this global ranking CA are a starting point for the iterative procedure, as explained next.

Step 2 - Identify the Current Best Algorithm. Indeed, global ranking CA permits us to identify the top-ranked algorithm as our initial best candidate algorithm a_{best}. In Table 1, $a_{best} = MLP$. This algorithm is then evaluated using a CV test to establish its performance on the new dataset d_{new}.

Table 1. Accuracies and ranks (in parentheses) of the algorithms 1-nearest neighbor (IB1), C4.5 (J48), RIPPER (JRip), LogisticDiscriminant (LogD), MultiLayerPerceptron (MLP) and naive Bayes (NB) on different datasets and their mean rank

Datasets	IB1	J48	JRip	LogD	MLP	NB
abalone	.197 (5)	.218 (4)	.185 (6)	.259 (2)	.266 (1)	.237 (3)
acetylation	.844 (1)	.831 (2)	.829 (3)	.745 (5)	.609 (6)	.822 (4)
adult	.794 (6)	.861 (1)	.843 (3)	.850 (2)	.830 (5)	.834 (4)
...	...	...	...	...	...	...
Mean rank	4.05	2.73	3.17	3.74	2.54	4.78

Step 3 - Identify the Most Promising Competitor. In the next step we identify a_k, the *best competitor* of a_{best}. To do this, all algorithms are considered one by one, except for a_{best} and the eliminated algorithms (see step 4).

For each algorithm we analyze the information of past experiments (metadata) to calculate the relative landmarks, as outlined in the previous section. As equation 3 shows, for each a_k, we sum up all relative landmarks involving a_{best}, weighted by a measure of similarity between dataset d_j and the new dataset d_{new}. The algorithm a_k that achieves the highest value is the most promising competitor in this iteration. In case of a tie, the competitor that appears first in ranking CA is chosen.

To calculate $Sim(d_{new}, d_j)$, the similarity between d_j and d_{new}, we have explored four different variants, AT0,AT1,ATWs,ATx, described below.

AT0 is a base-line method which ignores dataset similarity. It always returns a similarity value of 1 and so all datasets are considered similar. This means that the best competitor is determined by summing up the values of the relative landmarks.

AT1 method works as AT0 at the beginning, when no test have been carried out on d_{new}. In all subsequent iterations, this method estimates dataset similarity using only the most recent CV test. Consider the algorithms listed in Table 1 and the ranking CA. Suppose we started with algorithm MLP as the current best candidate. Suppose also that in the next iteration $LogD$ was identified as the best competitor, and won from MLP in the CV test: $(M(LogD, d_{new}) > M(MLP, d_{new}))$. Then, in the subsequent iteration, all prior datasets d_j satisfying the condition $M(LogD, d_j) > M(MLP, d_j)$ are considered similar to d_{new}. In general terms, suppose that the last test revealed that $M(a_k, d_{new}) > M(a_{best}, d_{new})$, then $Sim(d_{new}, d_j)$ is 1 if also $M(a_k, d_j) > M(a_{best}, d_j)$, and 0 otherwise. The similarity measure determines which RL's are taken into account when summing up their contributions to identify the next best competitor.

Another variant of AT1 could use the difference between $RL(a_k, a_{best}, d_{new})$ and $RL(a_k, a_{best}, d_j)$, normalized between 0 and 1, to obtain a real-valued (non-binary) similarity estimate $Sim(d_{new}, d_j)$. In other words, d_j is *more similar* to d_{new} if the relative performance difference between a_k and a_{best} is about as large on both d_j and d_{new}. We plan to investigate this in our future work.

ATWs is similar to AT1, but instead of only using the last test, it uses *all* CV tests carried out on the new dataset, and calculates the Laplace-corrected ratio of corresponding results. For instance, suppose we have conducted 3 tests on d_{new}, thus yielding 3 pairwise algorithm comparisons on d_{new}. Suppose that 2 tests had the same result on dataset d_j (i.e. $M(a_x, d_{new}) > M(a_y, d_{new})$ and $M(a_x, d_j) > M(a_y, d_j)$), then the frequency of occurrence is 2/3, which is adjusted by Laplace correction to obtain an estimate of probability $(2+1)/(3+2)$. As such, $Sim(d_{new}, d_j) = \frac{3}{5}$.

ATx is similar to ATWs, but requires that all pairwise comparisons yield the same outcome. In the example used above, it will return $Sim(d_{new}, d_j) = 1$ only if all three comparisons lead to same result on both datasets and 0 otherwise.

Step 4 - Determine which of the Two Algorithms is Better. Having found a_k, we can now run a CV test and compare it with a_{best}. The winner (which may be either the current best algorithm or the competitor) is used as the new current best algorithm in the new round. The losing algorithm is eliminated from further consideration.

Step 5 - Repeat the Process and Check the Stopping Criteria. The whole process of identifying the best competitor (step 3) of a_{best} and determining which one of the two is better (step 4) is repeated until a stopping criterium has been reached. For instance, the process could be constrained to a fixed number of CV tests: considering the results presented further on in Section 4, it would be sufficient to run at most 20% of all possible CV tests. Alternatively, one could impose a fixed CPU time, thus returning the best algorithm in h hours, as in budgeted learning. In any case, until aborted, the method will keep choosing a new competitor in each round: there will always be a next best competitor. In this respect our system differs from, for instance, hill climbing approaches which can get stuck in a local minimum.

Discussion - Comparison with Active Learning: The term *active testing* was chosen because the approach shares some similarities with *active learning* [7]. The concern of both is to speed up the process of improving a given performance measure. In active learning, the goal is to select the most informative data point to be labeled next, so as to improve the predictive performance of a supervised learning algorithm with a minimum of (expensive) labelings. In active testing, the goal is to select the most informative CV test, so as to improve the prediction of the best algorithm on the new dataset with a minimum of (expensive) CV tests.

4 Empirical Evaluation

4.1 Evaluation Methodology and Experimental Set-up

The proposed method AT was evaluated using a *leave-one-out* method [18]. The experiments reported here involve D datasets and so the whole procedure was repeated D times. In each cycle, all performance results on one dataset were left

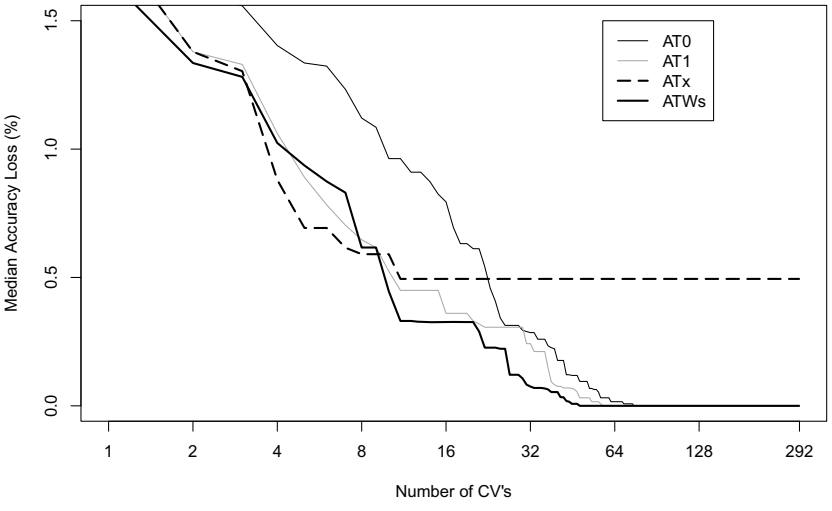


Fig. 1. Median loss as a function of the number of CV tests

out for testing and the results on the remaining $D - 1$ datasets were used as metadata to determine the best candidate algorithm.

This study involved 292 algorithm (algorithm-parameter combinations), which were extracted from the experiment database for machine learning (ExpDB) [11,22]. This set includes many different algorithms from the Weka platform [10], which were varied by assigning different values to their most important parameters. It includes SMO (a support vector machine, SVM), MLP (Multilayer Perceptron), J48 (C4.5), and different types of ensembles, including RandomForest, Bagging and Boosting. Moreover, different SVM kernels were used with their own parameter ranges and all non-ensemble learners were used as base-learners for the ensemble learners mentioned above. The 76 datasets used in this study were all from UCI [1]. A complete overview of the data used in this study, including links to all algorithms and datasets can be found on `http://expdb.cs.kuleuven.be/ref/blv11`.

The main aim of the test was to prove the research hypothesis formulated earlier: relative landmarks provide useful information for predicting the most promising algorithms on new datasets. Therefore, we also include two baseline methods:

TopN has been described before (e.g. [3]). It also builds a ranking of candidate algorithms as described in step 1 (although other measures different from mean rank could be used), and only runs CV tests on the first N algorithms. The overall winner is returned.

Rand simply selects N algorithms at random from the given set, evaluates them using CV and returns the one with the best performance. It is repeated 10 times with different random seeds and the results are averaged.

Since our AT methods are iterative, we will restart TopN and Rand N times, with N equal to the number of iterations (or CV tests).

To evaluate the performance of all approaches, we calculate the *loss* of the currently best algorithm, defined as $M(a_{best}, d_{new}) - M(a*, d_{new})$, where a_{best} represents the currently best algorithm, $a*$ the best possible algorithm and $M(.)$ represents the performance measure (success rate).

4.2 Results

By aggregating the results over D datasets, we can track the *median loss* of the recommended algorithm as a function of the number of CV tests carried out. The results are shown in Figure 1. Note that the number of CV tests is plotted on a logarithmic scale.

First, we see that $ATWs$ and $AT1$ perform much better than $AT0$, which indicates that it is indeed useful to include dataset similarity. If we consider a particular level of loss (say 0.5%) we note that these variants require much fewer CV tests than $AT0$. The results also indicate that the information associated with relative landmarks obtained on the new dataset is indeed valuable. The median loss curves decline quite rapidly and are always below the $AT0$ curve. We also see that after only 10 CV tests (representing about 3% of all possible tests), the median loss is less than 0.5%. If we continue to 60 tests (about 20% of all possible tests) the median loss is near 0.

Also note that $ATWs$, which uses all relative landmarks involving a_{best} and d_{new}, does not perform much better than $AT1$, which only uses the most recent CV test. This results suggests that, when looking for the most promising competitor, the latest test is more informative than the previous ones.

Method ATx, the most restrictive approach, only considers prior datasets on which *all* relative landmarks including a_{best} obtained similar results. As shown in Figure 1, this approach manages to reduce the loss quite rapidly, and competes well with the other variants in the initial region. However, after achieving a minimum loss in the order of 0.5%, there are no more datasets that fulfill this restriction, and consequently no new competitor can be chosen, causing it to stop. The other two methods, $ATWs$ and $AT1$, do not suffer from this shortcoming.

$AT0$ was also our best baseline method. To avoid overloading Figure 1, we show this separately in Figure 2. Indeed, $AT0$ is clearly better than the random choice method *Rand*. Comparing $AT0$ to $TopN$, we cannot say that one is clearly better than the other overall, as the curves cross. However, it is clear that $TopN$ looses out if we allow more CV tests, and that it is not competitive with the more advanced methods such as $AT1$ and $ATWs$.

The curves for mean loss (instead of median loss) follow similar trends, but the values are 1-2% worse due to outliers (see Fig. 3 relative to method $AT1$). Besides, this figure shows also the curves associated with quartiles of 25% and 75% for $AT1$. As the number of CV tests increases, the distance between the two curves decreases and approaches the median curve. Similar behavior has been observed for $ATWs$, but we skip the curves in this text.

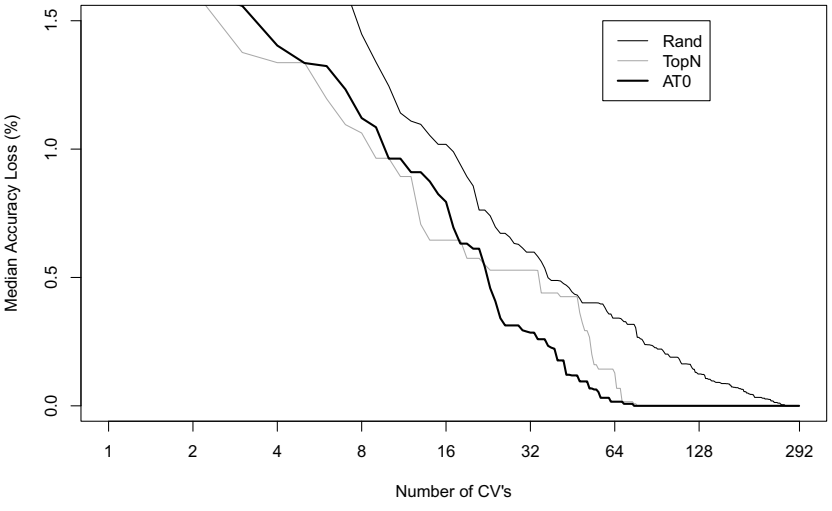


Fig. 2. Median loss of AT0 and the two baseline methods

Algorithm Trace. It is interesting to trace the iterations carried out for one particular dataset. Table 2 shows the details for method $AT1$, where *abalone* represents the new dataset. Column 1 shows the number of the iteration (thus the number of CV tests). Column 2 shows the most promising competitor a_k chosen in each step. Column 3 shows the index of a_k in our initial ranking CA, and column 4 the index of a_{best}, the *new best* algorithm after the CV test has been performed. As such, if the values in column 3 and 4 are the same, then the most promising competitor has won the duel. For instance, in step 2, $SMO.C.1.0.Polynomial.E.3$, i.e. SVM with complexity constant set to 1.0 and a 3rd degree polynomial kernel, (index 96) has been identified as the best competitor to be used (column 2), and after the CV test, it has won against $Bagging.I.75..100.PART$, i.e. Bagging with a high number of iterations (between 75 and 100) and PART as a base-learner. As such, it wins this round and becomes the new a_{best}. Columns 5 and 6 show the *actual* rank of the competitor and the winner on the *abalone* dataset. Column 7 shows the loss compared to the optimal algorithm and the final column shows the number of datasets whose similarity measure is 1.

We observe that after only 12 CV tests, the method has identified an algorithm with a very small loss of 0.2%: $Bagging.I.25..50.MultilayerPerceptron$, i.e. Bagging with relatively few iterations but with a MultiLayerPerceptron base-learner.

Incidentally, this dataset appears to represent a quite atypical problem: the truly best algorithm, $SMO.C.1.0.RBF.G.20$, i.e. SVM with an RBF kernel with kernel width (gamma) set to 20, is ranked globally as algorithm 246 (of all 292). AT1 identifies it after 177 CV tests.

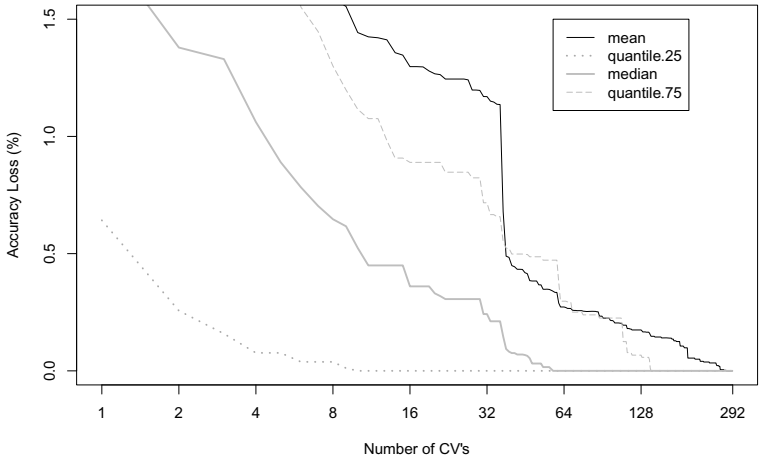


Fig. 3. Loss of $AT1$ as a function of the number of CV tests

5 Related Work in Other Scientific Areas

In this section we briefly cover some work in other scientific areas which is related to the problem tackled here and could provide further insight into how to improve the method.

One particular area is *experiment design* [6] and in particular *active learning*. As discussed before, the method described here follows the main trends that have been outlined in this literature. However, there is relatively little work on active learning for ranking tasks. One notable exception is [15], who use the notion of *Expected Loss Optimization (ELO)*. Another work in this area is [4], whose aim was to identify the most interesting substances for drug screening using a minimum number of tests. In the experiments described, the authors have focused on the top-10 substances. Several different strategies were considered and evaluated. Our problem here is not ranking, but rather simply finding the best item (algorithm), so this work is only partially relevant.

Another relevant area is the so called *multi-armed bandit problem (MAB)* studied in statistics and machine learning [9,16]. This problem is often formulated in a setting that involves a set of traditional slot machines. When a particular lever is pulled, a reward is provided from a distribution associated with that specific lever. The bandit problem is formally equivalent to a one-state Markov decision process. The aim is to minimize *regret* after T rounds, which is defined as a difference between the reward sum associated with an optimal strategy and the sum of collected rewards. Indeed, pulling a lever can be compared to carrying out a CV test on a given algorithm. However, there is one fundamental difference between MAB and our setting: whereas in MAB the aim is to maximize the sum

Table 2. Trace of the steps taken by $AT1$ in the search for the supposedly best algorithm for the *abalone* dataset

CV test	Algorithm used (current best competitor, a_k)	CA a_k	CA new a_{best}	abalone a_k	abalone new a_{best}	Loss (%)	D size
1	Bagging.I.75..100.PART	1	1	75	75	1.9	75
2	SMO.C.1.0.Polynomial.E.3	96	96	56	56	1.6	29
3	AdaBoostM1.I.10.MultilayerPerceptron	92	92	47	47	1.5	34
4	Bagging.I.50..75.RandomForest	15	92	66	47	1.5	27
...	...	...	...	...	...	...	...
10	LMT	6	6	32	32	1.1	45
11	LogitBoost.I.10.DecisionStump	81	6	70	32	1.1	51
12	Bagging.I.25..50.MultilayerPerceptron	12	12	2	2	0.2	37
13	LogitBoost.I.160.DecisionStump	54	12	91	2	0.2	42
...	...	...	...	...	...	...	...
177	SMO.C.1.0.RBF.G.20	246	246	1	1	0	9

of collected rewards, our aim it to maximize *one* reward, i.e. the reward associated with identifying the best algorithm. So again, this work is only partially relevant.

To the best of our knowledge, no other work in this area has addressed the issue of how to select a suitable algorithm from a large set of candidates.

6 Significance and Impact

In this paper we have addressed the problem of selecting the best classification algorithm for a specific task. We have introduced a new method, called *active testing*, that exploits information concerning past evaluation results (metadata), to recommend the best algorithm using a limited number of tests on the new dataset.

Starting from an initial ranking of algorithms on previous datasets, the method runs additional CV evaluations to test several competing algorithms on the new dataset. However, the aim is to reduce the number of tests to a minimum. This is done by carefully selecting which tests should be carried out, using the information of both past and present algorithm evaluations represented in the form of relative landmarks.

In our view this method incorporates several innovative features. First, it is an iterative process that uses the information in each CV test to find the most promising next test based on a history of prior 'algorithm duels'. In a tournament-style fashion, it starts with a current best (parameterized) algorithm, selects the most promising rival algorithm in each round, evaluates it on the given problem, and eliminates the algorithm that performs worse. Second, it continually focuses on the most similar prior datasets: those where the algorithm duels had a similar outcome to those on the new dataset.

Four variants of this basic approach, differing in their definition of algorithm similarity, were investigated in a very extensive experiment setup involving 292 algorithm-parameter combinations on 76 datasets. Our experimental results show that particularly versions $ATWs$ and $AT1$ provide good recommendations using a small number of CV tests. When plotting the median loss as a function of the number of CV tests (Fig. 1), it shows that both outperform all other variants and baseline methods. They also outperform $AT0$, indicating that algorithm similarity is an important aspect.

We also see that after only 10 CV tests (representing about 3% of all possible tests), the median loss is less than 0.5%. If we continue to 60 tests (about 20% of all possible tests) the median loss is near 0. Similar trends can be observed when considering mean loss.

The results support the hypothesis that we have formulated at the outset of our work, that relative landmarks are indeed informative and can be used to suggest the best contender. If this is procedure is used iteratively, it can be used to accurately recommend a classification algorithm after a very limited number of CV tests.

Still, we believe that the results could be improved further. Classical information-theoretic measures and/or sampling landmarks could be incorporated into the process of identifying the most similar datasets. This could lead to further improvements and forms part of our future plans.

References

1. Blake, C.L., Merz, C.J.: UCI repository of machine learning databases (1998)
2. Pfahringer, B., Bensussan, H., Giraud-Carrier, C.: Meta-learning by landmarking various learning algorithms. In: Proceedings of the 17th Int. Conf. on Machine Learning (ICML 2000), Stanford, CA (2000)
3. Brazdil, P., Soares, C., Costa, J.: Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. Machine Learning 50, 251–277 (2003)
4. De Grave, K., Ramon, J., De Raedt, L.: Active learning for primary drug screening. In: Proceedings of Discovery Science. Springer (2008)
5. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research 7, 1–30 (2006)
6. Fedorov, V.: Theory of Optimal Experiments. Academic Press, New York (1972)
7. Freund, Y., Seung, H., Shamir, E., Tishby, N.: Selective sampling using the query by committee algorithm. Machine Learning 28, 133–168 (1997)
8. Fürnkranz, J., Petrak, J.: An evaluation of landmarking variants. In: Proceedings of the ECML/PKDD Workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning (IDDM 2001), pp. 57–68. Springer (2001)
9. Gittins, J.: Multi-armed bandit allocation indices. Wiley Interscience Series in Systems and Optimization. John Wiley & Sons, Ltd. (1989)
10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explor. Newsl. 11(1), 10–18 (2009)
11. Blockeel, H.: Experiment Databases: A Novel Methodology for Experimental Research. In: Bonchi, F., Boulicaut, J.-F. (eds.) KDID 2005. LNCS, vol. 3933, pp. 72–85. Springer, Heidelberg (2006)

12. Fürnkranz, J., Petrak, J.: An evaluation of landmarking variants. In: Carrier, C., Lavrac, N., Moyle, S. (eds.) Working Notes of ECML/PKDD 2000 Workshop on Integration Aspects of Data Mining, Decision Support and Meta-Learning (2001)
13. Leite, R., Brazdil, P.: Predicting relative performance of classifiers from samples. In: ICML 2005: Proceedings of the 22nd International Conference on Machine Learning, pp. 497–503. ACM Press, New York (2005)
14. Leite, R., Brazdil, P.: Active testing strategy to predict the best classification algorithm via sampling and metalearning. In: Proceedings of the 19th European Conference on Artificial Intelligence, ECAI 2010 (2010)
15. Long, B., Chapelle, O., Zhang, Y., Chang, Y., Zheng, Z., Tseng, B.: Active learning for rankings through expected loss optimization. In: Proceedings of the SIGIR 2010. ACM (2010)
16. Mahajan, A., Teneketzis, D.: Multi-armed bandit problems. In: Castanon, D.A., Cochran, D., Kastella, K. (eds.) Foundations and Applications of Sensor Management. Springer (2007)
17. Michie, D., Spiegelhalter, D.J., Taylor, C.C.: Machine Learning, Neural and Statistical Classification. Ellis Horwood (1994)
18. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
19. Rice, J.R.: The algorithm selection problem. Advances in Computers, vol. 15, pp. 65–118. Elsevier (1976)
20. Smith-Miles, K.A.: Cross-disciplinary perspectives on meta-learning for algorithm selection. ACM Comput. Surv. 41(1), 1–25 (2008)
21. Soares, C., Petrak, J., Brazdil, P.: Sampling-Based Relative Landmarks: Systematically Test-Driving Algorithms before Choosing. In: Brazdil, P.B., Jorge, A.M. (eds.) EPIA 2001. LNCS (LNAI), vol. 2258, pp. 88–94. Springer, Heidelberg (2001)
22. Vanschoren, J., Blockeel, H.: A Community-Based Platform for Machine Learning Experimentation. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009. LNCS, vol. 5782, pp. 750–754. Springer, Heidelberg (2009)
23. Vilalta, R., Drissi, Y.: A perspective view and survey of meta-learning. Artif. Intell. Rev. 18(2), 77–95 (2002)

Comparing Logistic Regression, Neural Networks, C5.0 and M5′ Classification Techniques

Amit Thombre

Centre of Excellence, Tech Mahindra, Pune, India
amitmt@techmahindra.com

Abstract. The aim of the paper is to compare the prediction accuracies obtained using logistic regression, neural networks (NN), C5.0 and M5′ classification techniques on 4 freely available data sets. For this a feedforward neural network with a single hidden layer and using back propagation is built using a new algorithm. The results show that the training accuracies obtained using the new algorithm are better than that obtained using N2C2S algorithm. The cross-validation accuracies and the test prediction accuracies obtained by using both the algorithms are not statistically significantly different. Due to this and also since it is easy to understand and implement than N2C2S algorithm, the proposed algorithm should be preferred than the N2C2S algorithm. Along with this 3 different methods of obtaining weights for neural networks are also compared. The classification results show that NN is better than logistic regression over 2 data sets, equivalent in performance over 2 data sets and has low performance than logistic regression in case of 1 data set. It is observed that M5′ is a better classification technique than other techniques over 1 dataset.

Keywords: classification, logistic regression, feedforward neural network, backpropagation, N2C2S algorithm, C5.0, M5′.

1 Introduction

Classification is a data mining (machine learning) technique which divides up data instances such that each is assigned to one of a number of classes. The data instances are assigned to precisely one class and never to more than one class or never to no class at all. Classification problems can be found in business, science, industry, and medicine. Some of the examples include bankruptcy prediction, customer churn prediction, credit scoring, medical diagnosis (like predicting cancer), quality control, handwritten character recognition, speech recognition etc. Some of the widely used classification techniques are decision trees, rule based classification, neural networks (NN), Bayesian networks, logistic regression, k-nearest neighbor classifier, support vector machines etc. In this study, classification is carried out using four fundamentally different approaches, viz., the traditional statistical method based on logistic regression, the computationally powerful technique based on Artificial Neural Networks (ANN), the model tree technique based on M5' and the classical decision tree based technique based on C5.0.

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 132–140, 2012.

Logistic regression (sometimes called the logistic model or logit model) is used for prediction of the probability of occurrence of an event by fitting data to a logit function. The logistic function which is as given below takes z as the input and f (z) is the output which is always between 0 and 1.

$$f(z)= e^z /(e^z +1) =1/(1+e^{-z}) \ldots\ldots\ldots\ldots \quad (1)$$

The variable z comprises of other independent variables and is given as

$$z=\beta_0+\beta_1x_1+\beta_2x_2+\beta_3x_3+\beta_4x_4+\beta_5x_5+\ldots..+\beta_kx_k \quad \ldots \quad \ldots (2)$$

where β0 is the intercept and β1, β2, β3 are the regression coefficients of independent variables x1, x2, x3 respectively. Each of the regression coefficients describes the size of the contributing independent variable. Logistic regression is widely used in medical field [1], [2].

NN have gained popularity over a few years and are being successfully applied across wide problem domains such as finance, medicine, engineering, geology etc. NN can adapt to the data without any explicit specification of functional or distributional form for the underlying model [3]. NN can approximate any function thus making them flexible in modeling real world complex relationships. Neural networks have successfully been applied for classification problems in bankruptcy prediction [4], [5], medical diagnosis [6], [7], handwriting recognition [8] and speech recognition [9]. Artificial neural networks are computing systems made up of large number of simple, highly interconnected processing elements called nodes or artificial neurons. The focus of this work is on the feedforward multilayer networks or multilayer perceptrons (MLPs) with 1 input layer, 1 output layer and 1 hidden layer as one hidden layer is sufficient to map an arbitrary function to any degree of accuracy. The number of neurons in the hidden layer needs to be fixed to arrive at the correct architecture of the network. There are many algorithms which construct the networks with single hidden layer. In the Dynamic node creation (DNC) algorithm [10] the nodes are added to the hidden layer one at a time till a desired accuracy is obtained. In Feedforward Neural network Creation Algorithm (FNNCA) [11] and Constructive Algorithm for Real-Value Examples (CARVE)[12] the hidden units are added to the hidden layer one at a time until a network that completely recognizes all its input patterns is constructed. Using these algorithms can lead to overfitting of the training data and do not generalize well with unknown data. In Neural network Construction with Cross-Validation Samples algorithm (N2C2S) [13] the nodes are added to the hidden layer only if they improve the accuracy of the network on the training and the cross-validation data. This algorithm uses freezing of weights which can lead to increase in number of hidden units [14]. In the proposed algorithm, the weights are not frozen and the nodes are added to the hidden layer only if they improve the accuracy of the network on the cross-validation data only. The cross-validation data accuracy is selected because it is true measure of the performance of the model.

C5.0 is an algorithm to build decision trees using the concept of information entropy. The decision tree is used as a predictive model which maps observations about an item to conclusions about the item's target value. In these tree structures the leaves

represent the class labels and branches represent the rules of features that lead to those class labels.

M5′ builds tree based models in which the trees constructed have multivariate linear models. Thus these trees are analogous to piecewise linear functions. They are applied to classification problems by applying a standard method of transforming a classification problem into a problem of function approximation [15]. M5′ is based on M5 developed by Quinlan [16] but includes techniques to handle enumerated data and missing values effectively [17].

The organization of the paper is as follows. After the introduction in section 1, section 2 presents the algorithm used for building the NN. Section 3 describes the experimental setup. Section 4 presents the results from the experiments and section V gives the discussion and conclusion from the work carried out. The paper concludes with future work in Section 6.

2 The Proposed Algorithm

The steps of the algorithm used to build the neural network are as follows:-

1. Let N_1 denote the network with I input units and O output units and H hidden units.
2. The initialization of the weights and bias is done using Nguyen-Widrow method [18]. The reason for selecting this algorithm is because it chooses values in order to distribute the active region of each neuron in the layer approximately evenly across the layer's input space. Thus the wastage of neurons by this method is less as compared to random initialization [19].
3. The accuracy of this network on the training and validation data set is AT_1 and AV_1.
4. The number of hidden units is then increased by 1 unit and the weights are initialized with the Nguyen-Widrow method. In case of N2C2S the weights of the first H hidden units of N_2 is obtained from the optimal weights of N_1 and the remaining connection weights are set randomly. Thus in the proposed algorithm the weights are not frozen and the whole network is retrained as freezing requires large number of hidden units to achieve the same performance as that obtained without freezing [14].
5. Let the accuracy of this network N_2 be AT_2 and AV_2 on the training and the validation set respectively. If $AV_2 > AV_1$ then N_2 network is better than N_1 else the network is run with increasing the hidden units by 1. Thus the steps 4 and 5 are repeated till we get the network with highest accuracy over validation data set.

3 Experimental Setup

3.1 Neural Networks

a. The neural network was constructed using Matlab.
b. As mentioned above in section 1 the network consists of input layer, output layer and hidden layer. The number of neurons in the input layer corresponds to the

number of independent variables. The number of neurons in the hidden layer is obtained by using the algorithm mentioned in section 2. And the number of neurons in the output layer is corresponding to the category of classes in the dependent data minus one for programming purpose.

c. The back propagation training function is scaled conjugate gradient algorithm.
d. Each run of the network was carried out for 200 epochs with the goal for error set as 0.
e. The Nguyen-Widrow algorithm chooses the weight values in order to distribute the active region of each neuron in the layer approximately evenly across the layer's input space. These values contain a degree of randomness, so they are not the same each time this function is called. That's why 30 runs were carried out for each fold of cross-validation and for each configuration of the network to consider as many as random weight values and the average of 30 runs is taken. The 10 fold cross-validation was run 10 times and then the prediction accuracy was averaged over these 10 runs.

3.2 Logistic Regression

The programming for logistic regression is done using R [20], a free statistical software. 10 times, 10 fold cross-validation was used to carry out the runs.

3.3 C5.0 and M5′

The results using these algorithms are obtained from [15]. The experiments using these algorithms were carried out using 10 runs of ten fold cross-validation.

3.4 Data

The Chapman data set was obtained from [21] and the rest of the data sets were obtained from the UCI Machine Learning Laboratory [22]. The data sets were chosen in such a way that they are publicly available and the results of classification using neural nets, C5.0 and M5′, on these data sets (except one data set) is already available for comparison purpose. Also the data sets are small in size and contained only continuous data and binary data. The missing continuous attribute value was replaced by the average of the non-missing values. The data was normalized before carrying out the experiments on them.

Table 1. Details of the data sets

Data sets	Size	Missing values	Attributes		
			Continuous	Binary	Nominal
Glass(G2)	163	0	9	0	0
Chapman	200	0	6	0	0
Ionosphere	351	0	33	1	0
Voting	435	5.6	0	16	0
Breast Cancer	699	0.3	9	0	0

3.5 Experiments

All the data sets used were split up into 3 sets, viz training, validation and test data sets. Let them be referred as T_R, V and T_E data sets respectively. Initially the neural network is built using the algorithm mentioned in Section 2. The number of neurons in the hidden layer varied from 1 to 20 maximum. These steps are common to all the experiments mentioned below.

1. Experiment A
   a. The weights in the common step mentioned above were obtained corresponding to lowest generalization error over validation data set.
   b. Then the network with weights fixed as obtained from step a was run on the training set comprising of the training set T_R and validation set V and its accuracy was tested on the testing data set T_E. This prediction accuracy is reported in the results.

2. Experiment B

In this experiment, the steps a and b remain the same as experiment A except that the criteria for obtaining weights in step a is corresponding to the highest prediction accuracy over the validation data set.

3. Experiment C

In this experiment there was no external criteria laid down as done in experiments A and B for getting the weights. Thus the weights were not fixed. The weights which were obtained by each run were used as it is for getting the prediction accuracy over the training set.

4 Experimental Results

The prediction accuracies obtained by 2 different methods were compared by using the Welch's t test [23]. Welch's test is an adaptation of Student's t-test with the 2 samples having unequal variances. The null hypothesis that the two means are equal was rejected at the significance value of 0.01. Following are the results from various experiments tried out:-

1. The number of hidden units and the accuracy rates of the neural networks constructed by N2C2S and the proposed algorithm are given in the tables 2 and 3.

Table 2. Results using N2C2S algorithm

	Using N2C2S Algorithm		
Data set	Hidden units	Training Accuracy	Cross-validation Accuracy
Glass 2	7.06+-0.84	89.98+-1.10	81.90+-3.43
Chapman	*[1]	*	*
Ionosphere	4.20+-0.87	99.20+-0.15	92.65+-1.16
Voting	4.42+-0.61	98.55+-0.13	96.64+-0.48
Breast Cancer	3.4+-0.51	97.47+-0.06	96.94+-0.27

[1] * Indicates that the data is not available.

Table 3. Results using proposed algorithm

	Using Mentioned Algorithm		
Data set	Hidden units	Training Accuracy	Cross-validation Accuracy
Glass 2	17	95.15+-0.96	79.9+-2.31
Chapman	1	93.01+-0.67	84.44+-7.28
Ionosphere	2	99.13 +-0.2	91.35+-4.65
Voting	11	99.36+-0.63	96.03+-2.95
Breast Cancer	1	99.01+-0.55	95.40+-1.69

2. The prediction accuracy over test data set obtained using experiments A, B and C is given in the table 4. This prediction accuracy is also compared with that obtained from logistic regression.

Table 4. Prediction accuracies using logistic regression and by neural networks using experiments A, B and C

Data sets	Logistic Regression	Exp A	Exp B	Exp C
Glass 2	69.88+1.09 ◀[2]	77.97+-2.16	78.10+-1.97	76.6+-0.6
Chapman	86.45+-1.01 ▶[3]	80+-1.89	80.611+-1.75	81.33+-0.77
Ionosphere	87.74+-0.93 ◀	90.64+-1.1	90+-0.9	90.38+-0.75
Voting	95.53+-0.41	96.28+-1.58	95.8+-2.41	95.73+-2.65
Breast Cancer	96.55+-0.16	95.83+-0.81	95.79+-0.9	95.7+-1.87

3. From table 4, the summary of results showing the comparison of neural networks with logistic regression is given in the table 5. The wins and losses are decided as per the Welch's test described at the start of this section.

Table 5. Summary of results

NN versus	Win	Ties	Losses
Logistic Regression	2	2	1

4. The test prediction accuracies using the N2C2S algorithm and the proposed algorithm are given in the table 6.

Table 6. Prediction accuracies using N2C2S algorithm and prosed algorithm

Data sets	Using N2C2S	Exp A	Exp B	Exp C
Glass 2	77.91+-2.5	77.97+-2.16	78.10+-1.97	76.6+-0.6
Chapman	*	80+-1.89	80.611+-1.75	81.33+-0.77
Ionosphere	89.52+-2.26	90.64+-1.1	90+-0.9	90.38+-0.75
Voting	96.09+-0.48	96.28+-1.58	95.8+-2.41	95.73+-2.65
Breast Cancer	96.58+-0.24	95.83+-0.81	95.79+-0.9	95.7+-1.87

[2] ◀ indicates that the prediction accuracy is less compared with other methods.
[3] ▶ indicates that the prediction accuracy is high compared with other methods.

5. Reference [15] gives the prediction accuracies using C5.0 and M5′ on the same data sets used in this study. The prediction accuracies of logistic regression and NN are compared with that obtained by using C5.0 and M5 algorithms in table 7. The results of C5.0 and M5′ are the averages and standard deviations from 10 runs of ten fold cross-validation experiments. The prediction accuracies using NN from experiment A are used here.

Table 7. Prediction accuracies using NN, Logistic Regression, C5.0 and M5′

Data sets	Prediction Accuracy using			
	Neural Networks	Logistic Regression	C5.0	M5′
Glass 2	77.97+-2.16	69.88+1.09	78.27+-2.1	81.8+-2.2
Chapman	80+-1.89	86.45+-1.01	*	*
Ionosphere	90.64+-1.1	87.74+-0.93	88.9+-1.6	89.7+-.12
Voting	96.28+-1.58	95.53+-0.41	96.3+-0.6	96.2+-0.3
Breast Cancer	95.83+-0.81	96.55+-0.16	94.5+-0.3	95.3+-0.3

5 Discussion and Conclusions

1. Table 6 shows that the prediction accuracies using N2C2S is same as that obtained using the proposed algorithm. The proposed method of building the network used 30 iterations for each run, thus taking more combinations of weights using Nguyen-Widrow initialization method into consideration to arrive at the results as compared to running the N2C2S algorithm [11] which uses random weight initialization only once. Also the results using N2C2S algorithm are based on a smaller training set than the one used for the experiments carried out. Thus the results using N2C2S algorithm are likely to change if more iterations and a larger training set is considered.
2. The results in table 2 and 3 show that the hidden units obtained using proposed algorithm was less than that obtained using N2C2S over 2 data sets and greater in case of 2 other data sets. Thus it cannot be concluded if the non-freezing of weights has any advantage over freezing of weights on the number of hidden units and some more experiments need to be performed over different data sets to arrive at some conclusion.
3. The prediction accuracy over training data sets using proposed algorithm was statistically better than that obtained using N2C2S algorithm over 3 data sets. From tables 2, 3 and 6, it is observed that the cross-validation accuracies and test accuracies obtained using both the algorithms are not significantly different. Thus the proposed algorithm should be preferred as the changes are easy to understand and implement as compared to the N2C2S algorithm.
4. From table 4 it is observed there is no single experiment which has performance better than other 2 experiments.
5. From tables 5 it is observed that NN gives better performance than logistic regression 2 data sets; gives the same results on 2 data sets and lower than that of logistic regression on 1 data set. But it is observed that NN consumes more

execution time than logistic regression. Also logistic regression technique is a white-box technique which allows the interpretation of the model parameters whereas NN is a black box technique which does not allow the interpretation of the model.

6. NN is better than C5.0 over 1 data set and equivalent in performance on the remaining data sets.
7. Table 7 shows that M5′ is better than NN, logistic regression and C5.0 over 1 dataset. M5′ is better than logistic regression over 2 data sets and equivalent in performance for the remaining datasets. M5′ and NN do not have significantly different accuracies except for 1 data set as mentioned at the starting of this point. M5′ and C5.0 do not have significantly different accuracies on the data sets except for 1as mentioned at the starting of this point.
8. C5.0 is better than logistic regression over 1 data set and equivalent in performance on the remaining data sets.

6 Future Work

From tables 2, 3 and 6, the proposed algorithm for building the neural networks has given cross-validation prediction accuracies and test accuracies which are not significantly different with that obtained by using the N2C2S algorithm. The further work will be to improve this algorithm so that the cross-validation accuracies are better than that obtained using N2C2S algorithm and the prediction accuracies over test data are better than that obtained by using N2C2S algorithm and logistic regression. Also some more experiments need to be performed using N2C2S algorithm and the changes suggested in point 1 of section 5.

Acknowledgement. I am grateful to Tech Mahindra Centre of Excellence group for giving me the support to write this paper.

References

1. Zhang, G.P.: Neural Networks for classification: A Survey. IEEE Transactions on Systems, Man, and Cybernetics- Part C: Applications and Reviews 30(4), 451–462 (2000)
2. Liao, J.G., Chin: Logistic regression for disease classification using microarray data: model selection in a large *p* and small *n* case. Bioinformatics 23, 1945–1951 (2007)
3. Zhang, G.P.: Neural Networks for classification: A Survey. IEEE Transactions on Systems, Man, and Cybernetics- Part C: Applications and Reviews 30(4), 451–462 (2000)
4. Atiya, A.F.: Bankruptcy Prediction for Credit Risk Using Neural Networks: A Survey and New Results. IEEE Transactions on Neural Networks 12(4), 929–935 (2000)
5. Lacher, R.C., Coats, P.K., Sharma, S.C., Fant, L.F.: A neural network for classifying the financial health of a firm. Eur. J. Oper. Res. 85, 53–65 (1995)
6. Baxt, W.G.: Use of an artificial neural network for data analysis in clinical decision-making: The diagnosis of acute coronary occlusion. Neural Computing 2, 480–489 (1990)

7. Mazurowski, M.A., Habas, P.A., Zurada, J.M., Lo, J.Y., Baker, J.A., Tourassi, G.D.: Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. Neural Networks 21, 427–436 (2007)
8. Guyon, I.: Applications of neural networks to character recognition. International Journal of Pattern Recognition and Artificial Intelligence 5, 353–382 (1991)
9. Bourlard, H., Morgan, N.: Continuous speech recognition by connectionist statistical methods. IEEE Transactions on Neural Networks 4, 893–909 (1993)
10. Ash, T.: Dynamic node creation in backpropagation networks. Connection Science 1(4), 365–375 (2002)
11. Setiono, R.: Feedforward Neural Network Construction Using Cross Validation. Neural Computation 13(12), 2865–2877 (2001)
12. Young, S., Downs, T.: CARVE -a constructive algorithm for real-valued examples. IEEE Transaction on Neural Networks 9(6), 1180–1190 (1998)
13. Setiono, R.: A Neural Network Construction Algorithm which Maximizes the Likelihood Function. Connection Science 7(2), 147–166 (1996)
14. Kwok, T.-Y., Yeung, D.-Y.: Experimental analysis of input weight freezing in constructing neural networks. In: IEEE International Conference on Neural Networks, vol. 1, pp. 511–516 (1993)
15. Frank, E., Wang, Y., Inglis, S., Holmes, G., Witten, I.H.: Using Model trees for classification. Machine Learning 32(1), 63–76 (1997)
16. Quinlan, J.R.: Learning with continuous classes. In: Proceedings Australian Joint Conference on Artificial Intelligence, pp. 343–348. World Scientific, Singapore (1992)
17. Wang, Y., Witten, I.H.: Induction of model trees for predicting continuous classes. In: Proceedings of the poster papers of the European Conference on Machine Learning. Faculty of Informatics and Statistics, Prague (1997)
18. Nguyen, D., Widrow, B.: Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In: Proceedings of the International Joint Conference on Neural Networks, vol. 3, pp. 21–26 (1990)
19. Demuth, H., Beale, M., Hagan, M.T.: Neural Network Toolbox 7, User's Guide. The MathWorks, Inc., Natick, MA, Revised for Version 7.0 (Release 2010b) (September 2010), `http://www.mathworks.com`
20. `http://www.r-project.org/`
21. `http://www.stat.cmu.edu/~brian/720/christensen-data/`
22. UCI Machine learning repository data sets, `http://archive.ics.uci.edu/ml/datasets.html`
23. Welch, B.L.: The generalization of "Student's" problem when several different population variances are involved. Biometrika 34(1-2), 28–35 (1947)

Unsupervised Grammar Inference Using the Minimum Description Length Principle

Upendra Sapkota[1], Barrett R. Bryant[2], and Alan Sprague[1]

[1] Department of Computer and Information Sciences,
University of Alabama at Birmingham, Birmingham, AL 35294-1170, USA
upendra@uab.edu,
sprague@cis.uab.edu
[2] Department of Computer Science and Engineering, University of North Texas,
Denton, TX 76203-5017, USA
Barrett.Bryant@unt.edu

Abstract. Context Free Grammars (CFGs) are widely used in programming language descriptions, natural language processing, compilers, and other areas of software engineering where there is a need for describing the syntactic structures of programs. Grammar inference (GI) is the induction of CFGs from sample programs and is a challenging problem. We describe an unsupervised GI approach which uses simplicity as the criterion for directing the inference process and beam search for moving from a complex to a simpler grammar. We use several operators to modify a grammar and use the Minimum Description Length (MDL) Principle to favor simple and compact grammars. The effectiveness of this approach is shown by a case study of a domain specific language. The experimental results show that an accurate grammar can be inferred in a reasonable amount of time.

Keywords: grammar inference, context free grammar, domain specific language, minimum description length, unsupervised learning.

1 Introduction

It is difficult to retrieve information from unsupervised data about which little information is known. For unsupervised data generated by a grammar, induction of the underlying grammar is a challenge. However, machine learning of grammars finds many applications in software engineering, syntactic pattern recognition, computational biology, computational linguistics, speech recognition, natural language acquisition, etc. Recovery of grammars from legacy systems in software engineering is used to automatically generate different software analysis and modification tools. Grammars for this case can be generated semi-automatically from compilers and language references and other artifacts rather than generating from scratch [5]. But in the application of generating grammars for Domain Specific Languages (DSLs) [9] being designed by domain experts

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 141–153, 2012.

who may not be well versed in language design or implementation, no compilers or language references exist and the idea suggested in [5] does not apply. The syntactic structure of the underlying grammar needs to be generated entirely based on the sample sentences of the language, which the domain experts are able to provide.

Grammar inference [3] is a subfield of machine learning. Although it is believed that learning the structure of Natural Languages solely from positive instances is possible, Gold's theorem [2] states that based on positive evidence only, it is impossible to learn any class of languages other than those of finite cardinality. Gold's theorem provides one of the most important theoretical results in the grammar induction field and proves that a learner, if using an "identification in the limit" guessing rule, will be wrong only a finite number on times. Identification in the limit of any of the four classes of languages in the Chomsky hierarchy using only positive samples is impossible, but Chomsky hierarchy languages can be generated with the use of both positive and negative samples which becomes the complete representation of the corresponding language. Final generalization in Gold's theorem would be the automaton that accepts all the positive strings but rejects all the negative strings. But using only the positive samples, it is uncertain and difficult to determine when to stop the automaton process because of lack of negative samples. This suggests the need of some restriction during the inference process. Several grammar induction techniques have been developed for generating a grammar from positive samples only but it is still a challenging research field. Though identification of general purpose programming languages using grammar inference techniques is not feasible due to the large search space required, we have found that small Domain Specific Languages can still be inferred. The use of the Minimum Description Length (MDL) Principle for grammar inference [6,13] has been tested under simple artificial grammars but we show in this paper, along with experimental results, how well the MDL principle works on a simple but real and complete DSL. The paper describes how underlying grammars of a small DSL can be learned from positive samples. In Section 2, we present related work in this area. Section 3 explains the Minimum Description Length Approach. The experimental results and evaluation are given in Section 4. Finally we conclude in Section 5.

2 Related Work

Grammar inference has been mainly successful in inferring regular languages. RPNI, the Regular Positive and Negative Inference algorithm [11], was developed for learning regular languages from the complete representation (positive and negative samples). The genetic approach [1] gives results which are comparable to other approaches for regular languages. Context Free Grammar inference is more difficult than regular grammar inference. Even the use of both negative and positive (characteristics) samples has not resulted in good success. The best results have come from providing extra information to the inference process, for example, the structure of the generated parse trees. Our goal is to find a new algorithm

which entirely learns grammar from given positive samples when neither complete structured sentences nor partially structured sentences are available.

Various works have used clustering as an approach to grammar inference. An iterative biclustering approach [15] has been used for finding the clusters and then grammar. Based upon the given corpus, a table T of the number of appearances of each symbol pair in the corpus is created. This approach generates a basic grammar based on biclusters in the corpus C, and replaces the appearances of the symbol pairs (biclusters) by nonterminal symbols. The table T is updated according to the reduction. The process is repeated until no further rules can be generated.

GenInc [4] is an unsupervised CFG learning algorithm for assisting DSL developers who lack deep knowledge of computer science and programming language. GenInc uses ordered characteristic positive samples and it is based on the PACS (Probably Approximately Correct learning under Simple distributions)[7] learning paradigm and infers a grammar incrementally. GenInc analyzes one training sample at a time, maintains only one CFG in the memory and does not reprocess previously processed samples. It compares the current sample with the current CFG and infers the next grammar. The requirement of ordered characteristic samples limits the use of GenInc for real problems which may not be characteristic. A different ordering of samples might result in a wrong grammar and the difference between two successive samples should be small; only one new feature is allowed, which again restricts the entire grammar inference process.

A memetic algorithm is a population-based evolutionary algorithm enhanced with local search. MAGIc (Memetic Algorithm for Grammatical Inference) [8] infers context free grammars from positive samples. An initial population of the grammars is generated at the beginning using the Sequitur algorithm [10], that detects repetition in a sample and factors it out by forming grammar rules. The grammar generated by Sequitur is not generalized and parses only the samples from which the grammar was generated. After initialization, an evolutionary cycle is performed, where the population of grammars undergoes transformations through local search, mutation, generalization and selection operators, which selects grammars for the next generation based on individual fitness values. After a certain number of generations, only those generated grammars that parse all positive samples are returned as the result. In the local search step it uses the Linux diff command to find the difference between two random samples and this difference is used to change the selected grammar

3 Minimum Description Length Approach

We explore a method for grammar inference using Minimum Description Length (MDL) [14], which incorporates a heuristic that tries to compress the grammar as well as the encoding of the positive sentences by the grammar. Both GRIDS ("GRammar Induction Driven by Simplicity") [6] and e-GRIDS [13] direct the search towards a simple grammar. Most of the prior works that use MDL for grammar inference are either for some artificial grammars [13] or for sub-parts

of natural language grammar like adjective phrase grammar [6]. Application of the MDL approach to infer a grammar for domain specific languages like the DESK language [12] is explored in this paper. Entirely based on positive samples, the initial step is to generate one grammar rule per sample. The initial grammar is only able to recognize the input samples. We need to generalize this grammar. The main problem in generalization is that the inferred grammar can be overgeneralized and may recognize many sentences that are not in the target language leading to a grammar that we are not interested in. Our main goal is to infer a grammar that will be able to recognize all the positive samples but reject negative samples. Because of the great chance of a grammar being overgeneralized, most algorithms require negative samples as a part of the input, to prevent overgeneralization of grammars. But for most programming languages, we only have positive samples and hence, our approach should generate a grammar using positive samples only. The heuristic that is used to prevent overgeneralization is simplicity. This heuristic not only tries to compress the grammar but also the encoding of all the positive samples by the grammar. Initially, a *FLAT* grammar, which has one production rule per sentence, is generated. Once the *FLAT* grammar is generated, the grammar rules are generalized using some operators. Generalization can take a variable number of iterations but convergence is ensured as each iteration chooses the grammar that needs a minimum number of bits to encode both the grammar and all the samples with respect to the grammar. The evaluation function for choosing the best grammar is the sum of Grammar Description Length (GDL) and Derivation Description Length (DDL) and given by:

$$\begin{aligned} \textit{Number of bits} &= GDL + DDL \\ &= |G| + |code(D/G)| \end{aligned}$$

where $|G|$ is the number of bits required to represent the grammar and $|code(D/G)|$ is the number of bits required to represent the data (D) given the grammar $|G|$. This heuristic directs the inference process towards a simple grammar but prevents overly general grammars. Because of the consideration of both the Grammar Description Length (GDL) and Derivation Description Length (DDL), the grammar is neither overly general nor trivial. GDL prevents trivial grammar because it tries to choose a grammar that can be encoded with few bits. At the same time, DDL prevents unnecessarily overgeneralized grammar because it requires a larger number of bits to encode the samples when the grammar is over-general. Hence, in each iteration we improve our grammar and finally the process terminates giving the best grammar such that the minimum number of bits are required to encode both the grammar and samples given the grammar. Actually, DDL measures the derivation power of the grammar and the best way to measure that is to count all the derivations, which is not always possible. So, instead of counting all possible derivations, we encode the samples with respect to the grammar and the number of bits required is considered as the DDL. Three operators are used to modify the

grammar. Before explaining how each operator is used to generalize the grammar, calculation of GDL and DDL is explained with an example.

3.1 Grammar Description Length (GDL)

The encoding of grammar and data should be done in such a way that a hypothetical recipient should be able to decode the bits to the original grammar and data. The grammar is viewed as a piece of code and encoding is done on the code. We separate all the grammar rules into three subsets and these three subsets are encoded and sent to receiver sequentially. The first subset contains all the rules having the start symbol as the head, the second subset contains all the terminal (unary) rules and the third subset contains all the remaining rules.

1. First subset, which we call Start Rules (S_{rules}), contains all the rules having start symbol as head (i.e., left hand side of the rule is the start symbol).
2. Second subset, which we call Terminal Rules (T_{rules}), contains all the rules of the form $X \rightarrow \alpha$ where X is any nonterminal symbol excluding start symbol and α is any terminal symbol.
3. Third subset, which we call NonTerminal Rules (NT_{rules}), contains all other rules. Each rule in this subset will have only nonterminal symbols on the right side. The head of each rule is one of the nonterminal symbols excluding start symbol.

The bits required to encode all rules in the mentioned three subsets gives us the GDL. The separation of the rules into three subsets makes encoding of each subset independent of each other and hence, overall encoding is easier. First, all the rules in S_{rules} are encoded and transmitted. Similarly, encoding and transmission of T_{rules} is done and finally NT_{rules}. But before encoding, rules are considered as strings of symbols and each rule is separated from others by a special symbol called the STOP symbol. The STOP symbol helps to determine when each rule ends. Except the terminal rules whose right hand side (body of the rule) is a single terminal symbol, all other rules have nonterminal symbol on both body and head. The following three rules $A \rightarrow BC$, $C \rightarrow CD$ and $D \rightarrow DE$, which belong to the last subset NT_{rules}, are converted to $ABC\#CCD\#DDE$ so that each rule is separated by a special STOP symbol (#) and the first nonterminal symbol after each # is known to be head of the rule and the remaining nonterminal symbols are the body of the rule. Since, all the rules in the subset S_{rules} have the start symbol as the head and there is only one start symbol for the entire grammar, the head of these rules can be ignored as it is obvious. Three rules $S \rightarrow AB$, $S \rightarrow BC$ and $S \rightarrow SB$, belonging to the S_{rules} subset, are encoded as string of symbols as $AB\#BC\#SB$. Similarly, each rule in the T_{rules} subset has a single terminal symbol on the right side but can have any nonterminal symbol except the start symbol on the left side. Therefore, there is no need of # to separate these terminal rules as it is obvious that after each terminal symbol, a new rule starts. Three rules $A \rightarrow a$, $B \rightarrow b$ and $C \rightarrow c$ from the T_{rules} subset are encoded as $AaBbCc$.

Now, we explain how the exact number of bits to encode the entire grammar is calculated. As explained above, grammar rules are encoded as a string of symbols and these symbols can be either terminal or nonterminal. So, we need to find the number of unique nonterminal symbols and number of unique terminal symbols in the grammar. As all the rules having head as the start symbol are encoded separately, we exclude the start symbol while counting unique nonterminal symbols. Let $count_{UNTs}$ be the number of unique nonterminal symbols in our grammar and the introduction of special STOP symbol (#) makes the total count $count_{UNTs} + 1$. Therefore, the number of bits that are required to encode each nonterminal symbol is[1]:

$$Bits_{NT} = \log(count_{UNTs} + 1)$$

Let $count_{UTs}$ be the number of unique terminal symbols in our grammar; the number of bits that are required to encode each terminal symbol is:

$$Bits_{T} = \log(count_{UTs})$$

After calculating the bits required for encoding each terminal and nonterminal symbol, the total bits for encoding three subsets are calculated. Separation of one subset from another by the STOP symbol # should be considered to find the overall GDL after calculating the bits required for each subset. For encoding the first subset (S_{rules}), the total number of nonterminal symbols in each rule should be found. Let the body of a rule in this subset have $count_{bodyNTs}$ nonterminal symbols and when the STOP symbol (#) is added at the end of each rule, this count becomes $count_{ruleNTs} + 1$. Therefore, the total bits required for encoding all the rules in subset S_{rules} is:

$$Bits_{S_{rules}} = \sum_{\forall\, rule\ in\ S_{rules}} ((count_{bodyNTs} + 1) \times \log(count_{UNTs} + 1)) \quad (1)$$

Each rule in terminal subset T_{rules} has exactly one nonterminal as the head and one terminal as the body. As separation of rules does not require the STOP symbol, each rule can be encoded just by encoding a nonterminal and a terminal symbol. The total bits required for encoding all the rules in this subset is:

$$Bits_{T_{rules}} = \sum_{\forall\, rule\ in\ T_{rules}} (\log(count_{UNTs} + 1) + \log(count_{UTs})) \quad (2)$$

For the third subset, let the body of each rule have a total of $count_{bodyNTs}$ nonterminal symbols and adding the STOP symbol (#) and head symbol increments this count by two and makes it $count_{bodyNTs} + 2$. The total bits required for encoding all the rules in subset NT_{rules} is:

$$Bits_{NT_{rules}} = \sum_{\forall\, rule\ in\ NT_{rules}} ((count_{bodyNTs} + 2) \times \log(count_{UNTs} + 1)) \quad (3)$$

[1] Log represents the logarithm for base 2.

One subset is separated from other using a STOP symbol and the encoding of this STOP symbol requires $\log(count_{UNTs}+1)$. Now the total GDL is given by:

$$GDL = Bits_{S_{rules}} + \log(count_{UNTs}+1) + Bits_{T_{rules}} \quad (4)$$

$$+\log(count_{UNTs}+1) + Bits_{NT_{rules}} \quad (5)$$

Lets consider the grammar in Fig. 1. The start symbol of the grammar is N_0. There are only two nonterminal symbols (N_1 and N_3) and three terminal symbols in the grammar. The first subset S_{rules} contains the first rule, the second subset T_{rules} contains the remaining three rules and the third subset NT_{rules} does not have any rules. According to equation 1, we get

$$Bits_{S_{rules}} = \underbrace{(2+1)\times\log(2+1)}_{N_0 \to N_1\, N_3} = 3\log(3) = 4.75\, bits$$

Using the equation 2,

$$Bits_{T_{rules}} = \underbrace{\log(2+1)+\log(3)}_{N_1 \to print} + \underbrace{\log(2+1)+\log(3)}_{N_3 \to id}$$

$$+ \underbrace{\log(2+1)+\log(3)}_{N_3 \to number} = 9.5\, bits$$

And, $Bits_{NT_{rules}} = 0$

Therefore, using equation 4,

$$\begin{aligned} GDL &= Bits_{S_{rules}} + \log(count_{UNTs}+1) + Bits_{T_{rules}} \\ &\quad +\log(count_{UNTs}+1) + Bits_{NT_{rules}} \\ &= 4.75 + \log(2+1) + 9.5 + \log(2+1) + 0 \\ &= 17.42\, bits \end{aligned}$$

3.2 Derivation Description Length (DDL)

The number of bits required to encode all the samples based on the grammar G is the derivation description length. It should consider the complete derivation of each sample from the grammar G. As shown in Fig. 2, the complete derivation of the sample should be considered. To encode a sample *"print id"*, the reader should be told which one of the start rules is used for parsing the current sample. According to the grammar in Fig.1, as there is only one start rule $N_0 \to N_1\, N_3$, $\log(1)$ bits are required to specify the start rule. Again, other rules that are used for parsing the given sentence should also be specified uniquely. There is only one rule having N_1 as head, so $\log(1)$ bits required for specifying the rule $N_1 \to print$. Similarly, there are two rules having N_3 as the head and one of the two rules used in deriving the sample should be specified uniquely, so this requires $\log(2)$ bits. The total DDL for this single sample is $\log(1) + \log(1) + \log(2) = 1$.

In this way, by adding the contribution to the DDL from each sample, the overall DDL is found. So, search will be performed and the grammar that requires the minimum bits for encoding both the grammar and data is chosen.

$$N_0 \rightarrow N_1\, N_3$$
$$N_1 \rightarrow print$$
$$N_3 \rightarrow id$$
$$N_3 \rightarrow number$$

Fig. 1. Sample grammar to show calculation of GDL

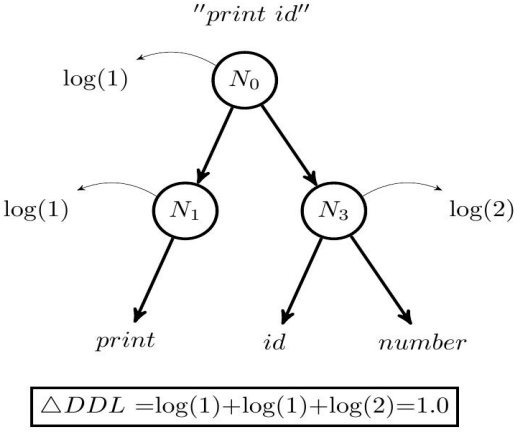


Fig. 2. Effect of sample "*print id*" on DDL

3.3 Operators

Different operators that are used to modify and generalize a grammar are explained in this section.

Merge Operator. The merge operator merges two nonterminal symbols into a new nonterminal symbol. Every occurrence of the merged nonterminals is replaced by the new nonterminal symbol. If nonterminals X and Y are merged into a new common nonterminal Z, then every occurrence of X is replaced by Z, and likewise for Y. For example, merging of X and Y into Z for a rewrite rule $S \rightarrow X\, P\, Y$ produces a rule $S \rightarrow Z\, P\, Z$. The resulting grammar after using this operator always has greater coverage than before. But at the same time, the total number of nonterminal decreases and some of the production rules might be eliminated because merging of two nonterminals might produce identical production rules and hence GDL always decreases. The DDL either decreases or increases depending upon the number of eliminated rules. If no rules are eliminated, DDL always increases because of the increase in grammar coverage. Table 1 shows the effect of the *Merge Operator* on a sample grammar.

Create Operator. The create operator creates a new nonterminal symbol from two consecutive nonterminals, that is, a sequence of nonterminals of length 2 is

Table 1. The effect of Merge Operator, when N_2 and N_3 are merged

Before *"Merge Operator"*	After *"Merge Operator"*
$N_0 \rightarrow N_5\, N_2$	$N_0 \rightarrow N_5\, N_4$
$N_0 \rightarrow N_5\, N_3$	$N_5 \rightarrow print$
$N_5 \rightarrow print$	$N_4 \rightarrow id$
$N_2 \rightarrow id$	$N_4 \rightarrow number$
$N_3 \rightarrow number$	

renamed as a new nonterminal. A new production rule is created and all the occurrences of those two consecutive nonterminals are replaced by the new nonterminal. If the sequence XY is renamed to Z, a rewrite rule $Z \rightarrow X\,Y$ is added and all the occurrences of XY are substituted by Z. The coverage of the grammar remains the same and hence the DDL is unchanged. Because of the introduction of the new rule, the GDL slightly increases, but the substitution of XY by Z slightly decreases the GDL. So, depending upon the substituted sequences of nonterminals, the GDL sometimes increases and sometimes decreases. Table 2 shows the effect of the *Create Operator* on a sample grammar.

Table 2. The effect of Create Operator, when N_2 and N_4 are combined

Before *"Create Operator"*	After *"Create Operator"*
$N_0 \rightarrow N_1\, N_3\, N_2\, N_4$	$N_0 \rightarrow N_1\, N_3\, N_5$
$N_0 \rightarrow N_1\, N_4\, N_2\, N_4$	$N_0 \rightarrow N_1\, N_4\, N_5$
$N_1 \rightarrow print$	$N_5 \rightarrow N_2\, N_4$
$N_3 \rightarrow id$	$N_1 \rightarrow print$
$N_4 \rightarrow number$	$N_3 \rightarrow id$
$N_2 \rightarrow +$	$N_4 \rightarrow number$
	$N_2 \rightarrow +$

Create Optional Operator. Only using merge and create operators resulted in grammars that were less general than we expected. Therefore, we defined an additional operator to create a new production rule by (optionally) attaching a nonterminal at the end of the rule produced by the create operator. If the create operator produces a rule $Z \rightarrow X\,Y$, then the create optional operator appends a new rule $Z \rightarrow X\,Y\,N$ where N is any existing nonterminal. This operator doesn't change the original rule but adds one more production rule and makes the appended nonterminal optional with respect to the original rule. GDL and

DDL both get affected because of the addition of the new rule and the chance of eliminating some duplicate rules. Table 3 shows the effect of *Create Optional Operator* on a sample grammar.

Table 3. The effect of Create Optional Operator, making N_5 optional on rule $N_5 \rightarrow N_2\ N_4$

Before *"Create Optional Operator"*	After *"Create Optional Operator"*
$N_0 \rightarrow N_1\ N_3\ N_5$	$N_0 \rightarrow N_1\ N_3\ N_5$
$N_0 \rightarrow N_1\ N_4\ N_5$	$N_0 \rightarrow N_1\ N_4\ N_5$
$N_5 \rightarrow N_2\ N_4$	$N_5 \rightarrow N_2\ N_4$
$N_1 \rightarrow print$	$N_5 \rightarrow N_2\ N_4\ N_5$
$N_3 \rightarrow id$	$N_1 \rightarrow print$
$N_4 \rightarrow number$	$N_3 \rightarrow id$
$N_2 \rightarrow +$	$N_4 \rightarrow number$
	$N_2 \rightarrow +$

Learning Process. A beam search is used for selecting the best n grammars. The initial grammar has a single production rule for a single sentence (the beam contains only the single grammar initially). During the learning process, each of the operators is repeatedly applied unless the beam search can't find any better grammars than the original. Initially, the merge operator is applied to the initial grammar and all ways of merging nonterminal symbols are explored. For each of the grammars in the beam, corresponding successor grammars are created by repeatedly applying the merge operator and those grammars that require the minimum number of bits to encode are chosen for the next iteration. When the merge operator can't produce any better grammars, the process switches mode and starts applying the create operator. As with the merge operator, for each grammar in the beam, the create operator is applied repeatedly and corresponding successor grammars are generated. These grammars are evaluated and the beam search selects the best n grammars. If this operator can't produce any successor grammars better than the parent grammar, it tries to use the next operator. At last the create optional operator is applied, and selection and termination are same as other operators. This entire process repeats until none of the operators can produce successor grammars better than the parent grammar. The best grammar from the beam is chosen and it is the inferred grammar.

4 Experimental Result and Evaluation

Since the main goal of this work is the inference of grammars for DSLs, we performed an experiment on samples of the DESK calculator language [12]. As the

Table 4. The original and inferred grammar of DESK language

Original DESK Grammar	Inferred DESK Grammar
$N_0 \rightarrow print\, E\, C$ $E \rightarrow E + F$ $E \rightarrow F$ $F \rightarrow id$ $F \rightarrow number$ $C \rightarrow where\, D_s$ $C \rightarrow \epsilon$ $D_s \rightarrow D$ $D_s \rightarrow D_s\, ;\, D$ $D \rightarrow id = number$	$N_0 \rightarrow N_5\, N_4$ $N_0 \rightarrow N_5\, N_8$ $N_5 \rightarrow N_5\, N_2$ $N_5 \rightarrow N_5\, N_3$ $N_2 \rightarrow N_3\, N_3$ $N_3 \rightarrow N_4\, +$ $N_8 \rightarrow N_4\, N_7$ $N_8 \rightarrow N_8\, N_9$ $N_7 \rightarrow where\, N_6$ $N_6 \rightarrow N_{10}\, N_4$ $N_{10} \rightarrow N_4\, =$ $N_9 \rightarrow ;\, N_6$ $N_5 \rightarrow print$ $N_4 \rightarrow id$ $N_4 \rightarrow number$

MDL approach requires a very large number of sample sentences, a large number of samples were generated from the original grammar. All the samples were randomly generated because expansion of ambiguous nonterminals was selected randomly. We tried the MDL approach by changing beam size, total number of positive samples and maximum length of each sample. The left column of Table 4 displays the original grammar and right column displays the inferred grammar when beam size = 2, maximum length of sample = 30 and total number of positive samples = 400.

The inferred grammar has recursive power that can generate samples of any length and can generate every string that the original grammar can. The inferred grammar is almost in Chomsky Normal Form (CNF), which makes it easier to visualize the grammar. Although the inferred grammar is not exactly same as the original grammar, we can conclude that the Minimum Description Length approach gives us a very good grammar that contains both recursive and optional production rules. In our inferred grammar, some unwanted production rules exist because the operators that were used in the learning process were not able to make the grammar completely unambiguous.

After inferring the final grammar from the MDL approach, each nonterminal symbol which only has a single terminal rule and is used only once in the entire grammar is replaced by the corresponding terminal symbol and that terminal rule is deleted from the grammar. This last step does not change the coverage of the grammar but makes it look user friendly and more understandable.

However, the inferred grammar has slightly higher coverage than the original grammar. In the original grammar, the condition after the terminal symbol *where* has a sequence of $id = number$ separated by semicolons, whereas the inferred grammar has a sequence of $N_4 = N_4$ separated by semicolons (and $N_4 \rightarrow id \mid number$). This is caused by the merge operator, which merges two nonterminals into a new nonterminal. As illustrated in Table 1 on a sample grammar, the merge operator merges nonterminals N_2 ($N_2 \rightarrow id$) and N_3 ($N_3 \rightarrow number$) into a new common nonterminal N_4 thereby resulting in production rules $N_4 \rightarrow id$ and $N_4 \rightarrow number$. Replacing every occurrence of both N_2 and N_3 by N_4 changes the sequence $id = number$ into $N_4 = N_4$. Although this operator is a part of the algorithm previously used in [6,13], the correction step to prevent such generalization has not been explored yet. We believe that the entire learning process should be adjusted to address this problem and we plan to explore this direction in future.

5 Conclusion and Future Work

We explained an approach of grammar induction that uses the minimum description length principle. The MDL approach, which tries to bias the learning process towards simple grammars, considers both the encoding length of a grammar as well as the encoding length of samples with respect to the grammar and is able to prevent the grammar both from being trivial and overly general. The entire grammar induction is based on MDL, beam search and three learning operators, and is able to infer a correct grammar for DESK language without the need of negative samples. Most of the previous work [6,13] that studies the use of the MDL principle for grammar inference draws a conclusion from the grammar inference of simple artificial languages, which might be different than the real existing languages in various aspects. We experimentally illustrate how well grammar inference based on the MDL principle works on a simple but real and complete DSL. This study evaluates the application of the MDL principle on grammar inference of a language that was never exploited using the learning algorithm and heuristic which we have used. We reach a conclusion that MDL based grammar inference approach is effective in learning the grammar not only of simple artificial languages but of a real existing DSL.

Exploring more operators for improving the inferred grammar will be our future goal. Comparing to the original grammar, the inferred grammar, although bigger, has almost the same coverage. Introduction of extra nonterminals and absence of rules with three symbols in the body in our inferred grammar increased its size. In the future, we want to explore more about the reduction of size of the inferred grammar. To automatically learn a grammar having exactly the same rules as in the original grammar is extremely difficult. So, rather than just comparing the size and the coverage of our inferred grammar with the original grammar, our future plan will be to compare it against the one inferred by other state-of-art grammar inference approaches. At this point, the MDL approach is tested only for the DESK language. We are interested to look at more real DSL languages and analyze the behavior of this learning algorithm.

Acknowledgments. This work was supported in part by United States National Science Foundation award CCF-0811630.

References

1. Dupont, P.: Regular Grammatical Inference from Positive and Negative Samples by Genetic Search: The GIG Method. In: Carrasco, R.C., Oncina, J. (eds.) ICGI 1994. LNCS, vol. 862, pp. 236–245. Springer, Heidelberg (1994), http://dl.acm.org/citation.cfm?id=645515.658234
2. Gold, E.M.: Language identification in the limit. Information and Control 10(5), 447–474 (1967)
3. de la Higuera, C.: Grammatical Inference: Learning Automata and Grammars. Cambridge University Press, New York (2010)
4. Javed, F., Mernik, M., Bryant, B.R., Sprague, A.: An unsupervised incremental learning algorithm for domain-specific language development. Applied Artificial Intelligence 22(7), 707–729 (2008)
5. Lammel, R., Verhoef, C.: Semi-automatic grammar recovery. Software — Practice & Experience 31(15), 1395–1438 (2001)
6. Langley, P., Stromsten, S.: Learning Context-Free Grammars with a Simplicity Bias. In: Lopez de Mantaras, R., Plaza, E. (eds.) ECML 2000. LNCS (LNAI), vol. 1810, pp. 220–228. Springer, Heidelberg (2000)
7. Li, M., Vitanyi, P.M.: An Introduction to Kolmogorov Complexity and Its Applications, 3rd edn. Springer Publishing Company, Incorporated (2008)
8. Mernik, M., Hrncic, D., Bryant, B., Sprague, A., Gray, J., Liu, Q., Javed, F.: Grammar inference algorithms and applications in software engineering. In: Proceedings of ICAT 2009, the XXII International Symposium on Information, Communication and Automation Technologies, pp. 1–7 (October 2009)
9. Mernik, M., Heering, J., Sloane, A.M.: When and how to develop domain-specific languages. ACM Comput. Surv. 37(4), 316–344 (2005), http://doi.acm.org/10.1145/1118890.1118892
10. Nevill-Manning, C.G., Witten, I.H.: Identifying hierarchical structure in sequences: A linear-time algorithm. Journal of Artificial Intelligence Research 7, 67–82 (1997)
11. Oncina, J., Garcia, P.: Inferring regular languages in polynomial update time. In: Pattern Recognition and Image Analysis, pp. 49–61 (1992)
12. Paakki, J.: Attribute grammar paradigms a high-level methodology in language implementation. ACM Comput. Surv. 27, 196–255 (1995), http://doi.acm.org/10.1145/210376.197409
13. Petasis, G., Paliouras, G., Karkaletsis, V., Halatsis, C., Spyropoulos, C.D.: E-grids: Computationally efficient grammatical inference from positive examples. Grammars 7 (2004)
14. Rissanen, J.: Stochastic Complexity in Statistical Inquiry Theory. World Scientific Publishing Co., Inc., River Edge (1989)
15. Tu, K., Honavar, V.: Unsupervised Learning of Probabilistic Context-Free Grammar using Iterative Biclustering. In: Clark, A., Coste, F., Miclet, L. (eds.) ICGI 2008. LNCS (LNAI), vol. 5278, pp. 224–237. Springer, Heidelberg (2008)

How Many Trees in a Random Forest?

Thais Mayumi Oshiro, Pedro Santoro Perez, and José Augusto Baranauskas

Department of Computer Science and Mathematics
Faculty of Philosophy, Sciences and Languages at Ribeirao Preto
University of Sao Paulo
{thaismayumi,pedrosperez,augusto}@usp.br

Abstract. Random Forest is a computationally efficient technique that can operate quickly over large datasets. It has been used in many recent research projects and real-world applications in diverse domains. However, the associated literature provides almost no directions about how many trees should be used to compose a Random Forest. The research reported here analyzes whether there is an optimal number of trees within a Random Forest, i.e., a threshold from which increasing the number of trees would bring no significant performance gain, and would only increase the computational cost. Our main conclusions are: as the number of trees grows, it does not always mean the performance of the forest is significantly better than previous forests (fewer trees), and doubling the number of trees is worthless. It is also possible to state there is a threshold beyond which there is no significant gain, unless a huge computational environment is available. In addition, it was found an experimental relationship for the AUC gain when doubling the number of trees in any forest. Furthermore, as the number of trees grows, the full set of attributes tend to be used within a Random Forest, which may not be interesting in the biomedical domain. Additionally, datasets' density-based metrics proposed here probably capture some aspects of the VC dimension on decision trees and low-density datasets may require large capacity machines whilst the opposite also seems to be true.

Keywords: Random Forest, VC Dimension, Number of Trees.

1 Introduction

A great interest in the machine learning research concerns ensemble learning — methods that generate many classifiers and combine their results. It is largely accepted that the performance of a set of many weak classifiers is usually better than a single classifier given the same quantity of train information [28]. Ensemble methods widely known are boosting [12], bagging [8], and more recently Random Forests [7,24].

The boosting method creates different base learners by sequentially reweighting the instances in the training set. At the beginning, all instances are initialized with equal weights. Each instance misclassified by the previous base learner will get a larger weight in the next round, in order to try to classify it correctly. The

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 154–168, 2012.

error is computed, the weight of the correctly classified instances is lowered, and the weight of the incorrectly classified instances is increased. The vote of each individual learner is weighted proportionally to its performance [31].

In the bagging method (bootstrap aggregation), different training subsets are randomly drawn with replacement from the entire training set. Each training subset is fed as input to base learners. All extracted learners are combined using a majority vote. While bagging can generate classifiers in parallel, boosting generates them sequentially.

Random Forest is another ensemble method, which constructs many decision trees that will be used to classify a new instance by the majority vote. Each decision tree node uses a subset of attributes randomly selected from the whole original set of attributes. Additionally, each tree uses a different bootstrap sample data in the same manner as bagging.

Normally, bagging is almost always more accurate than a single classifier, but it is sometimes much less accurate than boosting. On the other hand, boosting can create ensembles that are less accurate than a single classifier. In some situations, boosting can overfit noisy datasets, thus decreasing its performance. Random Forests, on the other hand, are more robust than boosting with respect to noise; faster than bagging and boosting; their performance is as good as boosting and sometimes better, and they do not overfit [7].

Nowadays, Random Forest is a method of ensemble learning widely used in the literature and applied fields. But the associate literature provides few or no directions about how many trees should be used to compose a Random Forest. In general, the user sets the number of trees in a *trial and error* basis. Sometimes when s/he increases the number of trees, in fact, only more computational power is spent, for almost no performance gain. In this study, we have analyzed the performance of Random Forests as the number of trees grows (from 2 to 4096 trees, and doubling the number of trees at every iteration), aiming to seek out for a number (or a range of numbers) of trees from which there is no more significant performance gain, unless huge computational resources are available for large datasets. As a complementary contribution, we have also analyzed the number (percentage) of attributes appearing within Random Forests of growing sizes.

The remaining of this paper is organized as follows. Section 2 describes some related work. Section 3 describes what Random Tree and Random Forest are and how they work. Section 4 provides some density-based metrics used to group datasets described in Section 5. Section 6 describes the methodology used, and results of the experiments are shown in Section 7. Section 8 presents some conclusions from this work.

2 Related Work

Since Random Forests are efficient, multi-class, and able to handle large attribute space, they have been widely used in several domains such as real-time face recognition [29], bioinformatics [16], and there are also some recent research

in medical domain, for instance [18,6,21,19] as well as medical image segmentation [33,22,34,15,32].

A tracking algorithm using adaptive random forests for real-time face tracking is proposed by [29], and the approach was equally applicable to tracking any moving object. One of the first illustrations of successfully analyzing genome-wide association (GWA) data with Random Forests is presented in [16]. Random Forests, support vector machines, and artificial neural network models are developed in [18] to diagnose acute appendicitis. Random Forests are used in [6] to detect curvilinear structure in mammograms, and to decide whether it is normal or abnormal. In [21] it is introduced an efficient keyword based medical image retrieval method using image classification with Random Forests. A novel algorithm for the efficient classification of X-ray images to enhance the accuracy and performance using Random Forests with Local Binary Patterns is presented in [19]. An enhancement of the Random Forests to segment 3D objects in different 3D medical imaging modalities is proposed in [33]. Random Forests are evaluated on the problem of automatic myocardial tissue delineation in real-time 3D echocardiography in [22]. In [34] a new algorithm is presented for the automatic segmentation and classification of brain tissue from 3D MR scans. In [15] a new algorithm is presented for the automatic segmentation of Multiple Sclerosis (MS) lesions in 3D MR images. An automatic 3D Random Forests method which is applied to segment the fetal femur in 3D ultrasound and a weighted voting mechanism is proposed to generate the probabilistic class label is developed in [32].

There is one similar work to the one presented here. In [20] is proposed a simple procedure that *a priori* determine the minimum number of classifiers. They applied the procedure to four multiple classifiers systems, among them Random Forests. They used 5 large datasets, and produced forests with a maximum of 200 trees. They concluded that it was possible to limit the number of trees, and this minimum number could vary from one classifier combination method to another. In this study we have evaluated 29 datasets in forests with up to 4096 trees. In addition, we have also evaluated the percentage of attributes used in each forest.

3 Random Trees and Random Forests

Assume a training set T with a attributes, n instances, and define T_k a bootstrap training set sampled from T with replacement, and containing m random attributes ($m \leq a$) with n instances.

A Random Tree is a tree drawn at random from a set of possible trees, with m random attributes at each node. The term "at random" means that each tree has an equal chance of being sampled. Random Trees can be efficiently generated, and the combination of large sets of Random Trees generally lead to accurate models [35,10].

A Random Forest is defined formally as follows [7]: it is a classifier consisting of a collection of tree structured classifiers $\{h_k(\mathbf{x}, T_k)\}$, $k = 1, 2, \ldots, L$, where T_k are independent identically distributed random samples, and each tree casts a unit vote for the most popular class at input $\mathbf{x}$.

As already mentioned, Random Forests employ the same method bagging does to produce random samples of training sets (bootstrap samples) for each Random Tree. Each new training set is built, with replacement, from the original training set. Thus, the tree is built using the new subset, and a random attribute selection. The best split on the random attributes selected is used to split the node. The trees grown are not pruned.

The use of bagging method is justified by two reasons [7]: the use of bagging seems to enhance performance when random attributes are used; and bagging can be used to give ongoing estimates of the generalization error of the combined ensemble of trees, as well as estimates for the strength and correlation. Theses estimates are performed out-of-bag. In a Random Forest, the out-of-bag method works as follows: given a specific training set T, generate bootstrap training sets T_k, construct classifiers $\{h_k(\mathbf{x}, T_k)\}$ and let them vote to create the bagged classifier. For each $(\mathbf{x}, y)$ in the training set, aggregate the votes only over those classifiers for which T_k does not contain $(\mathbf{x}, y)$. This is the out-of-bag classifier. Then the out-of-bag estimate for the generalization error is the error rate of the out-of-bag classifier on the training set.

The error of a forest depends on the strength of the individual trees in the forest, and the correlation between any two trees in the forest. The strength can be interpreted as a measure of performance for each tree. Increasing the correlation increases the forest error rate, and increasing the strength of the individual trees decreases the forest error rate inasmuch as a tree with a low error rate is a strong classifier. Reducing the number of random attributes selected reduces both the correlation and the strength [23].

4 Density-Based Metrics for Datasets

It is well known from the computational learning theory that, given a hypotheses space (in this case, defined by the Random Forest classifier), it is possible to determine the training set complexity (size) for a learner to converge (with high probability) to a successful hypothesis [25, Chap. 7]. This requires knowing the hypotheses space size (i.e., its cardinality) or its capacity provided by the VC dimension [30]. In practice, finding the hypotheses space size or capacity is hard, and only recently an approach has defined the VC dimension for binary decision trees, at least partially, since it was defined in terms of left and right subtrees [4], whereas the gold standard should be defined in terms of the instances space.

On the other hand, datasets (instances space) metrics are much less discussed in the literature. Our concern is, once the hypotheses space is fixed (but its size or its VC dimension are both unknown or infinite), which training sets *seem* to have enough content so that learning could be successful. In a related work we have proposed some class balance metrics [27]. Since in this study we have used

datasets with very different numbers of classes, instances and attributes, they cannot be grouped in some (intuitive) sense using these three dimensions. For this purpose, we suggest here three different metrics, shown in (1), (2), and (3), where each dataset has c classes, a attributes, and n instances.

These metrics have been designed using the following ideas. For a physical object, the density D is its mass divided by its volume. For a dataset, we have considered its mass as the number of instances; its volume given by its attributes. Here we have used the concept the volume of an object (dataset) is understood as its capacity, i.e., the amount of fluid (attributes) that the object could hold, rather than the amount of space the object itself displaces. Under these considerations, we have $D \triangleq \frac{n}{a}$. Since, in general, these numbers vary considerably, a better way to looking at them was using both numbers in the natural logarithmic scale, $D \triangleq \frac{\ln n}{\ln a}$ which lead us to (1). In the next metric we have considered the number of instances (mass) is rarefied by the number of classes, therefore providing (2), and the last one embraces empty datasets (no instances) and datasets without the class label (unsupervised learning).

$$D_1 \triangleq \log_a n \tag{1}$$

$$D_2 \triangleq \log_a \frac{n}{c} \tag{2}$$

$$D_3 \triangleq \log_a \frac{n+1}{c+1} \tag{3}$$

Considering the common assumption in machine learning that $c \leq n$ (in general, $c \ll n$), it is obvious that, for every metric D_i, $D_i \geq 0$, $i = 1, 2, 3$. We considered that if $D_i < 1$, the density is low, and *perhaps* learning from this dataset should be difficult, under the computational point of view. Otherwise, $D_i \geq 1$, the density is high, and learning *may be* easier.

According to [4] the VC dimension of a binary tree is $\mathrm{VC} = 0.7(1 + \mathrm{VC}_l + \mathrm{VC}_r - \log a) + 1.2 \log M$, where VC_l and VC_r represent the VC dimension of its left and right subtrees and M is the number of nodes in the tree. Considering this, our density-based metrics may capture important information about the VC dimension: (i) the number a of attributes is directly expressed in this equation; (ii) since having more classes implies the tree must have more leaves, the number c of classes is related to the number of leaves, and more leaves implies larger M, therefore c is related to M, and probably VC_l and VC_r; (iii) the number n of instances does not appear directly in this expression but it is surely related to VC_l, VC_r, a and/or M, once the VC dimension of a hypotheses space is defined over the instances space [25, Section 7.4.2].

Intuitively, decision trees are able to represent the family of boolean functions and, in this case, the number n of required training instances for a boolean attributes is $n = 2^a$, and therefore $a = \log_2 n$; in other words, n is related to a as well as M, since more nodes are necessary for larger a values. For these problems expressed by boolean functions with $a \geq 2$ attributes and $n = 2^a$ instances,

Table 1. Density-based metrics for binary class problems ($c = 2$) expressed by boolean functions with a attributes and $n = 2^a$ instances

a	n	D_1	D_2	D_3
2	4	2.00	1.00	**0.74**
3	8	1.89	1.26	1.00
4	16	2.00	1.50	1.25
5	32	2.15	1.72	1.49

$D_i \geq 1$ (except $D_3 = 0.74$ for $a = 2$), according to Table 1. Nevertheless, the proposed metrics are able to capture the fact binary class problems have high-density, indicating there is, probably, enough content so learning can take place.

5 Datasets

The experiments reported here used 29 datasets, all representing real medical data, and none of which had missing values for the class attribute. The biomedical domain is of particular interest since it allows one to evaluate Random Forests under real and difficult situations often faced by human experts.

Table 2 shows a summary of the datasets, and the corresponding density metrics defined in the previous section. Datasets are ordered according to metric D_2, obtaining 8 low-density and 21 high-density datasets. In the remaining of this section, a brief description of each dataset is provided.

Breast Cancer, Lung Cancer, CNS (Central Nervous System Tumour Outcome), *Lymphoma, GCM* (Global Cancer Map), *Ovarian 61902, Leukemia, Leukemia nom., WBC* (Wisconsin Breast Cancer), *WDBC* (Wisconsin Diagnostic Breast Cancer), *Lymphography* and *H. Survival* (*H.* stands for *Haberman's* are all related to cancer and their attributes consist of clinical, laboratory and gene expression data. *Leukemia* and *Leukemia nom.* represent the same data, but the second one had its attributes discretized [26]. *C. Arrhythmia* (*C.* stands for *Cardiac*), *Heart Statlog, HD Cleveland, HD Hungarian* and *HD Switz.* (*Switz.* stands for *Switzerland*) are related to heart diseases and their attributes represent clinical and laboratory data. *Allhyper, Allhypo, ANN Thyroid, Hypothyroid, Sick* and *Thyroid 0387* are a series of datasets related to thyroid conditions. *Hepatitis* and *Liver Disorders* are related to liver diseases, whereas *C. Method* (*C.* stands for *Contraceptive*), *Dermatology, Pima Diabetes* (Pima Indians Diabetes) and *P. Patient* (*P.* stands for *Postoperative*) are other datasets related to human conditions. *Splice Junction* is related to the task of predicting boundaries between exons and introns. Datasets were obtained from the UCI Repository [11], except *CNS, Lymphoma, GCM* and *ECML* were obtained from [2]; *Ovarian 61902* was obtained from [3]; *Leukemia* and *Leukemia nom.* were obtained from [1].

Table 2. Summary of the datasets used in the experiments, where n indicates the number of instances; c represents the number of classes; a, $a_{\#}$ and a_a indicates the total number of attributes, the number of numerical and the number of nominal attributes, respectively; MISS represents the percentage of attributes with missing values, not considering the class attribute; the last 3 columns are the density metrics D_1, D_2, D_3 of each dataset, respectively. Datasets are in ascending order by D_2.

Dataset	n	c	$a(a_{\#},a_a)$	MISS	D_1	D_2	D_3
GCM	190	14	16063 (16063, 0)	0.00%	0.54	0.27	0.26
Lymphoma	96	9	4026 (4026, 0)	5.09%	0.55	0.28	0.27
CNS	60	2	7129 (7129, 0)	0.00%	0.46	0.38	0.34
Leukemia	72	2	7129 (7129, 0)	0.00%	0.48	0.40	0.36
Leukemia nom.	72	2	7129 (7129, 0)	0.00%	0.48	0.40	0.36
Ovarian 61902	253	2	15154 (15154, 0)	0.00%	0.57	0.50	0.46
Lung Cancer	32	3	56 (0, 56)	0.28%	0.86	0.59	0.52
C. Arrhythmia	452	16	279 (206, 73)	0.32%	1.08	0.59	0.58
Dermatology	366	6	34 (1, 33)	0.06%	1.67	1.17	1.12
HD Switz.	123	5	13 (6, 7)	17.07%	1.88	1.25	1.18
Lymphography	148	4	18 (3, 15)	0.00%	1.73	1.25	1.17
Hepatitis	155	2	19 (6, 13)	5.67%	1.71	1.48	1.34
HD Hungarian	294	5	13 (6, 7)	20.46%	2.21	1.59	1.52
HD Cleveland	303	5	13 (6, 7)	0.18%	2.22	1.60	1.53
P. Patient	90	3	8 (0, 8)	0.42%	2.16	1.63	1.50
WDBC	569	2	30 (30, 0)	0.00%	1.86	1.66	1.54
Splice Junction	3190	3	60 (0, 60)	0.00%	1.97	1.70	1.63
Heart Statlog	270	2	13 (13, 0)	0.00%	2.18	1.91	1.75
Allhyper	3772	5	29 (7, 22)	5.54%	2.44	1.97	1.91
Allhypo	3772	4	29 (7, 22)	5.54%	2.44	2.03	1.97
Sick	3772	2	29 (7, 22)	5.54%	2.44	2.24	2.12
Breast Cancer	286	2	9 (0, 9)	0.35%	2.57	2.26	2.07
Hypothyroid	3163	2	25 (7, 18)	6.74%	2.50	2.29	2.16
ANN Thyroid	7200	3	21 (6, 15)	0.00%	2.92	2.56	2.46
WBC	699	2	9 (9, 0)	0.25%	2.98	2.66	2.48
C. Method	1473	3	9 (2, 7)	0.00%	3.32	2.82	2.69
Pima Diabetes	768	2	8 (8, 0)	0.00%	3.19	2.86	2.67
Liver Disorders	345	2	6 (6, 0)	0.00%	3.26	2.87	2.65
H. Survival	306	2	3 (2, 1)	0.00%	5.21	4.58	4.21

6 Experimental Methodology

Using the open source machine learning Weka [17], experiments were conducted building Random Forests with varying number of trees in exponential rates using base two, i.e., $L = 2^j$, $j = 1, 2, \ldots, 12$. Two measures to analyze the results were chosen: the weighted average area under the ROC curve (AUC), and the percentage of attributes used in each Random Forest. To assess performance, the experiments used ten repetitions of 10-fold cross-validation. The average of all repetitions for a given forest on a certain dataset was taken as the value of performance (AUC and percentage) for the pair.

In order to analyze if the results were significantly different, we applied the Friedman test [13], considering a significance level of 5%. If the Friedman test rejects the null hypothesis, a *post-hoc* test is necessary to check in which classifier pairs the differences actually are significant [9]. The *post-hoc* test used was the Benjamini-Hochberg [5], and we performed an all versus all comparison, making

all possible comparisons among the twelve forests. The tests were performed using the R software for statistical computing (http://www.r-project.org/).

7 Results and Discussion

The AUC values obtained for each dataset, and each number of trees within a Random Forest are presented in Table 3. We also provide in this table mean and median figures as well as the average rank obtained in the Friedman test. Mean, median and average rank are presented for the following groups: all datasets; only the 8 low-density datasets; and only the 21 high-density ones.

As can be seen, in all groups (all/8 low-density/21 high-density) the forest with 4096 trees has the smallest (best) rank of all. Besides, in the 21 high-density group, we can observe the forests with 2048 and 4096 trees have the same rank. Analyzing the group using all datasets and the 8 low-density datasets, we can notice that the forest with 512 trees has a better rank than the forest with 1024 trees, contrary to what would be expected. Another interesting result concerns mean and median values of high-density datasets for each one of the first three iterations, $L = 2, 4, 8$, are larger than low-density ones; the contrary is true for $L = 16, \ldots, 4096$. This may suggest low-density datasets, in fact, require more expressiveness power (larger forests) than high-density ones. This expressiveness power, of course, can be expressed as the Random Forests (hypotheses) space size or its VC dimension, as explained in Section 4.

In order to get a better understanding, AUC values are also presented in boxplots in Figures 1, 2 and 3 considering all datasets, only the 8 low-density datasets and only the 21 high-density datasets, respectively. As can be seen, in Figures 1 and 2, both mean and median increase as the number of trees grows, but from 64 trees and beyond these figures do not present major changes. In Figure 3, mean and median do not present major changes from 32 trees and 16 trees, respectively.

With these results we can notice an asymptotical behavior, where increases in the AUC values are harder to obtain, even doubling the number of trees within a forest. One way to comprehend this asymptotical behavior is computing the AUC difference from one iteration to the next (for instance, from 2 to 4, 4 to 8, etc.). These results are presented in Figures 4, 5 and 6 for all, 8 low-density and 21 high-density datasets, respectively. For this analysis, we have excluded AUC differences from datasets reaching AUC value equal to 99.99% before 4096 trees (boldface figures in Table 3). Analyzing them, we can notice that using all datasets and the 8 low-density datasets AUC differences (mean and median) between 32 and 64 trees in the forest are below 1%. Considering the 21 high-density datasets, these differences are below 1% between 16 and 32 trees in the forest, and below 0.3% between 32 and 64 trees.

Analyzing Figures 4, 5 and 6, we have adjusted mean and median values by least squares fit to the curve $g = aL^b$, where g represents the percentage AUC difference (gain), and L is the number of trees within the forest. We have obtained

Table 3. AUC values, mean, median and average rank obtained in the experiments. Boldface figures represent values excluded from the AUC difference analysis.

Datasets	Number of Trees											
	2	4	8	16	32	64	128	256	512	1024	2048	4096
GCM	0.72	0.77	0.83	0.87	0.89	0.91	0.91	0.92	0.92	0.92	0.93	0.93
Lymphoma	0.85	0.92	0.96	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99
CNS	0.50	0.52	0.56	0.58	0.59	0.59	0.59	0.58	0.60	0.60	0.60	0.60
Leukemia	0.76	0.85	0.93	0.97	0.98	0.98	0.99	0.99	0.99	0.99	0.99	1.00
Leukemia nom.	0.72	0.81	0.91	0.96	0.99	1.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Ovarian 61902	0.90	0.96	0.98	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
Lung Cancer	0.58	0.64	0.66	0.65	0.65	0.66	0.66	0.68	0.69	0.68	0.68	0.69
C. Arrhythmia	0.71	0.77	0.82	0.85	0.87	0.88	0.89	0.89	0.89	0.89	0.89	0.89
Dermatology	0.97	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
HD Switz.	0.55	0.55	0.58	0.58	0.60	0.61	0.60	0.60	0.60	0.61	0.61	0.61
Lymphography	0.82	0.87	0.90	0.92	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
Hepatitis	0.76	0.80	0.83	0.84	0.85	0.85	0.85	0.85	0.86	0.85	0.86	0.86
HD Hungarian	0.80	0.84	0.86	0.87	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88
HD Cleveland	0.80	0.84	0.87	0.88	0.89	0.89	0.90	0.89	0.89	0.89	0.90	0.90
P. Patient	0.45	0.45	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
WDBC	0.96	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Splice Junction	0.87	0.93	0.97	0.99	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00
Heart Statlog	0.80	0.84	0.87	0.89	0.89	0.89	0.90	0.90	0.90	0.90	0.90	0.90
Allhyper	0.89	0.95	0.98	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Allhypo	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Sick	0.92	0.97	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Breast Cancer	0.60	0.63	0.64	0.65	0.65	0.66	0.66	0.67	0.66	0.66	0.66	0.66
Hypothyroid	0.95	0.97	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
ANN Thyroid	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
WBC	0.97	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
C. Method	0.62	0.64	0.66	0.66	0.67	0.67	0.67	0.68	0.68	0.68	0.68	0.68
Pima Diabetes	0.72	0.76	0.79	0.81	0.81	0.82	0.82	0.82	0.82	0.82	0.83	0.83
Liver Disorders	0.66	0.70	0.72	0.74	0.75	0.76	0.76	0.77	0.77	0.77	0.77	0.77
H. Survival	0.58	0.60	0.61	0.62	0.63	0.63	0.64	0.64	0.64	0.64	0.64	0.64
All												
Mean	0.77	0.81	0.84	0.85	0.86	0.86	0.86	0.87	0.87	0.87	0.87	0.87
Median	0.80	0.84	0.87	0.89	0.89	0.91	0.91	0.92	0.92	0.92	0.93	0.93
Average rank	11.83	10.55	8.79	8.05	6.88	5.81	5.12	4.62	4.31	4.39	3.91	3.72
8 low-density												
Mean	0.72	0.78	0.83	0.85	0.87	0.88	0.88	0.88	0.88	0.88	0.89	0.89
Median	0.72	0.79	0.87	0.91	0.93	0.94	0.95	0.96	0.96	0.96	0.96	0.96
Average rank	12.00	11.00	9.62	8.81	7.94	6.25	4.81	4.44	3.37	3.69	3.37	2.69
21 high-density												
Mean	0.79	0.82	0.84	0.85	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86
Median	0.80	0.84	0.87	0.89	0.89	0.89	0.90	0.90	0.90	0.90	0.90	0.90
Average rank	11.76	10.38	8.47	7.76	6.47	5.64	5.24	4.69	4.66	4.66	4.12	4.12

(for all datasets) using the median AUC difference $a = 6.42$ and $b = -0.83$ with correlation coefficient $R^2 = 0.99$, and using the mean AUC difference $a = 6.06$ and $b = -0.65$ with correlation coefficient $R^2 = 0.98$. For practical purposes, it is possible to approximate to $g \backsimeq \frac{7}{L}\%$ with correlation coefficient $R^2 = 0.99$, which indicates that this is a good fit as well. For instance, having $L = 8$ trees with AUC equals 0.90 its possible to estimate the AUC for 16 trees (doubling L), therefore $g \backsimeq \frac{7}{8}\%$ and the expected AUC value for 16 trees is $0.90 \times (1 + \frac{7/8}{100}) \backsimeq 0.91$. Of course, this formula may be used with any positive number of trees, for example, having an forest of 100 trees, the expected gain in AUC for a forest with 200 trees is 0.07%.

In Table 4 are presented the results of the *post-hoc* test after the Friedman's test, and the rejection of the null hypothesis. It shows the results using all datasets, the 8 low-density datasets, and the 21 high-density datasets. In this table $\triangle$ ($\blacktriangle$) indicates the Random Forest at the specified row is better (significantly) than the Random Forest at the specified column; $\triangledown$ ($\blacktriangledown$) the Random Forest at the specified row is worse (significantly) than the Random Forest at the specified column; $\circ$ indicates no difference whatsoever.

Some important observations can be made from Table 4. First, we can observe that there is no significant difference between a given number of trees (2^j) and its double (2^{j+1}), in all cases. When there is a significant difference, it only appears when we compare the number of trees (2^j) with at least four times this number (2^{j+2}). Second, from $64 = 2^6$ a significant difference was found only at $4096 = 2^{12}$, only when the Random Forest grew sixty four times. Third, it can be seen that from $128 = 2^7$ trees, there is no more significant difference between the forests until 4096 trees.

In order to analyze the percentage of attributes used, boxplots of these experiments are presented in Figures 7, 8 and 9 for all datasets, the 8 low-density datasets, and the 21 high-density datasets, respectively. Considering Figure 7, the mean and median values from 128 trees corresponds to 80.91% and 99.64% of the attributes, respectively. When we analyze the 8 low-density datasets in Figure 8, it is possible to notice that even with 4096 trees in the forest, not all attributes were used. However, as can be seen, this curve has a different shape (sigmoidal) than those in Figures 7 and 9 (exponential). Also, the sigmoidal seems to grow up to its maximum at 100%.

Even Random Forests do not overtrain, this appear to be a unwanted side effect of them, for instance, datasets of gene expression have thousands genes, and in that case a large forest will use all the genes, even if not all are important to learn the biological/biomedical concept. In [26], trees have only 2 genes among 7129 genes expression values; and in [14] the aim of their work was to build classifiers composed by rules with few conditions, and when they use the same dataset with 7129 genes they only use 2 genes in their subgroup discovery strategy. Considering the 21 high-density datasets in Figure 9, from 8 trees the mean and median already corresponds to 96.18% and 100% of attributes, respectively.

Table 4. Friedman's test results for AUC values using all datasets/8 low-density datasets/21 high-density datasets

Number of Trees	2	4	8	16	32	64	128	256	512	1024	2048	4096
2	○	▽ / ▽ / ▽	▼ / ▽ / ▼ ▼	/ ▽ / ▼ ▼	/ ▽ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼
4		○	▽ / ▽ / ▽	▼ / ▽ / ▼ ▼	/ ▽ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼
8			○	▽ / ▽ / ▽ ▽	/ ▽ / ▽ ▼	/ ▽ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼
16				○	▽ / ▽ / ▽	▼ / ▽ / ▽	▼ / ▽ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼ ▼	/ ▼ / ▼
32					○	▽ / ▽ / ▽	▽ / ▽ / ▼	/ ▽ / ▽ ▼	/ ▼ / ▽ ▼	/ ▼ / ▽ ▼	/ ▼ / ▽	▼ / ▼ / ▽
64						○	▽ / ▽ / ▽	▽ / ▽ / ▽	▽ / ▽ / ▽	▽ / ▽ / ▽	▼ / ▽ / ▽	▽ / ▽ / ▽
128							○	▽ / ▽ / ▽	▽ / ▽ / ▽	▽ / ▽ / ▽	▽ / ▽ / ▽	▽ / ▽ / ▽
256								○	▽ / ▽ / ▽	▽ / ▽ / ▽	▽ / ▽ / ▽	▽ / ▽ / ▽
512									○	△ / △ / ○	▽ / ○ / ▽	▽ / ▽ / ▽
1024										○	▽ / ▽ / ▽	▽ / ▽ / ▽
2048											○	▽ / ▽ / ○
4096												○

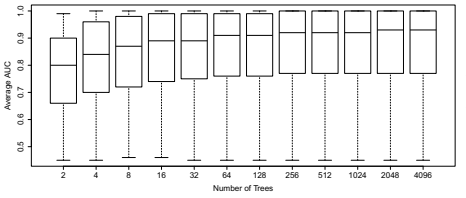


Fig. 1. AUC in all datasets

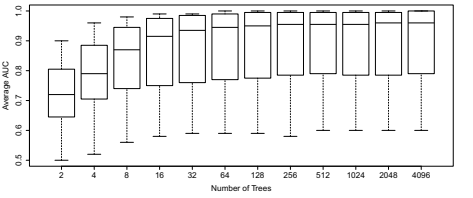


Fig. 2. AUC in the 8 low-density datasets

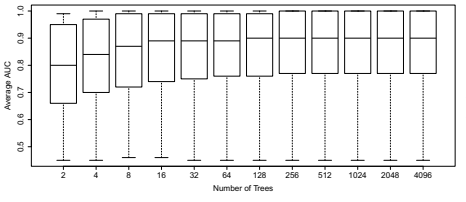


Fig. 3. AUC in the 21 high-density datasets

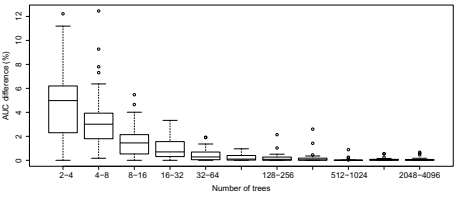


Fig. 4. AUC differences in all datasets

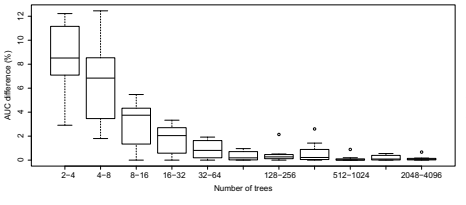


Fig. 5. AUC differences in the 8 low-density datasets

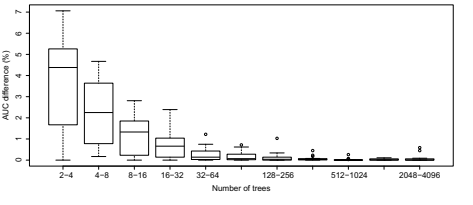


Fig. 6. AUC differences in the 21 high-density datasets

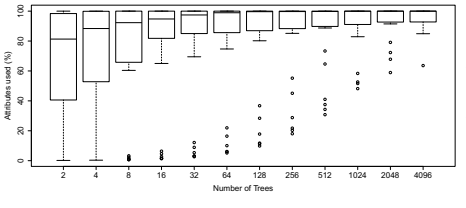


Fig. 7. Percentage of attributes used in all datasets

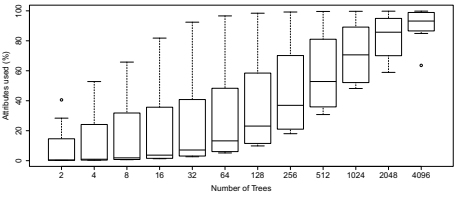


Fig. 8. Percentage of attributes used in the 8 low-density datasets

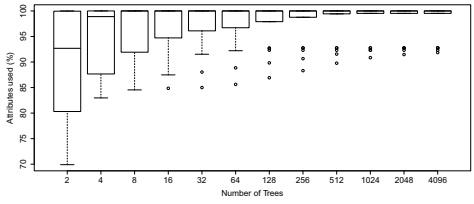


Fig. 9. Percentage of attributes used in the 21 high-density datasets

8 Conclusion

The results obtained show that, sometimes, a larger number of trees in a forest only increases its computational cost, and has no significant performance gain. They also indicate that mean and median AUC tend to converge asymptotically. Another observation is that there is no significant difference between using a number of trees within a Random Forest or its double. The analysis of 29 datasets shows that from 128 trees there is no more significant difference between the forests using 256, 512, 1024, 2048 and 4096 trees. The mean and the median AUC values do not present major changes from 64 trees. Therefore, it is possible to suggest, based on the experiments, a range between 64 and 128 trees in a forest. With these numbers of trees it is possible to obtain a good balance between AUC, processing time, and memory usage. We have also found an experimental relationship (inversely proportional) for AUC gain when doubling the number of trees in any forest.

Analyzing the percentage of attributes used, we can notice that using all datasets, the median reaches the full attribute set with 128 trees in the forest. If the total number of attributes is small, the median reaches the 100% with fewer trees (from 8 trees or more). If this number is larger, it reaches 100% with more trees, in some cases with more than 4096 trees. Thus, asymptotically the tendency indicates the Random Forest will use all attributes, and it is not interesting in some cases, for example in datasets with many attributes (i.e., gene expression datasets), since not all are important for learning the concept [26,14].

We have also proposed density-based metrics for datasets that probably capture some aspects of the VC dimension of decision trees. Under this assumption, low-density datasets may require large capacity learning machines composed by large Random Forests. The opposite also seems to be true.

Acknowledgments. This work was funded by FAPESP (São Paulo Research Foundation) as well as a joint grant between the National Research Council of Brazil (CNPq), and the Amazon State Research Foundation (FAPEAM) through the Program National Institutes of Science and Technology, INCT ADAPTA Project (Centre for Studies of Adaptations of Aquatic Biota of the Amazon).

References

1. Cancer program data sets. Broad Institute (2010), http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi
2. Dataset repository in arff (weka). BioInformatics Group Seville (2010), http://www.upo.es/eps/bigs/datasets.html
3. Datasets. Cilab (2010), http://cilab.ujn.edu.cn/datasets.htm
4. Aslan, O., Yildiz, O.T., Alpaydin, E.: Calculating the VC-dimension of decision trees. In: International Symposium on Computer and Information Sciences 2009, pp. 193–198 (2009)
5. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society Series B 57, 289–300 (1995)
6. Berks, M., Chen, Z., Astley, S., Taylor, C.: Detecting and Classifying Linear Structures in Mammograms Using Random Forests. In: Székely, G., Hahn, H.K. (eds.) IPMI 2011. LNCS, vol. 6801, pp. 510–524. Springer, Heidelberg (2011)
7. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
8. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
9. Demšar, J.: Statistical comparison of classifiers over multiple data sets. Journal of Machine Learning Research 7(1), 1–30 (2006)
10. Dubath, P., Rimoldini, L., Süveges, M., Blomme, J., López, M., Sarro, L.M., De Ridder, J., Cuypers, J., Guy, L., Lecoeur, I., Nienartowicz, K., Jan, A., Beck, M., Mowlavi, N., De Cat, P., Lebzelter, T., Eyer, L.: Random forest automated supervised classification of hipparcos periodic variable stars. Monthly Notices of the Royal Astronomical Society 414(3), 2602–2617 (2011), http://dx.doi.org/10.1111/j.1365-2966.2011.18575.x
11. Frank, A., Asuncion, A.: UCI machine learning repository (2010), http://archive.ics.uci.edu/ml
12. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Proceedings of the Thirteenth International Conference on Machine Learning, pp. 123–140. Morgan Kaufmann, Lake Tahoe (1996)
13. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. The Annals of Mathematical Statistics 11(1), 86–92 (1940)
14. Gamberger, D., Lavrač, N., Zelezny, F., Tolar, J.: Induction of comprehensible models for gene expression datasets by subgroup discovery methodology. Journal of Biomedical Informatics 37, 269–284 (2004)
15. Geremia, E., Menze, B.H., Clatz, O., Konukoglu, E., Criminisi, A., Ayache, N.: Spatial Decision Forests for MS Lesion Segmentation in Multi-Channel MR Images. In: Jiang, T., Navab, N., Pluim, J.P.W., Viergever, M.A. (eds.) MICCAI 2010. LNCS, vol. 6361, pp. 111–118. Springer, Heidelberg (2010)
16. Goldstein, B., Hubbard, A., Cutler, A., Barcellos, L.: An application of random forests to a genome-wide association dataset: Methodological considerations and new findings. BMC Genetics 11(1), 49 (2010), http://www.biomedcentral.com/1471-2156/11/49
17. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining Explor. Newsl. 11(1), 10–18 (2009)
18. Hsieh, C., Lu, R., Lee, N., Chiu, W., Hsu, M., Li, Y.J.: Novel solutions for an old disease: diagnosis of acute appendicitis with random forest, support vector machines, and artificial neural networks. Surgery 149(1), 87–93 (2011)

19. Kim, S.-H., Lee, J.-H., Ko, B., Nam, J.-Y.: X-ray image classification using random forests with local binary patterns. In: International Conference on Machine Learning and Cybernetics 2010, pp. 3190–3194 (2010)
20. Latinne, P., Debeir, O., Decaestecker, C.: Limiting the Number of Trees in Random Forests. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 178–187. Springer, Heidelberg (2001)
21. Lee, J.H., Kim, D.Y., Ko, B.C., Nam, J.Y.: Keyword annotation of medical image with random forest classifier and confidence assigning. In: International Conference on Computer Graphics, Imaging and Visualization, pp. 156–159 (2011)
22. Lempitsky, V., Verhoek, M., Noble, J.A., Blake, A.: Random Forest Classification for Automatic Delineation of Myocardium in Real-Time 3D Echocardiography. In: Ayache, N., Delingette, H., Sermesant, M. (eds.) FIMH 2009. LNCS, vol. 5528, pp. 447–456. Springer, Heidelberg (2009)
23. Leshem, G.: Improvement of adaboost algorithm by using random forests as weak learner and using this algorithm as statistics machine learning for traffic flow prediction. Research proposal for a Ph.D. Thesis (2005)
24. Liaw, A., Wiener, M.: Classification and regression by randomforest. R News 2/3, 1–5 (2002)
25. Mitchell, T.M.: Machine Learning. McGraw-Hill (1997)
26. Netto, O.P., Nozawa, S.R., Mitrowsky, R.A.R., Macedo, A.A., Baranauskas, J.A.: Applying decision trees to gene expression data from dna microarrays: A leukemia case study. In: XXX Congress of the Brazilian Computer Society, X Workshop on Medical Informatics, p. 10. Belo Horizonte, MG (2010)
27. Perez, P.S., Baranauskas, J.A.: Analysis of decision tree pruning using windowing in medical datasets with different class distributions. In: Proceedings of the Workshop on Knowledge Discovery in Health Care and Medicine of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD KDHCM), Athens, Greece, pp. 28–39 (2011)
28. Sirikulviriya, N., Sinthupinyo, S.: Integration of rules from a random forest. In: International Conference on Information and Electronics Engineering, vol. 6, pp. 194–198 (2011)
29. Tang, Y.: Real-Time Automatic Face Tracking Using Adaptive Random Forests. Master's thesis, Department of Electrical and Computer Engineering McGill University, Montreal, Canada (June 2010)
30. Vapnik, V., Levin, E., Cun, Y.L.: Measuring the vc-dimension of a learning machine. Neural Computation 6, 851–876 (1994)
31. Wang, G., Hao, J., Ma, J., Jiang, H.: A comparative assessment of ensemble learning for credit scoring. Expert Systems with Applications 38, 223–230 (2011)
32. Yaqub, M., Mahon, P., Javaid, M.K., Cooper, C., Noble, J.A.: Weighted voting in 3d random forest segmentation. Medical Image Understanding and Analysis (2010)
33. Yaqub, M., Javaid, M.K., Cooper, C., Noble, J.A.: Improving the Classification Accuracy of the Classic RF Method by Intelligent Feature Selection and Weighted Voting of Trees with Application to Medical Image Segmentation. In: Suzuki, K., Wang, F., Shen, D., Yan, P. (eds.) MLMI 2011. LNCS, vol. 7009, pp. 184–192. Springer, Heidelberg (2011)
34. Yi, Z., Criminisi, A., Shotton, J., Blake, A.: Discriminative, Semantic Segmentation of Brain Tissue in MR Images. In: Yang, G.-Z., Hawkes, D., Rueckert, D., Noble, A., Taylor, C. (eds.) MICCAI 2009. LNCS, vol. 5762, pp. 558–565. Springer, Heidelberg (2009)
35. Zhao, Y., Zhang, Y.: Comparison of decision tree methods for finding active objects. Advances in Space Research 41, 1955–1959 (2008)

Constructing Target Concept in Multiple Instance Learning Using Maximum Partial Entropy

Tao Xu[1], David Chiu[1], and Iker Gondra[2]

[1] School of Computer Science, University of Guelph
Ontario, Canada
[2] Mathematics, Statistics, Computer Science, St. Francis Xavier University
Nova Scotia, Canada

Abstract. Multiple instance learning, when instances are grouped into bags, concerns learning of a target concept from the bags without reference to their instances. In this paper, we advance the problem with a novel method based on computing the partial entropy involving only the positive bags using a partial probability scheme in the attribute subspace. The evaluation highlights what could be obtained if information only from the positive bags is used, while the contributions from the negative bags are identified. The proposed method attempts to relax the dependency on the distribution of the whole probability of training data, but focus only on the selected subspace. Experimental evaluation explores the effectiveness of using maximum partial entropy in evaluating the merits between the positive and negative bags in the learning.

Keywords: Maching learning, pattern recognition, multiple instance learning, total entropy, partial entropy.

1 Introduction

A multiple instance learning (MIL) problem can be defined as follows. Given a set of instances divided into bags, each of the bags is labelled positive if at least one instance in a bag coincides with a target concept, or negative if none of the instances has the target concept. If there is only one instance in each bag, then the MIL problem degrades to a standard supervised classification problem. From this perspective, MIL can be thought of as a generalization of the standard supervised learning. The difficulty of learning from training samples in this problem is that we are given only the class labels of the bags but not those of the instances.

MIL was initially formulated for characterizing drug behaviour [6]. Later, application domain extends to a wide variety of problems, including image analysis [4], drug discovery [14], content-based image retrieval [8], supervised image segmentation [7], stock selection [12], etc.

The first MIL scheme was proposed in 1997 by Dietterich *et al.* [6]. Their work was motivated by the drug activity prediction problem where a bag is a

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 169–182, 2012.

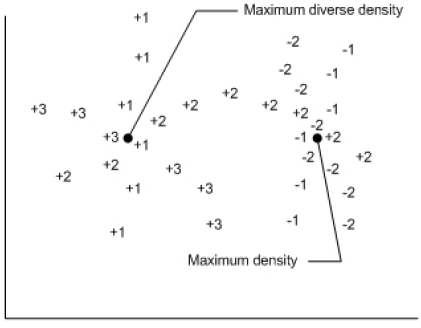


Fig. 1. The idea of MDD is to find areas that are close to at least one instance from every positive bag and far from instances in negative bags. In a two-dimensional feature space, each positive instance is denoted by the + sign with its bag number. Negative instances are similarly represented except that the - sign is used. The true concept point $\mathbf{t} \in \Re^2$ where the diverse density is maximized is not necessarily the point with the maximum density.

molecule (i.e., a drug) of interest and instances in the bag correspond to possible configurations (i.e., shapes) that the molecule is likely to take. The efficacy of a molecule (i.e., how well it binds to a "binding site") can be tested experimentally, but there is no way to control for individual configurations. Thus, the objective is to determine those shapes which will bind with a receptor molecule. In their approach, a set of shape features was extracted to represent the molecule in a high-dimensional feature space. Subsequently, in order to narrow down the possible conformations that cause the special property, the smallest axis-parallel hyper-rectangle (APR) that bounds at least one instance from each positive bag but excludes any negative ones is found. The execution order of the program could be either "outside-in": starting from a bound that includes all instances in positive bags and keeps shrinking it until all false positive instances are excluded; or "inside-out": starting from an instance in a positive bag and grows it until the smallest APR is obtained. Subsequently, a PAC-learning based scheme was proposed in [11] and a few publications [3,2] followed up along this direction.

Maximum diverse density (MDD) [12] is popular for its conceptual intuition. Based on the fact that a prototype should be close to at least one instance in each positive bag while far from all negative ones, a maximum likelihood estimate can be formulated to find the most likely estimate(s) of the prototype(s). Assuming a unique prototype $\mathbf{t} \in \Re^d$ accounting for the labels for all bags, $\mathbf{t}$ is located in a region that is not only dense in positive instances but also diverse in that it includes at least one instance from each positive bag (See Fig. 1).

Because the instance labels in a positive bag are assumed unknown, Zhang and Goldman [18] extended MDD by adopting hidden variables to model unknown instance labels. They proposed an expectation-maximization version of MDD (or EM-DD) in the likelihood estimation. In each E-step, the most likely positive

instance in each bag is selected and used in estimating the new true concept in M-step. EM-DD largely simplifies the computational complexity of MDD and achieves a remarkable performance improvement on the 'MUSK' data set.

The method Citation k nearest neighbor (kNN) has been proposed for MIL [15]. It is a generalization of the standard kNN by introducing a bag-level distance metric. The label of a query bag is predicted not only by the majority voting of its k nearest bags but also the number of times that the bag is cited by them. The introduction of citers evidently enhances the robustness of the method over the standard kNN. On the other hand, suppose every positive bag is fairly loaded in positive instances, then by associating each instance with the label of the bag it belongs to, support vector machine (SVM) may achieve a competitive performance [13]. For example, Andrews et al. [1] treated the unobservable instance labels as hidden variables and formulated the MIL problem as a SVM problem in which the optimization of the margin between different classes is subject to the joint constraints of a kernelized discriminant function and unknown labels. Other approaches include the multi-decision tree [5] which is an extension of C4.5 decision tree for MIL, and artificial neural network (ANN) variants for MIL [17,10].

Despite the availability of several methods for solving the MIL problem, a more practical approach is still desirable. APR is simple yet effective but has the risk of not finding such a rectangle that contains no negative instances. As a method based on maximum likelihood estimation, MDD requires prior knowledge about the number of true concepts that is usually unknown in practice. In the absence of this knowledge, MDD ends up with false estimates. As indicated in [12], applying the single-prototype formulation to the problem with two (or more) distinct prototypes results in an estimate close to neither one of them but somewhere in between. This was also verified in the experiments conducted in [7,8]. Citation kNN presents robustness by taking into account the impact of citers but is still sensitive to local variations. For SVM-based approaches, a proper kernel function (as well as the corresponding parameter values) has to be decided empirically.

In our previous work [16], we have made an attempt with *adaptive kernel diverse density estimate* to address the issue of the coexistence of multiple true concepts, which is common in textual analysis and image understanding problems. In this paper, we consider another theoretical criterion for solving the problem by investigating the properties of partial entropy. Briefly, using the instance-bag distance as a measure of the degree of an instance belonging to a bag, the partial entropy taking into account all training samples is defined as the amount contributed by the positive bags to the total entropy. As shown in this paper, to maximize the partial entropy is therefore to reduce the inter-class uncertainty, and increase the intra-class certainty. When related to MIL, it is equivalent to locate in the instance space a point that is far from negative bags while close to all the positive bags simultaneously. Thus, the instances that maximize the partial entropy summarize that to the target concept(s).

The rest of the paper is organized as follows. Section II gives the definition of partial entropy, and investigates some useful properties for later analysis. Section III explains the rationale for using the maximum partial entropy to the MIL problem. Preliminary experimental results are given in section IV.

2 Partial Entropy

In information theory, entropy is a measure that quantizes the information contained in a message, usually in the unit of bits. This measure gives the absolute bound on lossless compression if a message is assumed to be a sequence of independently and identically distributed random variables. Mathematically, the entropy for a random variable x with N possible outcomes $X = \{x_1, \ldots, x_N\}$ is defined as

$$H(p_1, \ldots, p_N) = -\sum_{i=1}^{N} p_i \log_2 p_i, \tag{1}$$

where $p_i = Pr(x = x_i)$ denotes the probability of taking the i^{th} outcome and $-\log_2 p_i$ is known as the corresponding self-information. In this sense, entropy is the expected amount of information provided by X. Hereafter, we use log to denote $\log_2$ to simplify the notations.

Entropy is also frequently used as a measure of uncertainty for that it is maximized only if all outcomes are equally likely, i.e.,

$$H(p_1, \ldots, p_N) \leq H(\frac{1}{N}, \ldots, \frac{1}{N}) = \log N \tag{2}$$

Denoted by X_M a subset of X with M $(1 \leq M \leq N)$ outcomes, the quantity

$$H'(p_1, \ldots, p_M) = -\sum_{i=1}^{M} p_i \log_2 p_i, \tag{3}$$

is defined as *partial entropy* [9] on the subset X_M. From now on, we use H' to denote partial entropy.

We reveal that equal probabilities over X_M result in the maximum partial entropy H' by justifying the following Lemma.

Lemma 1. *Let x be a random variable with outcomes $X = \{x_1, \ldots, x_N\}$, then a uniform distribution over a subset $X_M \subseteq X$ has the maximum partial entropy.*

Proof: Let $0 \leq \alpha \leq 1$ be the sum of probabilities $\{p_i\}_{i=1,\ldots,M \leq N}$ for the outcomes in subset X_M, then

$$\sum_{i=1}^{M} p_i = \alpha \Rightarrow \sum_{i=1}^{M} \frac{p_i}{\alpha} = 1,$$

which defines a *partial probability scheme* $\{p_1/\alpha, \ldots, p_M/\alpha\}$ with total entropy expressed as

$$\begin{aligned} H &= -\sum_{i=1}^{M} \frac{p_i}{\alpha} \log \frac{p_i}{\alpha} \\ &= -\frac{1}{\alpha} \sum_{i=1}^{M} (p_i \log p_i - p_i \log \alpha) \\ &= -\frac{1}{\alpha} \sum_{i=1}^{M} p_i \log p_i + \frac{\log \alpha}{\alpha} \sum_{i=1}^{M} p_i \\ &= -\frac{1}{\alpha} \sum_{i=1}^{M} p_i \log p_i + \frac{\log \alpha}{\alpha} \alpha \\ &= -\frac{1}{\alpha} \sum_{i=1}^{M} p_i \log p_i + \log \alpha. \end{aligned}$$

By Ineq.2, we have $H \leq \log M$. Therefore,

$$-\frac{1}{\alpha} \sum_{i=1}^{M} p_i \log p_i + \log \alpha \leq \log M,$$

and thus

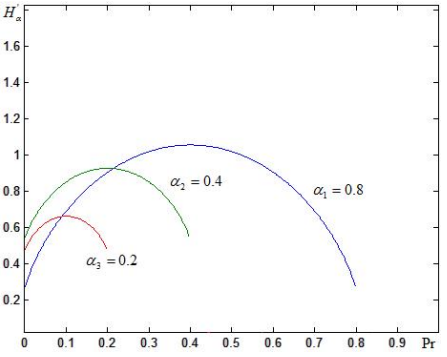
$$H' = -\sum_{i=1}^{M} p_i \log p_i \leq \alpha(\log M - \log \alpha). \tag{4}$$

It is easy to verify that partial entropy H' is a concave function of variables $\{p_i\}$ and the unique upper bound $\alpha(\log M - \log \alpha)$ is achieved only if all p_i are the same, i.e., $p_i = \alpha/M$.

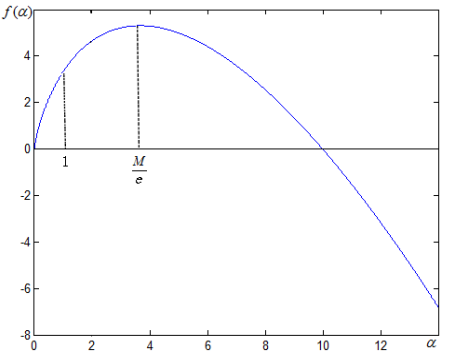
Q.E.D

Ineq.4 generalizes Ineq.2 to any non-empty subset of X, and when $X_M = X$ partial entropy becomes total entropy. Lemma 1 justifies the intuition that a non-empty subset X_M with equally probable outcomes has the maximum partial entropy. This property is true independent of M. Fig. 2(a) demonstrates this point graphically by indicating the locations of maximum of the curves for different α values. Nevertheless, another noticeable phenomena in the figure is that the greater the α value is, the higher the maximum partial entropy will be. To avoid confusion, hereafter we use the term *α-level local maximum partial entropy* (LMPE) to express the maximum partial entropy associated with a specific α value. The tight relation between the maximum partial entropy and α has been implied in Ineq.4, and we state it formally as the following Lemma.

Lemma 2. *For any subset $X_M \subseteq X$ with $M \geq 3$ equally probable outcomes, the maximum partial entropy H' is non-decreasing for $0 \leq \alpha \leq 1$.*



(a) Partial entropy contributed by a binary subset ($M = 2$) with different α



(b) Maximum partial entropy as a function of α (plotted with $M = 10$)

Fig. 2. (a) A uniform distribution over outcomes in a subset X_M results in the maximum partial entropy. (b) Maximum partial entropy increases as α increases in $[0, 1]$ for $M \geq 3$. This is because the maximum partial entropy function $f(\alpha)$ has the maximum at $\alpha = M/e$, for $M \geq 3$ the interval $[0, 1]$ always falls into the increasing part of $f(\alpha)$.

Proof: If X_M has equal probabilities for all M outcomes, equality always holds in Ineq.4 and the right side of it gives the LMPE for a certain α. We can thus express the LMPE with respect to different α as a function of α:

$$\begin{aligned} f(\alpha) &= \sup\{-\sum_{i=1}^{M} p_i \log p_i \;\mid\; \sum_{i=1}^{M} p_i = \alpha\} \\ &= \alpha(\log M - \log \alpha), \end{aligned} \tag{5}$$

which is termed as the *local maximum partial entropy function*. We prove this Lemma by showing that $f(\alpha)$ is concave, and $[0, 1]$ is contained in the increasing part of $f(\alpha)$.

Let b ($b = 2$ in our case) denote the base for the logarithm we are using, take the first-order derivative of Eq.5 and make it to zero

$$f'(\alpha) = \log M - \log \alpha - \frac{1}{\ln b} = 0,$$

we get

$$\log \alpha = \log M - \log b^{(1/\ln b)} = \log M - \log e,$$

so

$$\alpha = \frac{M}{e}$$

which is beyond $[0, 1]$ on the right for $M \geq 3$. Taking the second-order derivative results in

$$f''(\alpha) = -\frac{1}{\ln b}\alpha^{-1}.$$

Because $f''(\alpha)$ is negative for $\alpha \in \Re^+$, $f(\alpha)$ is concave in $\Re^+$. Combined with the previous conclusion that $f(\alpha)$ is monotone in $[0, M/e]$, it is proved that $f(\alpha)$ is increasing in $[0, 1]$ given $M \geq 3$. The maximum attainable value of $f(\alpha)$ is thus $f(\alpha = 1) = \log M$, which is different from the maximum of $f(\alpha)$ at $\alpha = M/e$.

Q.E.D

Lemma 2 simply states that the local maximum partial entropy function (Eq. 5) is increasing in partial sum probabilities α, given that the the cardinality of the subset is at least three. Fig. 2(b) explains Lemma 2 with a graph for $M = 10$. Although α, as the partial sum of probabilities, could never be greater than one, the curve shown here relaxes the domain so its trend can be easily observed. More importantly, $f(\alpha)$ is never strictly increasing in $[0, 1]$ unless $M \geq 3$.

An immediate conclusion we can draw is that α determines the upper-bound that the maximum partial entropy can reach, i.e., the maximum partial entropy is the LMPE with the maximum α. Hence, if one is pursuing the maximum partial entropy to the most possible extent, α should be as large as possible while maintaining equal probabilities over outcomes in the subset. We also notice that $M \geq 3$ is critical because for $M < 3$, the monotonicity does not hold in $[0, 1]$. Hence, $M = 3$ is the lower bound of the cardinality of a subset to support the statement.

Up to this point, we have established the theoretical properties that are useful in formulating the MIL problem. To summarize the discussions, let us assume, without loss of generality, that X can be divided into two subsets X_M (positive class) and $X_M^c = X - X_M$ (negative class). In order to maximize the partial entropy $H'(X_M)$ on X_M, one must:

- Following Lemma 2 to increase α by decreasing the inter-class information (between X_M^c and X_M) so as to increase the upper bound of $H'(X_M)$.
- Following Lemma 1, increase the intra-class information (within X_M) by balancing probabilities among all the outcomes in X_M.

The next section explains how we make use of these two properties to analyze MIL so as to construct a generic concept based on the positive bag information while taking the contributions of the negative bags into consideration.

3 Approaching MIL by Maximizing Partial Entropy

Let $\mathcal{B}$ denote the entire MIL training set composed of positive samples and negative samples, i.e, $\mathcal{B} = \{\mathcal{B}^+, \mathcal{B}^-\} = \{\{B_i^+\}, \{B_i^-\}\}$. Assuming a continuous instance space, we estimate the probability of a point $\mathbf{x}$ in this space belonging to a bag B_i (regardless of the label) as

$$Pr(\mathbf{x} \in B_i) = \frac{g(dist(\mathbf{x}, B_i))}{\sum_{B_j \in \mathcal{B}} g(dist(\mathbf{x}, B_j))}, \tag{6}$$

where $g(\cdot)$ is a decreasing function to indicate a higher estimate for a shorter distance, and $dist(\mathbf{x}, B_i)$ is the instance-bag distance defined as the Euclidean distance from the nearest instance in B_i to $\mathbf{x}$:

$$dist(\mathbf{x}, B_i) = \inf_{B_{i,j} \in B_i} \{\|\mathbf{x} - B_{i,j}\|_2\}, \tag{7}$$

where $B_{i,j}$ is the j-th instance in B_i. This probability estimate agrees with the intuition that the closer $\mathbf{x}$ to a bag B_i, the greater the probability $\mathbf{x}$ to be assigned to B_i. In this approach, we take a more general probability estimate that distinguishes the importance of the positive and negative bags with a weighting parameter $\gamma \in \Re^+$:

$$Pr(\mathbf{x} \in B_i^+) = \frac{g(dist(\mathbf{x}, B_i^+))}{\sum_{B_j^+ \in \mathcal{B}^+} g(dist(\mathbf{x}, B_j^+)) + \gamma \sum_{B_j^- \in \mathcal{B}^-} g(dist(\mathbf{x}, B_j^-))}. \tag{8}$$

Clearly, Eq.6 and Eq.8 produce the same probability estimate for $\gamma = 1$. Here, we only need to consider the probability measure of positive bags $Pr(\mathbf{x} \in B_i^+)$ because it is what is necessary to compute the partial entropy on the positive subset. Note that the formulation of probabilities with respect to the entire training set $\mathcal{B}$ is important for the reason to be interpreted later in this section. The total entropy based on this probability estimate is thus

$$\begin{aligned} H(\mathbf{x} \in \mathcal{B}) &= H'(\mathbf{x} \in \mathcal{B}^+) + H'(\mathbf{x} \in \mathcal{B}^-) \\ &= -\sum_{B_i^+ \in \mathcal{B}^+} Pr(\mathbf{x} \in B_i^+) \log Pr(\mathbf{x} \in B_i^+) \\ &\quad - \sum_{B_i^- \in \mathcal{B}^-} Pr(\mathbf{x} \in B_i^-) \log Pr(\mathbf{x} \in B_i^-). \end{aligned} \tag{9}$$

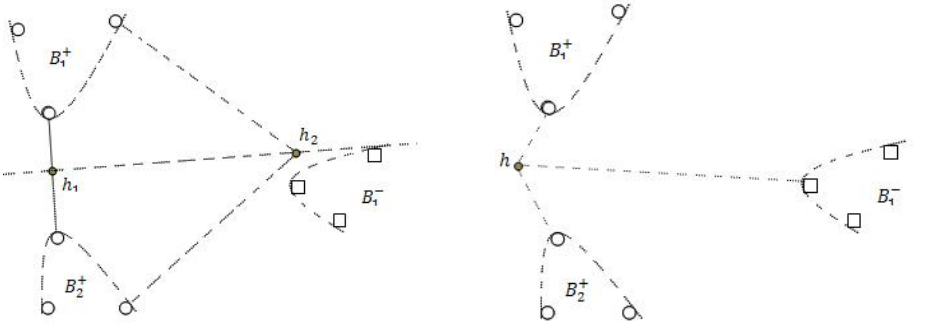
Taking away the amount from the negative bags, we are left with the partial entropy contributed by the positive ones:

$$H'(\mathbf{x} \in \mathcal{B}^+) = -\sum_{B_i \in \mathcal{B}^+} Pr(\mathbf{x} \in B_i) \log Pr(\mathbf{x} \in B_i). \tag{10}$$

Note that the impact of negative bags is implicitly included in Eq.10 due to the formulation of probabilities given in Eq.8 (as part of the the denominator).

For $\gamma > 1$, it implies that the impact of negative bags is considered more important, and as γ grows this impact becomes more important. To the other extreme end where $\gamma = 0$, it is thus ignoring all negative bags, and the partial entropy is then nothing but the total entropy on positive bags. The consequence is that an estimate is somewhere close to all the positive bags but not subject to the restriction of the negative ones. We will compare the difference between partial entropy and total entropy at the end of this section.

We then show that partial entropy as given by Eq.10 provides a reasonably good measure for estimating the target concepts of MIL.



(a) Maximum total entropy defined on positive bags

(b) Maximum partial entropy contributed by positive bags

Fig. 3. (a) Maximum total entropy $H(\mathbf{x} \in \mathcal{B}^+)$ defined on B_1^+ and B_2^+ corresponds to points at the same distance to the nearest instance in each bag. Hence, every point on the line through h_1 and $h2$ results in the maximum total entropy. E.g., the obviously false hypothesis h_2 has the same maximum entropy measure as h_1: $H'(\mathbf{h_1} \in \mathcal{B}^+) = H'(\mathbf{h_2} \in \mathcal{B}^+)$. (b) Maximum partial entropy $H'(\mathbf{x} \in \mathcal{B}^+)$ in contrast, does not suffer from the same problem because the estimate h will be pushed away from B_1^- and dragged towards the center of B_1^+ and B_2^+. Also notice that the estimate will be a little off the line connecting B_1^+ and B_2^+ due to the impact from B_1^-. The weighting parameter γ can be used to control the degree of the deviation.

As an optimization criterion for the MIL problem, we expect the estimates to be close to all the positive bags while far from the negative ones. Furthermore, we demand the estimates to be close to all the positive bags at roughly the same distance. These preferences are naturally satisfied by maximizing the partial entropy given in Eq.10. The rationale is supported by :

- According to Lemma 2, maximizing $H'(\mathbf{x} \in \mathcal{B}^+)$ by Eq.10 will force an estimate away from negative bags towards the positives so as to maximize α (the proportion of probabilities over the positive bags).
- According to Lemma 1, maximizing $H'(\mathbf{x} \in \mathcal{B}^+)$ given α being maximized in previous step will equalize the distances from $\mathbf{x}$ to all the positive bags and thus the maximum $H'(\mathbf{x} \in \mathcal{B}^+)$.

Hence, maximizing $H'(\mathbf{x} \in \mathcal{B}^+)$ results in an estimate $\mathbf{x}$ close to all positive bags at roughly the same distance while far from negative ones. In terms of entropy, the estimate presents the maximum inter-class (between the positive and negative bags) certainty as well as the maximum intra-class (within the positive bags) uncertainty (equal probabilities). All of these summarize the characteristics of the true concepts of MIL. Therefore, any $\mathbf{x}$ that maximizes the partial entropy given by Eq.(10) can be thought of as a good estimate of the true concept of MIL, i.e.,

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{x}} H'(\mathbf{x} \in \mathcal{B}^+) \tag{11}$$

Again, in order to have an unbiased estimate, one should be cautious when selecting γ. A small γ (e.g., $\gamma = 0$) leads to a result that equalizes the distances to all positive bags with less consideration of the impact from negative ones. In the absence of knowledge that one class of bags is more important than the other, it is normal to choose $\gamma = 1$.

Now, one might be curious about if the same goal can be achieved by the total entropy defined on positive bags only, i.e.,

$$H(\mathbf{x} \in \mathcal{B}^+) = -\sum_{B_i^+ \in \mathcal{B}^+} Pr(\mathbf{x} \in B_i^+) \log Pr(\mathbf{x} \in B_i^+) \tag{12}$$

given probabilities defined as

$$Pr(\mathbf{x} \in B_i^+) = \frac{g(dist(\mathbf{x}, B_i^+))}{\sum_{B_j^+ \in \mathcal{B}^+} g(dist(\mathbf{x}, B_j^+))}, \tag{13}$$

The difference between Eq.10 and Eq.12 appears unobvious but significant in consequence, hidden behind the formulation of probabilities. Eq.8 takes the impact of negative samples into account (as part of denominator) while Eq.13 does not. This little difference accounts for the change in effect. Ignoring negative samples could leads to uncertainty in the estimate. This is illustrated as a simple example in Fig. 3.

4 Implementation and Experiments

Only for simple cases with a unique global maximum, it is possible to maximize α independent of pursuing an equal probability distribution over the subset. However, for general problems with a complicated function landscape, it is prohibitive to separate the optimization process into isolated steps. Without a closed-form solution, a gradient-based method can be used to search all local maxima of $H'(\mathcal{B}^+)$, in which $g(\cdot)$ in Eq.8 is simply chosen as

$$g(z) = \frac{1}{z}. \tag{14}$$

Because the target positive concept is located somewhere among the positive bags, the algorithm starts with every instance in all the positive bags so as to get the global maximum.

4.1 Artificial Datasets

In order to verify the correctness of the proposed scheme, we first applied the algorithm to an artificial data set of 100 bags in a 10-dimensional space. In this experiment, every bag contained 20 randomly generated instances restricted in the hyper-rectangle $[0.0, 1.0]^{10}$. A bag was labeled positive if any of its instances fell into the hyper-rectangle $[0.29, 0.31]^{10}$, or negative otherwise. The generating

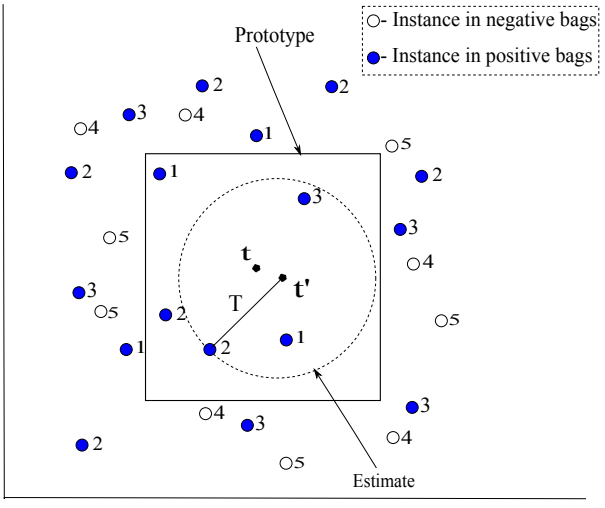


Fig. 4. There are 5 bags in this example (3 positive, 2 negative). Instances from each bag are shown together with their corresponding bag numbers. The rectangle represents the predefined true concept centered at $\mathbf{t}$. The dashed hypersphere centered at the estimate $\hat{\mathbf{t}}$ with threshold T as the radius is used for classifying future unlabeled bags (i.e., positive if it falls within the hypersphere or negative otherwise).

algorithm was designed to ensure that at least 30% of the bags in the training set were positive.

For performance evaluation, we adopted the leave-one-out cross validation strategy. For each round of the experiment, all bags except one take part into the training and the left-out is used for verification. The label of the left-out is predicted as positive if any of its instances falls into the hyper-sphere centered at the acquired prototype $\hat{\mathbf{t}}$ with a threshold (radius) T. A reasonable threshold T can be defined as the distance from the estimate to the furthest positive bag (Fig. 4 illustrates this):

$$T = \max_{B_i^+ \in \mathcal{B}^+} \{dist(\hat{\mathbf{t}}, B_i^+)\}.$$

$\gamma = 1$ was used for all experiments on the artificial data set. The program first normalizes all the instances into $[0, 1]^d$, dimension by dimension, and then performs the hill-climbing algorithm from every instance in positive bags. All obtained intermediate maxima are ranked for the final decision of the best estimate.

The performance measure was obtained as the consistent ratio between the number of the correct predictions and the total number of experiments. The experiment on such 20 artificial data sets resulted in an average correctness ratio of 98.6% with all obtained target concepts within 0.30 ± 0.0023 in each dimension. Those missed predictions were due to our particular choice of threshold strategy used (see Fig. 4). By increasing T with a factor 1.05, we can get perfect correctness

ratio occasionally. However, when $\hat{\mathbf{t}}$ does not coincide very well with the center of the predefined target concepts, the false positive error might happen.

4.2 MUSK Datasets

We also experimented with two 'MUSK' data sets (from UCI machine learning repository) from the drug-discovery domain, using the same parameter settings that were used for the artificial data sets. The data set MUSK#1 consists of 92 molecules represented in a 166-dimensional features space with 47 samples labeled positive by human experts. MUSK#2 consists of 102 samples of which 39 are positive. The number of instances in MUSK#2 is roughly 10 times that of MUSK#1.

The performance measures on MUSK#1 and MUSK#2 were 80.1% and 84.2% respectively for $\gamma = 1$. We intentionally do not include a comparison with other algorithms on the MUSK data sets for two reasons. Firstly, because we do not have the exact knowledge of the target concepts in the MUSK data sets, a high accuracy on the MUSK data sets does not necessarily mean a good understanding of the underlying patterns. Secondly, reports on the MUSK data set were all based on the results obtained with the optimal parameter settings. This can be misleading because exhaustive parameter tuning is nearly impractical for real problems. Therefore, we also encourage researchers to post experimental results on data sets that have explicit knowledge of the target concepts for an objective evaluation.

The performance was very sensitive to the threshold strategy adopted. By manually adjusting T by a factor within $[0.5, 1.5]$, the measure could vary from 58% to 86% ($\gamma = 1$) on MUSK#2. In contrast, adjusting γ did not have significant influence on the results for $\gamma \geq 1$. The best performance measure obtained was 87% for $\gamma = 10$ and $T = 1.2$. When $\gamma < 0.2$, there was a persisting low correctness ratio below 55% regardless of the selection of threshold factor.

The experimental result from MUSK#2 was better than the one from MUSK#1. An explanation for this is that because there are more instances per bag in MUSK#2, more accurate (or compact) T was estimated and thus less false positive predictions were made. It is thus concluded that the proposed algorithm is very suitable for large training set. Through the intermediate outputs, there was also a strong evidence of the presence of multiple true concepts in both MUSK#1 and MUSK#2. The differences in measure between the first few candidate estimates are very small for both MUSK#1 and MUSK#2. Experiments using multiple true concepts will be carried out in the future for verification.

5 Conclusion

In this paper, we make a preliminary study on partial entropy and use the result to approach MIL. We first show that partial entropy generalizes the entropy in the sense that a subset with equally probable outcomes has the maximum

partial entropy, thus equalizing the information from the subset. We next show that the maximum partial entropy function is non-decreasing in the sum of self information of a subset. Applying these two properties, we solve the MIL problem by maximizing the partial entropy contributed by the positive bags. It in turn gives us estimates close to all the positive bags at roughly equal distance but far from all the negative ones, as an effective classifier. Notice that no explicit assumptions are required for the formulation.

References

1. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: Advances in Neural Information Processing Systems, vol. 15, pp. 561–568 (2003)
2. Auer, P.: On learning from multi-instance examples: empirical evaluation of a theoretical approach. In: Proceedings of the 4th International Conference on Machine Learning, pp. 21–29 (1997)
3. Blum, A., Kalai, A.: A note on learning from multiple-instance examples. Machine Learning 30(1), 23–29 (1998)
4. Bolton, J., Gader, P.: Multiple instance learning for hyperspectral image analysis. In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium, pp. 4232–4235 (2010)
5. Chevaleyre, Y., Zucker, J.D.: Solving multiple-instance and multiple-part learning problems with decision trees and rule sets. Application to the mutagenesis problem. In: Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence, pp. 204–214 (2001)
6. Dietterich, T.G., Lathrop, R.H., Pérez, T.L.: Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence 89(1-2), 31–71 (1997)
7. Gondra, I., Xu, T.: Adaptive mean shift-based image segmentation using multiple instance learning. In: Proceedings of the Third IEEE International Conference on Digital Information Management, pp. 716–721 (2008)
8. Gondra, I., Xu, T.: Image region re-weighting via multiple instance learning. Signal, Image and Video Processing 4(4), 409–417 (2010)
9. Guiasu, S.: Information Theory with Applications. McGraw-Hill (1977)
10. Li, C.H., Gondra, I.: A novel neural network-based approach for multiple instance learning. In: Proceedings of the 2010 10th IEEE International Conference on Computer and Information Technology, CIT 2010, pp. 451–456. IEEE Computer Society, Washington, DC (2010)
11. Long, P.M., Tan, L.: PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples. In: Proceedings of the 9th Annual Conference on Computational Learning Theory, pp. 228–234 (1996)
12. Maron, O., Perez, T.L.: A framework for multiple-instance learning. In: Proceedings of the Advances in Neural Information Processing Systems, pp. 570–576 (1998)
13. Tao, Q., Scott, S., Vinodchandran, N.V., Osugi, T.T.: SVM-based generalized multiple-instance learning via approximate box counting. In: Proceedings of the 21st International Conference on Machine Learning, pp. 779–806 (2004)
14. Teramoto, R., Kashima, H.: Prediction of protein-ligand binding affinities using multiple instance learning. Journal of Molecular Graphics and Modelling 29(3), 492–497 (2010)

15. Wang, J., Zucker, J.D.: Solving the multiple-instance problem: A lazy learning approach. In: Proceedings of the 17th International Conference on Machine Learning, pp. 1119–1126 (2000)
16. Xu, T., Gondra, I., Chiu, D.: Adaptive kernel diverse density estimate for multiple instance learning. In: Proceedings of the 7th International Conference on Machine Learning and Data Mining in Pattern Recognition, MLDM 2011, pp. 185–198 (2011)
17. Zhang, M.L., Zhou, Z.H.: Adapting RBF neural networks to multi-instance learning. Neural Process. Lett. 23(1), 1–26 (2006)
18. Zhang, Q., Goldman, S.A.: EM-DD: An improved multiple-instance learning technique. In: Advances in Neural Information Processing Systems, vol. 14, pp. 1073–1080. MIT Press (2001)

A New Learning Structure Heuristic of Bayesian Networks from Data

Heni Bouhamed[1], Afif Masmoudi[2], Thierry Lecroq[1], and Ahmed Rebaï[3]

[1] University of Rouen, LITIS EA 4108, 1 rue Thomas Becket,
76821 Mont-Saint-Aignan cedex, France
Heni.bouhamed@yahoo.fr,
Thierry.lecroq@univ-rouen.fr
[2] Department of Mathematics, Faculty of Science of Sfax, Soukra B.P 802 Sfax, Tunisia
Afif.masmoudi@fss.rnu.tn
[3] Bioinformatics Unit, Centre of Biotechnologie of Sfax, 3018 Sfax, Tunisia
Ahmed.rebai@cbs.rnrt.tn

Abstract. Nowadays, Bayesian Networks (BNs) have constituted one of the most complete, self-sustained and coherent formalisms useful for knowledge acquisition, representation and application through computer systems. Yet, the learning of these BNs structures from data represents a problem classified at an NP-hard range of difficulty. As such, it has turned out to be the most exciting challenge in the learning machine area. In this context, the present work's major objective lies in setting up a further solution conceived to be a remedy for the intricate algorithmic complexity problems imposed during the learning of BN-structure through a massively-huge data backlog. Our present work has been constructed according to the following framework; on a first place, we are going to proceed by defining BNs and their related problems of structure-learning from data. We, then, go on to propose a novel heuristic designed to reduce the algorithmic complexity without engendering any loss of information. Ultimately, our conceived approach will be tested on a car diagnosis as well as on a Lymphography diagnosis data-bases, while our achieved results would be discussed, along with an exposition of our conducted work's interests as a closing step to this work.

Keywords: Bayesian Network, structure learning, modeling, algorithmic complexity.

1 Introduction

The huge amounts of data, made recently available, pertaining to the various research fields, have made it crucially critical for the learning techniques to be efficient, in so far as the processing of complex data dependences is concerned. Owing to their flexibility and easily-recognizable mathematical formulations, BNs are most often the basic selected model opted for in a wide-array of application-fields whether astronomic, textual, bioinformatics and web-mining applications. Yet, with an incredibly huge

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 183–197, 2012.

number of variables, the learning BNs structure from data remains a big challenge to be retained and considered in terms of calculation power, algorithmic complexity and execution time [1]. Most recently, however, various algorithms have been developed with respect to the BNs learning structures from data-bases [2, 3, 4]. A considerable class of these algorithms rests on the metric-scoring methods, excessively compared and exhaustively applied as approaches [5, 6]. Nevertheless, these algorithms and scoring methods remain still insufficient with regards to those cases in which the number of variables exceeds hundreds of thousands [7]. In so far as our work is concerned, these algorithms and metric scores are not going to be dealt with or questioned. Rather, we seek to further enrich them through a new heuristic based on clustering pertaining to structure learning, in a bid to further reduce the algorithmic complexity as well as the execution time, with the purpose of modeling some previously non-modelizeable information systems, by using, exclusively, the underway available algorithms.

Our work, we reckon, is critically important for a number of various reasons. First, we have managed to demonstrate, throughout its scope, that by wholly subdividing an information system into sub-sets, we tend to dramatically reduce the number of possible structures necessary for learning the BNs structures. Second, a special heuristic has been devised and proposed whereby this reduction could be exploited without engendering any significant loss of data. Ultimately, by combining our proper heuristic with the existing prevailing structure-learning algorithms, one can considerably reduce the extent of algorithmic complexity as well as the learning of BNs structure from data execution time, in such a way that even a large number of non-modelizeable variables could be treated or processed.

The remainder of this article has been arranged as follows. The next upcoming section deals with the BNs and their structure learning problems. In the following section, we are going to put forward a new heuristic which we shall test upon a car-diagnosis and Lymphography diagnosis data bases. Finally, we will close up our work by concluding and paving the way for certain potential perspectives relevant to future researches.

2 BNs and Structure Learning from Data Problems

It is worth highlighting that knowledge representation and the related reasoning, thereof, have given birth to numerous models. The graphic probability models, namely, BN, introduced by Judea Pearl in the 1980s, have been manifested in the practical tools useful for the representation of uncertain knowledge and reasoning process from incomplete information.

Hence, the BN graphic representation indicates the dependences (or independences) between variables and provides a visual knowledge representation tool, that turns out to be more easily understood by its users. Furthermore, the use of probability allows to take into account the uncertainty, by quantifying the dependences between variables. These two properties have been at the origin of the first terms allotted, initially, of BN, "probabilistic expert systems", where the graph used to be

compared with some set rules pertaining to a classic expert system, and conditional probability presented as a quantification measurement of the uncertainty related to these rules.

The number of all BN possible structures has been shown to ascend sharply as a super-exponential on the number of variables. Indeed, Reference [12] derived the following recursive formula for the number of Directed Acyclic Graph (DAG) with n variables:

$$r(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-1)} r(n-i) = n^{2^{O(n)}} \tag{1}$$

which gives: $r(1)=1$, $r(2)=3$, $r(3)=25$, $r(5)=29281$, $r(10)=4{,}2\times10^{18}$

This means that, it is impossible to perform an exhaustive search of all structures in a reasonable time in cases the number of nodes exceeds seven. In fact, most structure-learning methods use heuristics to search the DAGs space.

3 A New Clustering-Based Heuristic: Theoretical Framework and Methodology

The idea lying behind our conceived proceeding lies in the rapid super-exponential surge of algorithmic complexity of learning BN structure from data with respect to the rise in the number of variables. To remediate this problem, our preconceived idea consists in subdividing the variables into subsets (or clusters), by treating each cluster's learning structure separately, while looking for a convenient procedure whereby the different structures could be assembled into a final structure version. In this regard, it has been noticed that in numerous information systems, so as not to say in most of them, there exists, at least, one single central variable of a global interest constituting the basis of the system's modelization. In this respect, we reckon to execute the processing of each cluster learning structure with the central interest variable, then, proceed by assembling the different various structures around this central variable as a next step.

In the upcoming part (3.1), we shall demonstrate, mathematically, that by subdividing the variables and by separately processing each cluster's learning structure with the interest variable, we dramatically reduce the number of possible DAG in respect of the simultaneous learning structure of the entire variables. After that, in part (3.2), we are going to explain our proposed framework procedure as well as the methodologies to be pursued.

3.1 Theoretical Background

The below represented Robinson formula depicts the number of possible DAG in respect of the variables' number:

The Robinson formula is $r(n) = \sum_{i=1}^{n} (-1)^{i+1} 2^{i(n-i)} \binom{n}{i} r(n-i)$; $r(1) = 1$, where n stands for the number of variables.

In this section, we will prove that $r(n) > \sum_{l=1}^{k} r(J_l + 1)$, where $n\text{-}1 = J_1 + J_2 + \ldots + J_k$; $J_l + 1 < n$ and $l = 1, \ldots, k$.

Proposition 1
For all $n \geq 2$, we have:

i) $r(n) \geq 2^{n-2} n\, r(n-1)$

ii) $r(n) \geq 2^{\frac{(n-1-J)(n+J-2)}{2}} n(n-1) \ldots (J+2) \times r(J+1)\ \forall\ (1+J) \leq n$

View Proof in Appendix A

We denote by: $n\text{-}1 = J_1 + J_2 + \ldots + J_k$; $J_l + 1 < n$

$J_{l^-} = \min_{1 \leq l \leq k} (J_l)$

$J_{l^+} = \max_{1 \leq l \leq k} (J_l)$

Proposition 2

$\frac{r(n)}{\sum_{l=1}^{k} r(J_l+1)} \geq \rho\left(n, J_{l^-} J_{l^+}, k\right) \gg 1$; Where

$$\rho\left(n, J_{l^-} J_{l^+}, k\right) = \frac{2^{\frac{\left(n-1-(J_{l^+})\right)(n+(J_{l^-})-2)}{2}} n(n-1)\ldots\left((J_{l^+})+2\right)}{k}$$

Proof

According to ii) of Proposition 1, we have

$$r(n) \geq 2^{\frac{(n-1-J_l)(n+J_l-2)}{2}} n(n-1) \ldots (J_l+2) r(J_l+1); l = 1, \ldots, k.$$

Hence,

$$\sum_{l=1}^{k} \mathrm{r(n)} \geq \sum_{l=1}^{k} 2^{\frac{(n-1-J_l)(n+J_l-2)}{2}} n(n-1) \ldots (J_l+2) r(J_l+1)$$

$$\mathrm{k} \times \mathrm{r(n)} \geq \sum_{l=1}^{k} 2^{\frac{\left(n-1-(J_{l^+})\right)(n+(J_{l^-})-2)}{2}} n(n-1) \ldots (J_{l^+}+2) r(J_l+1)$$

Where $2^{\frac{\left(n-1-(J_{l^+})\right)(n+(J_{l^-})-2)}{2}} n(n-1) \ldots (J_{l^+}+2) = \text{constant}$

$$\mathrm{k} \times \mathrm{r(n)} \geq 2^{\frac{\left(n-1-(J_{l^+})\right)(n+(J_{l^-})-2)}{2}} n(n-1) \ldots (J_{l^+}+2) \sum_{l=1}^{k} r(J_l+1)$$

$$\mathrm{r(n)} \geq \frac{2^{\frac{\left(n-1-(J_{l^+})\right)(n+(J_{l^-})-2)}{2}} n(n-1)\ldots\left((J_{l^+})+2\right)}{k} \sum_{l=1}^{k} r(J_l+1)$$

Therefore: $\frac{r(n)}{\sum_{l=1}^{k} r(J_l+1)} \geq \rho\left(n, J_{l^-} J_{l^+}, k\right) \gg 1$,

where $\rho\left(n, J_{l^-} J_{l^+}, k\right) = \frac{2^{\frac{\left(n-1-(J_{l^+})\right)(n+(J_{l^-})-2)}{2}} n(n-1)\ldots\left((J_{l^+})+2\right)}{k}$

Finally, we can conclude that $r(n) >> \sum_{l=1}^{k} r(J_l+1)$, where $n\text{-}1 = J_1 + J_2 + \ldots + J_k$; $J_l + 1 < n$ and $l = 1, \ldots, k$.

3.2 Procedure and Applied Methodologies

3.2.1 Choice of a Global Interest Variable

The aim of such a step is to select a diagnosis variable, or a global interest variable, of the information system to be modelled. This variable could be a status variable, for instance: status of individuals (ill/sound), cars' status (start/not start), customer status (solvent/not solvent) etc. In such cases, the choice is an easy and immediate one. As for those cases in which the choice is not evident, due to the analyst's ignorance of the

studied variables' nature, one might eventually resort to some classical automatic data-exploring methods. For instance, to the principal component analysis in a bid to dismantle, from the first resulting axis, the mostly intervening variable in the information system subject of study.

3.2.2 The Variables' Clustering

The automatic type of clustering is the most frequently used and widespread technique among the data-analysis and data mining descriptive techniques. It is often used when we get a huge amount of data, within which we intend to distinguish some homogeneous subsets suitable for processing and for differential analyses [13].

Actually, there exist two major well-known algorithm classifying families in the literature, namely, the partition methods as well as the ascending hierarchical-clustering ones. The advantage of the ascending-hierarchical methods, as compared to the partitioning one, lies in the fact that they enable to choose, appropriately, the optimum number of clusters. Nevertheless, the partitioning criterion is not global; it exclusively depends on the already-obtained clusters, since two variables placed in different clusters could by no means be compared any more. Contrary to the hierarchical methods, the partitioning algorithms might continuously improve the clusters-quality [13], in addition to the fact that their algorithmic complexities are linear (for the most popular algorithms). Regarding our present work, however, we have chosen to use the K-means algorithm, as it is the most popular and applied in the literature, added to fact that its algorithmic complexity is linear (O(n)) [14]. We also propose to use a hierarchical clustering algorithm along with the bootstrap technique to obtain the optimal number of clusters that will be introduced as entries in the K-means algorithm. To note, the databases that will be applied to test our approach, in the experimentation section, consist of categorical variables, and regarding the performance of clustering we will use the toolbox ClustOfVar with the software R [15]. In particular, we will use the variant K-means for categorical variables [16] and the link-likelihood approach [17] (hierarchical clustering algorithm for categorical variables). To assess the stability of all possible partitions, 2 to p-1 (where p is the total number of variables) clusters from the hierarchical clustering, we will use a feature called "Stability" (also developed in the ClustOfVar toolbox) based on the "bootstrap" technique. The result is a graph which is then a tool to help to select the number of clusters. The user can be choosing the number K of clusters to the heights of the first increase in the stability.

3.2.3 Structure Learning

A structure learning of each cluster's variable with the interest variable, will be undertaken. The ultimate structure would be the n structures obtained from each cluster around the interest variable.

Numerous algorithms have been devised with regards to the learning of BNs structure, noteworthy among which is the algorithm PC [18], Maximum weight spanning tree (MWST) [19], the algorithm K2 [3], Greedy Search (GS) [4] etc. Still, the most frequently used algorithm, according to the specialized literature, remains the algorithm K2. It is characterized by its rapidity, promptness and the stability of constancy of its results. Yet, its major problem remains the initial order required for the entries,

which is very influential on the final results. As a remedy to their problem, the most frequently used solution consists in applying the upstream MWST algorithm [20], to obtain a certain order of nodes useful to be introduced as entries for the K2 algorithm. Less sensitive to the data-base size variation, the MWST algorithm yields a graph quite similar to the original one. Nevertheless, this method runs exclusively through the (very poor) trees space [20]. It, therefore, turns out to us to be the most exclusive effective tool necessary for getting an initial order of nodes very accurate and close to the data, useful to be used in entry with the K2 algorithm.

To note that in our work, we will use the BNT toolbox [23] running on Matlab software (2010 version) to apply the MWST and K2 algorithms for learning structure. We will use also the BNT toolbox to learning parameters and inference.

4 Experimentations Procedures

4.1 Data-Bases

We are going to test our designed approach, firstly, on a car diagnosis data-base dubbed "Car Diagnosis 2". It is made up of eighteen variables (see Table 1), among which is a statue variable called "Car starts", the global interest variable of the information system. The parameters' generating file of this data base is available on the site http://www.norsys.com/downloads/netlib/. According to these parameters, we have been able to generate some 10000 examples, among which thirty two have been left aside for the references' testing phase. Secondly, the model will be applied on a Lymphography diagnosis data-base dubbed "Lymphography". It is made up of nineteen variables (see Table 2), among which is a statute variable called "Diagnosis", the global interest variable of the information system. This lymphography domain has been obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. In this respect, we would like to thank Mr. Zwitter and Mr. Soklic for providing the data. Among the 148 instances of data, thirty two have been left aside for the references' testing phase.

Table 1. "Car diagnosis 2" variables

Variables' names	possible states	Variables names	possible states
AL : Alternator	(Okay, Faulty)	HL: Head lights	(bright, dim, off)
CS : Charging System	(Okay, Faulty)	SP: Spark plugs	(okay, too_wide, fouled)
BA : Battery age	(new, old, very_old)	SQ: Spark Quality	(good, bad, very_bad)
BV: Battery voltage	(strong, weak, dead)	CC: Car cranks	(True, False)
MF: Main fuse	(okay, blown)	TM: Spark timing	(good, bad, very_bad)
DS: Distributor	(Okay, Faulty)	FS: Fuel system	(Okay, Faulty)
PV: Voltage at plug	(strong, weak, none)	AF: Air filter	(clean, dirty)
SM: Starter Motor	(Okay, Faulty)	AS: Air system	(Okay, Faulty)
SS: Starter system	(Okay, Faulty)	ST: Car starts	(True, False)

Table 2. "Lymphography" variables

Variables' names	Possible states
V1: Lymphatics	(normal, arched, deformed, displaced)
V2: Block of affere	(no, yes)
V3: bl. of lymph. C	(no, yes)
V4: bl. of lymph. s	(no, yes)
V5: by pass	(no, yes)
V6: extravasates	(no, yes)
V7: regeneration of	(no, yes)
V8: early up take in	(no, yes)
V9: lym.nodes dimin	(0, 1, 2, 3)
V10: lym.nodes enla	(1, 2, 3, 4)
V11: changes in lym.	(bean, oval, round)
V12: defect in node	(no, lacunar, marginal, lac_central)
V13: changes in node	(no, lacunar, marginal, lac_central)
V14: changes in stru	(no, grainy, draplike, coarse, diluted, reticular, stripped, faint)
V15: special forms	(no, chalices, vesicles)
V16: dislocation	(no, yes)
V17: exclusion	(no, yes)
V18: no. of nodes	(1, 2, 3, 4, 5, 6, 7, 8)
VI: Diagnosis	(normal, metastases, malign_lymph, fibrosis)

4.2 Clustering

Regarding the clustering, we are going to use the stability function (bootstrap approach using the mean of corrected rand criterion) of the toolbox ClustOfVar [16] after the application of an hirarchical ascendant algorithm, in order to estimate, approximately, the number of clusters to be entered in the algorithm K-means.

Using the stability graphics, the optimal number of clusters selected, for "Car diagnosis 2" database, has been equal to three and the clustering result of variables is presented in "Table 3".

Table 3. Clustering results of the "Car diagnosis 2" data base

Cluster 1	Cluster 2	Cluster 3
AL : Alternator	DS: Distributor	FS: Fuel system
CS : Charging System	TM: Spark timing	AF: Air filter
BA : Battery age		AS: Air system
BV: Battery voltage		
MF: Main fuse		
PV: Voltage at plug		
SM: Starter Motor		
SS: Starter system		
HL: Head lights		
SP: Spark plugs		
SQ: Spark Quality		
CC: Car cranks		

Using the stability graphics, the optimal number of clusters selected, for "Lymphography" database, has been equal to two and the variables clustering results is presented in "Table 4" below.

Table 4. Clustering results of the "Lymphography" data base

Cluster 1	Cluster 2
V1, V8, V9, V10 V11, V12, V13, V14, V15, V16, V17, V18	V2, V3, V4 V5, V6, V7

4.3 The Classical Learning Structure Compared to Our New Heuristic

For the "Car diagnosis 2" database, "Figure 1" below depicts the classical structure learning result of the entire variables after applying the K2 algorithm, with as entry, the obtained order reached via the tree resulting from the implementation of the MWST algorithm (to note: we have chosen the interest variable as an initial variable during the application of the MWST algorithm). The execution time has been *3.45* seconds.

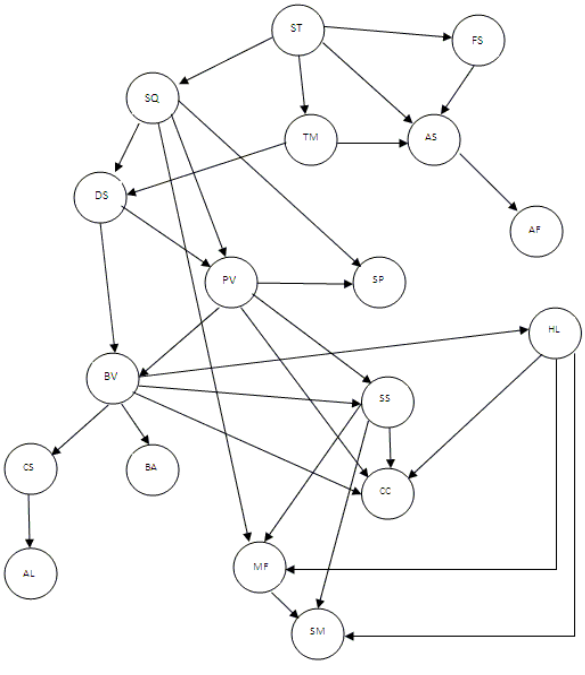


Fig. 1. The classical structure learning result

The Figures 2, 3 and 4, appearing below, depict the structures resulting from the learning structure pertaining to every cluster of variables after applying the K2 algorithm, with, as an entry, the order obtained from the MWST resulting tree (we have selected the interest variable as being the initial variable during the MWST algorithm application to each cluster). The final structure is automatically represented by reassembling the clusters' structures around the interest variable (see Figure 5). The global execution time has been *1.45* seconds (over *1.32* seconds for cluster 1; *0.05* seconds for cluster 2 and *0.09* seconds for cluster 3). The sum of these executions' time (*1.45* seconds) remains significantly inferior to the structure learning of the entire variables simultaneously, which equals *3.45* seconds.

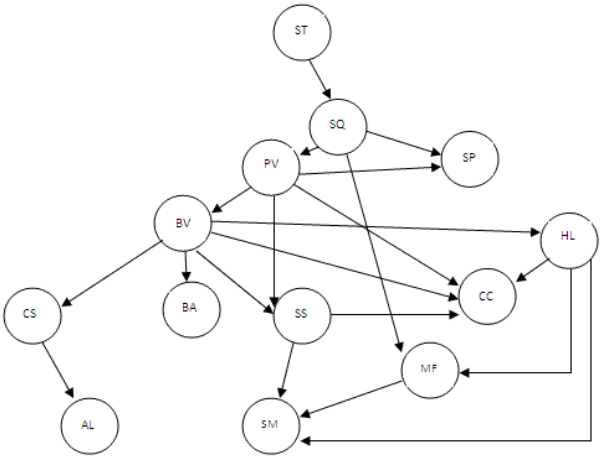


Fig. 2. Cluster 1 structure

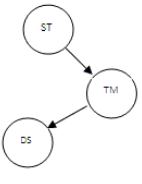


Fig. 3. Cluster 2 structure

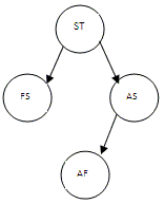


Fig. 4. Cluster 3 structure

For the "Lymphography" database, the same treatment and the same algorithms are applied. The sum of learning structure of "cluster 1" and "cluster 2" executions' time (equal to *1.65* seconds) remains significantly inferior to the structure learning of the entire variables, simultaneously, which equals *2.67* seconds.

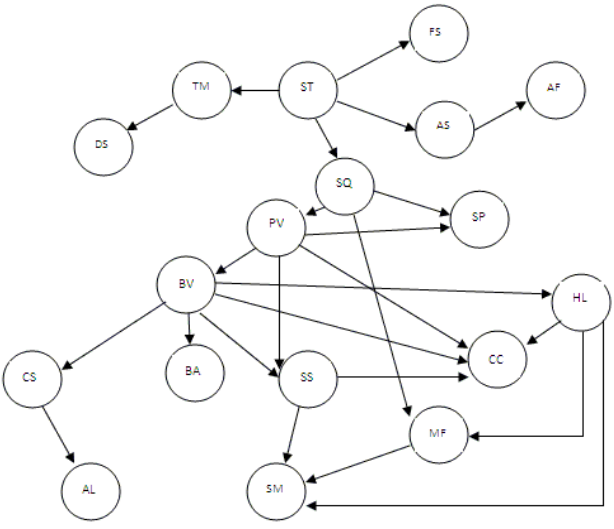


Fig. 5. The ultimate Structure

4.4 Both Attained Structures' Relevant Inferences and Result Comparisons

As our approach favors the preservation of data, principally for the interest variable's sake, we will learn the parameters of the two structures found for each of the databases studied (structure found after learning all the variables simultaneously and structure found after assembling the various structures of the clusters around the interest variables). As for the interest variable, we are going to calculate the probabilities of its different possible corresponding states, bearing in mind the states of the network's other nodes in respect of the two obtained BN structures. Thus, a thirty-two-example database will be used for experimenting the interest variables of both databases. Naturally, the experimentation examples have been excluded during the structures' learning. The differential statistical significance between the obtained probabilities, with respect to both structures, will be measured via the "*Z*" test (comparing the two observed means belonging to two different samples), according to the following formula: $Z = \frac{P1-P2}{\sqrt{variance(P1)+variance(P2)-2\times covariance(P1,P2)}}$ [24]

Hypothesis H_0: the difference between both probabilities is significant *(|Z|>1.96)*.

HypothesisH_1: the difference between both probabilities is non-significant *(|Z|≤1.96)*.

The two tested variables are "Car starts" of "Car Diagnosis 2" database and "Diagnosis" of "Lymphography" database. "Appendix B" contains two graphs showing the variation of the *Z*-test for each variable studied according to its different possible states.

4.5 Discussion

Based on the achieved experimental results, the pairs of probabilities for the variable "diagnosis" of the "Lymphography" database are identical; the preservation of information has been complete (see Appendix B, Figure 7). As for the variable "Car Start" of "Car Diagnosis 2" database, the probabilities pairs are very similar but not identical; the hypothesis H_0 has always been rejected, even with very small Z values, not exceeding the value of $|0.46|$, very distant from the threshold of $|1.96|$, as set by the Z test theory (see Appendix B, Figure 6). It can, therefore, be deduced that the inference results, regarding both of learning structures approaches, are very similar even at eye sight, and without applying any statistical tests to measure the difference's significance. Through our heuristic, we have managed to reduce, considerably, the algorithmic complexity of the BN structure learning without any significant loss of information, especially with regards to the interest variable. The clustering constancy and trustiness plays a determining role in the accuracy of the resulting structure. In fact, the more independent the obtained clusters are, the more the number of inter-cluster edges to be lost would shrink; consequently, the more independent the clusters are, the more negligible the lost information would be.

Throughout the present study, we have, firstly, demonstrated mathematically that the algorithmic complexity of the BN from data-base structure learning decreases dramatically in the cases when the variables' subsets are treated in a separate way. In a second place, a heuristic has been proposed whereby the demonstrated conduct could be exploited by adding a solution serving to reassemble the sub-sets' structures into a single structure framework. This solution has been based on the implementation of the information system's interest variable as a linking variable among the subsets' different structures. Through our proper experimentation procedure, we have proved that by implementing this undertaking, we can be immune against the information loss problem while achieving a considerable gain in terms of execution time. Our original solution has been improved; firstly because no criterion has been defined for the applicability of our approach on a certain database (possibility of having clusters sufficiently independent to avoid losing information). Secondly, the method applied for determining the optimal number of clusters is known to be greedy in computational complexity (in the order of $O(n^3)$). So, a heuristic, less complicated yet effective would be among our aim in future research. Inversely, however, with the help of our newly-devised concept, new large-scale horizons have been opened, paving the way for other more global solutions, taking advantage of the fact that the possible number of DAG decreases incredibly by treating the variables and subsets during the BN structure learning from data-base.

5 Conclusion

Within the scope of the present work, we have set up a new well-defined approach for the BN structure learning from data-base, so useful that it can be jointly applied with the already existing algorithms and underway heuristics. As a first step, we have demonstrated, mathematically, that the BN structures' possible space decreases,

dramatically, by subdividing the relevant variables into clusters before processing the BN structure learning corresponding to each cluster apart. In the second step, a specially-devised heuristic has been proposed with the aim of joining each cluster's different structures. Actually, through a specially-conducted experimentation administered over tow data-bases, we have proved that loss in data turns out to be so negligible that it does not affect the extracted BNs stemming results during the inference stage, while saving a great deal of execution time.

In a potential future research, we reckon to make a serious attempt to investigate other possible alternatives, useful and fit to exploit the considerable reduction of algorithmic complexity during the BN structure learning by examining and treating variables' sub-sets, developing some structure-retrieving oriented heuristics, encompassing the already achieved sub-structures, a framework that would be the closest possible to the discovered structure, while simultaneously treating the whole set of variables in their entirety.

References

1. Nefian, V.: Learning SNP using embedded Bayesian Networks. In: IEEE Computational Systems Bioinformatics Conference (2006)
2. Cooper, G., Hersovits, E.: A Bayesian method for the induction of probabilistic networks from data. Machine learning 9, 309–347 (1992)
3. Neapolitan, R.E.: Learning Bayesian Networks. Prentice Hall, Newyork (2003)
4. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search, 2nd edn. The MIT Press (2000)
5. Shulin, Y., Chang, K.: Comparison of Score Metrics for Bayesian Network Learning. IEEE Transactions on Systems, Man and Cybernetics-part A: Systems and Humans 32(3), 419–428 (2002)
6. Bouchaala, L., Masmoudi, A., Gargouri, F., Rebai, A.: Improving algorithm for structure learning in Bayesian Networks using a new implicit score. Expert Systems with Application 37, 5470–5475 (2010)
7. Mourad, R., Sinoquet, C., Leray, P.: A hierarchical Bayesian Network approach for linkage disequilibrium modelling and data dimensionality reduction prior to genome-wide association studies. BMC Bioinformatics 16 (2011)
8. Zhang, Y., Ji, L.: Clustering of SNPs by structural EM algorithm. In: Proceeding of International Joint Conference on Bioinformatics. Systems Biology and Intelligent Computing, pp. 147–150 (2009)
9. Hwang, K.-B., Kim, B.-H., Zhang, B.-T.: Learning Hierarchical Bayesian Networks for Large-Scale Data Analysis. In: King, I., Wang, J., Chan, L.-W., Wang, D. (eds.) ICONIP 2006. LNCS, vol. 4232, pp. 670–679. Springer, Heidelberg (2006)
10. Mourad, R., Sinoquet, C., Leray, P.: Learning hierarchical Bayesian Networks for genome-wide association studies. In: 19th International Conference on Computational Statistics, COMPSTAT, pp. 549–556 (2010)
11. Judea, P., Tom, V.: A theory of inferred causation. In: Allen, J., Fikes, R., Sandewall, E. (eds.) KR 1991, Principles of Knowledge Representation and Reasoning, pp. 441–452 (1991)
12. Robinson, R.W.: Counting unlabeled acyclic digraphs. Combinatorial Mathematics 622, 28–43 (1977)

13. Tufféry, S.: Data mining et statistique décisionnelle: l'intelligence des données. Editions TECHNIP (2010)
14. Jain, A.K.: Data clustering: 50 years beyond K-means. Pattern Recognition Letters 31, 651–666 (2010)
15. Chavent, M., Kuentz, V., Liquet, B., Saracco, J.: ClustOfVar: an R package for the clustering of variables. In: The R User Conference, University of Warwick Coventry UK (2011)
16. Chavent, M., Kuentz, V., Saracco, J.: A partitioning method for the clustering of categorical variables. In: Locarek-Junge, H., Weihs, C. (eds.) Classification as a Tool for Research, International Federation of Classification Societies Conference. Springer (2009)
17. Lerman, I.C.: Likelihood linkage analysis (LLA) classification method: An example treated by hand. Biochimie. 75(5), 379–397 (1993)
18. Spirtes, P., Glymour, C., Scheines, R.: Causation prediction and search (1993)
19. Chow, C., Liu, C.: Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory 14(3), 462–467 (1968)
20. Francois, O., Leray, P.: Evaluation d'algorithmes d'apprentissage de structure pour les réseaux bayésiens. In: 14ème Congrès Francophone Reconnaissance des Formes et Intelligence Artificielle, pp. 1453–1460 (2004)
21. Heckerman, D., Geiger, D., Chickering, M.: Learning Bayesian Networks: The combination of knowledge and statistical data. In: 10th Conference on Uncertainty in Artificial Intelligence, pp. 293–301 (1994)
22. Kruskal, J.: On the shortest spanning subtree of a graph and traveling salesman problem. The American Mathematical Society 7, 48–50 (1956)
23. Murphy, K.: The BayesNet Toolbox for Matlab. Computing Science and Statistics: Proceedings of Interface, vol. 33, `http://www.ai.mit.edu/~murphyk/Software/BNT/bnt.html`
24. Sprinthall, R.C.: Basic Statistical Analysis, 7th edn. (2003)

Appendix A

Proof of Proposition 1 (i)

For all $n \geq 2$, we have:

i) $r(n) \geq 2^{n-2} n\, r(n-1)$

Proof by induction on n

For $n=2$; $r(2)= 3 \geq 2$ is verified

Tel $n \in IN$, we assume that $\forall\, i \leq n;\ r(i) \geq 2^{i-2}\, i\ r(i-1)$. (1)

By applying Robinson formula, we have:

$r(n+1) = \sum_{i=1}^{n} (-1)^{i+1} 2^{i(n+1-i)} \binom{n+1}{i} r(n+1-i)$

We set $Vi = 2^{i(n+1-i)} \binom{n+1}{i} r(n+1-i)$

We will firstly prove that $(V_i)_{1 \leq i \leq n+1}$ is decreasing

$$\frac{Vi}{Vi+1} = \frac{2^{i(n+1-i)}\binom{n+1}{i} r(n+1-i)}{2^{(i+1)(n+1-i-1)}\binom{n+1}{i+1} r(n-i)}$$

$$= \frac{2^{i}\ \frac{1}{(n+1-i)!}\ r(n+1-i)}{2^{n-i}\ \frac{1}{(i+1)(n-i)!}\ r(n-i)}$$

By using (1),

$$\frac{Vi}{Vi+1} \geq \frac{2^{2i-n}(i+1)2^{n+1-i-2}\,(n+1-i) r(n-i)}{(n+1-i)}$$

Which imply, $\frac{Vi}{Vi+1} \geq 2^{i-1}(i+1) > 1$. This means that $(V_i)_{1\leq i\leq n+1}$ is decreasing.
Secondly we will prove that $r(n+1) - 2^{n-1}(n+1)\, r(n) \geq 0$
Observe that,
$2^{n-1}(n+1)\, r(n) - 2^{2(n-1)}\binom{n+1}{2} r(n-1) \geq$
$2^{n-1}(n+1)2^{n-2}n\, r(n-1) - 2^{2(n-1)}\frac{(n+1)n}{2} r(n-1) \geq$
$2^{2n-3}(n+1)n\, r(n-1) - 2^{2n-3}(n+1)n\, r(n-1) = 0$
So, $2^{n-1}(n+1)\, r(n) + 2^{2(n-1)}\binom{n+1}{2} r(n-1) \leq 2 \times 2^{n-1}(n+1)\, r(n) = 2^n(n+1)\, r(n)$
=> The sum of the tow first elements of $r(n+1)$ minus $2^{n-1}(n+1)\, r(n)$ is positive
=> $r(n+1) - 2^{n-1}(n+1) r(n) = positive\ element + S$; where $S = \sum_{i=3}^{n+1}(-1)^{i+1} Vi$
Thirdly we will prove that $S \geq 0$

Case 1: n is impair: $n=2p-1;\ p \in IN^*$
$S = \sum_{i=3}^{n+1}(-1)^{i+1} Vi$
$= \sum_{i=3}^{2p}(-1)^{i+1} Vi$
$= \sum_{1\leq j\leq p}(-1)^{2j+1} V_{2j} + \sum_{1\leq j\leq p-1}(-1)^{2j+2} V_{2j+1}$
(Where in the first sum $i=2j$ and in the second $i=2j+1$)
$= \sum_{1\leq j\leq p-1} V_{2j+1} - \sum_{2\leq j\leq p} V_{2j}$
$= \sum_{1\leq j\leq p-1} V_{2j+1} - \sum_{2\leq j\leq p-1} V_{2j+2} = \sum_{j=1}^{p-1}(V_{2j+1} - V_{2j+2})$. Since $(V_i)_{1\leq i\leq n+1}$ is decreasing we can conclude that $\sum_{j=1}^{p-1}(V_{2j+1} - V_{2j+2}) \geq 0$

Case 2: n is pair: $n=2p$
$S = \sum_{i=3}^{n+1}(-1)^{i+1} Vi$
$= \sum_{i=3}^{2p+1}(-1)^{i+1} Vi$
$= \sum_{1\leq j\leq p}(-1)^{2j+1} V_{2j} + \sum_{2\leq j\leq p+1}(-1)^{2j} V_{2j-1}$
(Where in the first sum $i=2j$ and in the second $i=2j-1$)
$= \sum_{2\leq j\leq p}(V_{2j-1} - V_{2j}) + V_{2p+1}$. Since (V_i) is decreasing and $V_{2p+1} \geq 0$ then $\sum_{2\leq j\leq p}(V_{2j-1} - V_{2j}) + V_{2p+1} \geq 0$
Therefore $S \geq 0$, then
$r(n+1) - 2^{n-1}(n+1) r(n) \geq 0$, then
$r(n+1) \geq 2^{n-1}(n+1) r(n)$; Which proves the proposition.

Proof of Proposition 1 (ii)

ii) $r(n) \geq 2^{\frac{(n-1-J)(n+J-2)}{2}} n(n-1) \dots (J+2) \times r(J+1)\ \forall\ (1+J) \leq n$

Proof:
By using i) of proposition 1, we have the desired result.
$r(n) \geq 2^{n-2} \times 2^{n-3} \times \dots \times 2^J n(n-1) \dots (J+2) \times r(J+1)$
$r(n) \geq 2^{\frac{(n-1-J)(n+J-2)}{2}} n(n-1) \dots (J+2) \times r(J+1)$
where $(n-2)+(n-3)+\dots+J = \frac{(n-1-J)}{2}(J+n-2)$
We can conclude that our proposition 1 (ii) is confirmed.

Appendix B

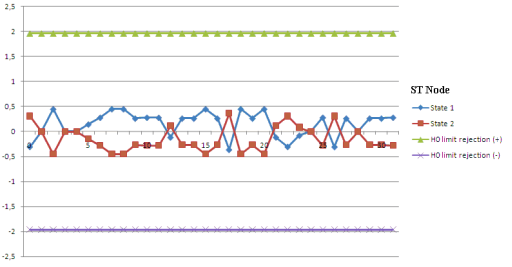


Fig. 6. Z-test variation for the "Car starts" variable ("Car Diagnosis 2" database)

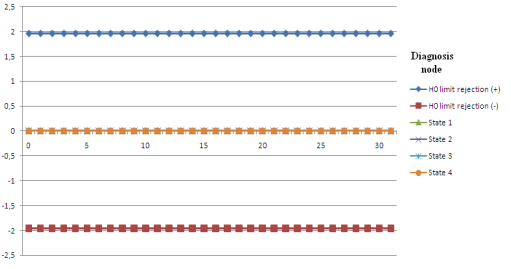


Fig. 7. Z-test variation for the "Diagnosis" variable ("Lymphography" database)

Discriminant Subspace Learning Based on Support Vectors Machines

Nikolaos Pitelis[1] and Anastasios Tefas[2]

[1] School of Electronic Engineering and Computer Science,
Queen Mary, University of London, UK
nikolaos.pitelis@eecs.qmul.ac.uk
[2] Department of Informatics, Aristotle University of Thessaloniki, Greece
tefas@aiia.csd.auth.gr

Abstract. A new method for dimensionality reduction and feature extraction based on Support Vector Machines and minimization of the within-class data dispersion is proposed. An iterative procedure is proposed that successively applies Support Vector Machines on perpendicular subspaces using the deflation transformation in such a way that the within-class variance is minimized. The proposed approach is proved to be a successive SVM using deflation kernels. The normal vectors of the successive hyperplanes contain discriminant information and they can be used as projection vectors for feature extraction and dimensionality reduction of the data. Experiments on various datasets are conducted in order to highlight the superior performance of the proposed algorithm.

1 Introduction

In pattern recognition and machine learning problems with high-dimensional data have always been difficult to cope with. That is, the so-called "curse of dimensionality", which constitutes motivation for the development of dimensionality reduction methods. Simple classification algorithms which are very commonly used in a variety of disciplines, like *k-Nearest Neighbor* (KNN) [1] or *Nearest Centroid* (NC) [2], favor greatly when they have to treat the same problem in a lower-dimensional space, especially when it is redundant.

The benefits lie in reducing computational complexity, since the size of the problem is reduced, and improving classification accuracy. The first gives the possibility to deal with more complex problems that cannot be treated in their original form. In order for the latter to be succeeded, the dimensionality reduction has to take place in such a way that will augment discriminant information and remove information that does not contribute discriminability, e.g noise.

A closely related term to dimensionality reduction is feature extraction, which entails the transformation of the data from the high-dimensional space to the lower-dimensional one. This transformation can be either linear or non-linear and although linear transformations have a more solid mathematical background, non-linear transformations, which are usually extensions of previously proposed linear ones, are usually more powerful. These non-linear generalizations are usually achieved using the *kernel trick* [3], which gives us the opportunity to compute

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 198–212, 2012.

only the dot products of the inpute patterns, rather to explicitly compute the mapping, i.e. their projection onto a very high-dimensional space.

Feature extraction can also be used for visualization tasks so as to get better understanding and an overview of a problem. In the framework of this paper we are interested both in visualization and classification tasks. In the following we shortly describe the most commonly used dimensionality reduction techniques that are related to our proposed method.

1.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) [4], also known as Karhunen-Loeve transformation, was fisrt developed by Pearson [5] (1901) and Hotelling [6] (1933). It is one of the most widely used dimensionality reduction technique in problems as data compression and clustering, pattern recognition and visualization. The main idea is to reduce the dimensionality of a data population trying to keep its spatial characteristics. This is achieved with a linear transformation to the space of principal components, which are ordered in such a way that the first few retain most of the data variation. The principal components are obtained by performing eigenanalysis of the data covariance matrix.

More specifically, if $\boldsymbol{A}_k$ is the matrix of the k eigenvectors that correspond to the k largest eigenvalues of the covariance matrix and $\boldsymbol{X}$ is the initial data matrix, then the transformed data of dimensionality k is given by $\boldsymbol{Z} = \boldsymbol{A}_k\boldsymbol{X}$. PCA has been generalized into kernel-PCA [7] using the kernel trick. Since PCA is an unsupervised learning method, i.e. class label information is not taken into account, it is not always suitable for classification tasks.

1.2 Linear Discriminant Analysis (LDA)

On the contrary to PCA, *Linear Discriminant Analysis* (LDA), [8], also known as *Fisher's Discriminant Analysis* (FDA or FLDA), is a supervised learning technique, which exploits the class label information in order to maximize the classes discriminality in the extracted space. This is achieved by maximizing Fisher's discriminant ratio, that is, the ratio of between-class variance to within-class variance. For a training set of d-dimensional samples $\boldsymbol{x}_i, i = 1, \ldots, N$ that belong to two classes these notions are expressed by the following quantities

$$\begin{aligned} \boldsymbol{S}_i &= \sum_{\boldsymbol{x}\in\omega_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T, \\ \boldsymbol{S}_W &= \boldsymbol{S}_1 + \boldsymbol{S}_2 \quad \text{and} \\ \boldsymbol{S}_B &= (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T. \end{aligned} \tag{1}$$

By $\boldsymbol{\mu}_i$ we denote the mean value of class ω_i. We call $\boldsymbol{S}_W$ *within-class scatter matrix* and $\boldsymbol{S}_B$ *between-class scatter matrix*. The quantity that LDA seeks to maximize is defined as

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{S}_W\boldsymbol{w}}, \tag{2}$$

where $\boldsymbol{w}$ is the projection vector that transforms the data to the one-dimensional subspace. If the number of classes is more than two, then the reduced dimensionality can be at most equal to the number of classes minus one. The performance of LDA is optimal provided that the data distributions are normal for all classes with the same covariance matrix. LDA was also extended for the non-linear case in Kernel-FDA [9], similarly to PCA, using the kernel trick.

1.3 Margin Maximizing Discriminant Analysis (MMDA)

Margin Maximizing Discriminant Analysis (MMDA) [10] investigated the possibility of projecting the input data onto the normal of a hyperplane that separates two classes in a binary problem. This hyperplane should provide good generalization for future data and make no assumptions regarding the distribution of the input patterns. The authors proposed a deflation approach to be able to perform this process in subsequent orthogonal subspaces, by projecting onto the space spanned by the normal of such a margin maximizing hyperplane. The first hyperplane is obtained solving the *Maximum Margin Separation* (MMS) problem, which is expressed as a quadratic programming problem:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}||\boldsymbol{w}||_2^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{subject to} \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n. \end{aligned} \tag{3}$$

The resulting weighting vector of the hyperplane is normalized and used for projecting the data by $\boldsymbol{x}_i' = \boldsymbol{x}_i - (\boldsymbol{w}^T\boldsymbol{x}_i)\boldsymbol{w}$. Then, problem 3 is solved again for the projected data.

1.4 Proposed Approach

In this paper we propose a novel supervised learning technique that seeks to exploit *Support Vector Machines* (SVMs) in a dimensionality reduction scheme. More specifically we intend to use the discriminant information contained in the resulting hyperplane of SVMs to perform feature extraction and the corresponding normal vector of the hyperplane can be used as projection vector. Thus, the first step of the proposed method is the standard SVM optimization that generates the first dimension/feature. In order to be able to extract additional discriminant information we adopt a deflation procedure similar to MMDA. On the same time, inspired by the maximization of Fisher's discriminant ratio, we desire to minimize the within-class variance similarly to [11] and [12]. This results to the definition of a new optimization problem incorporating both the deflation procedure and the within-class variance minimization. This approach can be regarded as a modification of the standard SVMs optimization, employing a deflation kernel.

The novelty of our work lies in three different aspects. The first is the idea of combining SVMs for maximizing the between-class margin and Fisher's discriminant ratio for minimizing the within class variance in one dimensionality

reduction technique. The second is the iterative generation of successive orthonormal projections onto deflated subspaces, according to this criterion, for feature extraction. And the third is the incorporation of the within-class variance minimization and the deflation procedure in the SVMs optimization, using the kernel trick.

The manuscript is organized as follows: The proposed approach is described in Sec. 2. In Sec. 2.1, we discuss in detail the deflation procedure that we adopt and in Sec. 2.2 we show how the deflation of the within-class scatter matrix can be included in the same procedure. The modified optimization problem is presented in Sec. 2.3 and in Sec. 2.4 we show how this problem can be efficiently solved using the kernel trick. The way we perform the feature extraction and the final form of the algorithm are presented in Sec.2.5. In Sec. 3.1 we demonstrate the visualization capability of our method and in Sec. 3.2 we present the experimental results for classification tasks. Finally, conclusions are drawn in Sec. 4.

2 Definition and Derivation of the Problem

In the proposed approach, our goal is to use the information regarding the distribution of the data in space, which is contained in the resulting hyperplane of a classification task with SVMs minimizing in parallel the within-class variance. Moreover, we want to do that in an iterative way so as each iteration of the procedure will provide us with a new feature, which will contain additional discriminant information for our data with respect to the preceding steps. In order to achieve that we need to apply the SVMs in successive subspaces, which are pairwise perpendicular. The proposed method is called *Within Support Vector Discriminant Analysis* (WSVDA) and the algorithm can be overseen in Table 2.

Table 1. The main steps of the iterative procedure of WSVDA

1: compute the within scatter matrix $\boldsymbol{S}_W$ for the data
2: solve SVMs for the data minimizing $\boldsymbol{S}_W$
3: compute weighting vector $\boldsymbol{w}$
4: compute projection matrix $\boldsymbol{P}$ using the normalized weighting vector of previous step
5: deflate the data along the direction of $\boldsymbol{w}$
6: iterate from step 1 to step 5 for as many times as the desired reduced dimensionality
7: use the normalized weighting vectors $\boldsymbol{w}$ for feature extraction

2.1 Deflation Procedure

Let us present this idea with a simple example. If we think of a three-dimensional example of a binary problem, the resulting hyperplane (actually a plane) of linear SVMs would be as depicted in Fig. 1. The projection of the data onto the hyperplane is additionally a transformation to a space perpendicular to the initial one. Consequently if we apply SVMs to these transformed data, projected

onto the hyperplane, the resulting hyperplane (actually a line) will be perpendicular to the initial one. That means that the two hyperplanes contain exclusive information regarding the data distribution. The orthogonality property stands for the corresponding normal vectors of the hyperplanes, which can be used for the feature extraction.

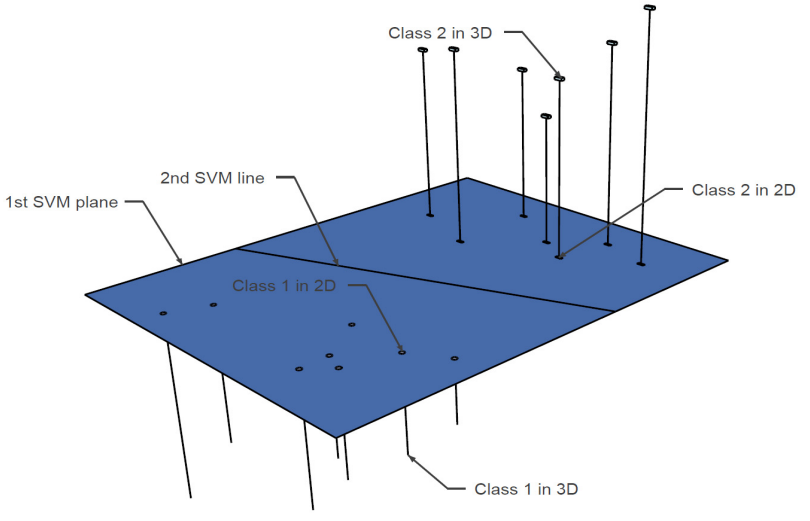


Fig. 1. The resulting hyperplanes of linear SVMs for a three-dimensional example of a binary problem. The three-dimensional data are projected onto the plane, which is the decision surface when the three-dimensional data are the input to the SVMs. The deflated two-dimensional data are the input to the SVMs in the second iteration of the method, resulting to a separating line. Best viewed in color.

Suppose the training set with finite number of elements $\boldsymbol{x}_i, i = 1, \ldots, N$, of dimensionality d, which can be separated into two different classes ω_1 and ω_2. The corresponding labels for these training samples are denoted by y_i with a value equal to 1, if $\boldsymbol{x}_i \in \omega_1$ or -1 if $\boldsymbol{x}_i \in \omega_2$. We also use the notation $\boldsymbol{X}$ for the data matrix, which contains the vectors $\boldsymbol{x}_i$ in its columns, i.e. $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$.

In order to project the data onto the successive hyperplanes we use a deflation transformation algorithm similar to the one in [13], which is used for deflating the data in the space of principal components. Similarly, if $\boldsymbol{w}^k$ is the normal vector of the hyperplane in iteration k of the procedure, then $\boldsymbol{P}_{\boldsymbol{w}^k} = \boldsymbol{I}_{d\times d} - \boldsymbol{w}^k \boldsymbol{w}^{kT}$, where $\boldsymbol{P}_{\boldsymbol{w}^k}$ is the projection matrix along the direction of vector $\boldsymbol{w}^k$ and $\boldsymbol{I}_{d\times d}$ is the identity matrix of dimension d. It is important to mention here, that the weighting vector $\boldsymbol{w}^k$, which is the result of SVMs in our algorithm, has to be normalized before used for the deflation process. Consequently, the data matrix $\boldsymbol{X}$ can be deflated along the direction of $\boldsymbol{w}^k$, that is, projected onto the hyperplane of iteration k, by multiplying it with the corresponding projection matrix $\boldsymbol{P}_{\boldsymbol{w}^k}$, $\boldsymbol{X}^k = \boldsymbol{P}_{\boldsymbol{w}^k} \boldsymbol{X}$.

Since in each iteration we transform the data to a subspace having removed a dimension, the deflation should be done on all the directions of the normal vectors of the previously computed hyperplanes. This multiple deflation can be done in a successive way: $\boldsymbol{X}^1 = \boldsymbol{P}_{\boldsymbol{w}^1}\boldsymbol{X}, \boldsymbol{X}^2 = \boldsymbol{P}_{\boldsymbol{w}^2}\boldsymbol{X}^1, \ldots, \boldsymbol{X}^k = \boldsymbol{P}_{\boldsymbol{w}^k}\boldsymbol{X}^{k-1}$, but it can also be applied on the initial data matrix using the product of all the projection matrices of the previous steps: $\boldsymbol{P}^k = \boldsymbol{P}_{\boldsymbol{w}^1}\boldsymbol{P}_{\boldsymbol{w}^2}\ldots\boldsymbol{P}_{\boldsymbol{w}^k}$. In this way, the matrix $\boldsymbol{X}^k = \boldsymbol{P}^k\boldsymbol{X}$ has been simultaneously deflated along the multiple directions of the normal vectors.

If we express the product of the successive projection matrices using the weighting normal vectors we have

$$\boldsymbol{P}^k = (\boldsymbol{I}_{d\times d} - \boldsymbol{w}^1\boldsymbol{w}^{1T})(\boldsymbol{I}_{d\times d} - \boldsymbol{w}^2\boldsymbol{w}^{2T})\ldots(\boldsymbol{I}_{d\times d} - \boldsymbol{w}^k\boldsymbol{w}^{kT}), \tag{4}$$

it is shown that the order of multiplication has no effect to the final result and because of the orthogonality property we conclude to

$$\boldsymbol{P}^k = \boldsymbol{I}_{d\times d} - \sum_{i=1}^{k} \boldsymbol{w}^k\boldsymbol{w}^{kT}. \tag{5}$$

However, for the implementation task, the first form proves to be numerically more stable. Finally, as a result of the symmetry of all the projection matrices, they are equivalent to their respective transposed matrices, e.g. $\boldsymbol{P}^k = \boldsymbol{P}^{k^T}$.

2.2 Within-Class Variance Deflation

We have already discussed in the previous section what is the input data, i.e. the deflated data to the successive SVMs, but we also need to provide them with the within scatter matrix of the deflated data. In order to avoid the computation of this matrix for the deflated data in each iteration we investigate the possibility of 'projecting' the within scatter matrix onto the subspace of each iteration. Indeed,

$$\begin{aligned}
\boldsymbol{S}_W^k &= \boldsymbol{S}_1^k + \boldsymbol{S}_2^k \\
&= \sum_{\boldsymbol{x}_i^k\in\omega_1} (\boldsymbol{x}_i^k - \boldsymbol{\mu}_1^k)(\boldsymbol{x}_i^k - \boldsymbol{\mu}_1^k)^T + \sum_{\boldsymbol{x}_j^k\in\omega_2} (\boldsymbol{x}_j^k - \boldsymbol{\mu}_2^k)(\boldsymbol{x}_j^k - \boldsymbol{\mu}_2^k)^T \\
&= \sum_{\boldsymbol{x}_i\in\omega_1} \boldsymbol{P}^k(\boldsymbol{x}_i - \boldsymbol{\mu}_1)(\boldsymbol{x}_i - \boldsymbol{\mu}_1)^T\boldsymbol{P}^{kT} + \sum_{\boldsymbol{x}_j\in\omega_2} \boldsymbol{P}^k(\boldsymbol{x}_j - \boldsymbol{\mu}_2)(\boldsymbol{x}_j - \boldsymbol{\mu}_2)^T\boldsymbol{P}^{kT} \\
&= \boldsymbol{P}^k \sum_{\boldsymbol{x}_i\in\omega_1} (\boldsymbol{x}_i - \boldsymbol{\mu}_1)(\boldsymbol{x}_i - \boldsymbol{\mu}_1)^T\boldsymbol{P}^k + \boldsymbol{P}^k \sum_{\boldsymbol{x}_j\in\omega_2} (\boldsymbol{x}_j - \boldsymbol{\mu}_2)(\boldsymbol{x}_j - \boldsymbol{\mu}_2)^T\boldsymbol{P}^k \\
&= \boldsymbol{P}^k(\boldsymbol{S}_1 + \boldsymbol{S}_2)\boldsymbol{P}^k \\
\boldsymbol{S}_W^k &= \boldsymbol{P}^k\boldsymbol{S}_W\boldsymbol{P}^k,
\end{aligned} \tag{6}$$

where $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ are the within-class scatter matrices for class ω_1 and ω_2 respectively. Similarly, $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are the corresponding mean values.

2.3 Deflated Within-Class Support Vector Machines

According to the aforementioned we can modify the standard SVMs [15] optimization problem and define a new one, which simultaneously will maximize the margin and minimize the within-class variance in deflated subspaces. For the separable case [14] would be expressed as

$$\text{minimize} \quad \boldsymbol{w}^T \boldsymbol{P} \boldsymbol{S}_W \boldsymbol{P} \boldsymbol{w}, \quad \boldsymbol{w}^T \boldsymbol{P} \boldsymbol{S}_W \boldsymbol{P} \boldsymbol{w} > \boldsymbol{0} \tag{7}$$

subject to the separability constraints

$$y_i(\boldsymbol{w}^T \boldsymbol{P} \boldsymbol{x}_i + b) \geq 1, \quad i = 1, \ldots, N. \tag{8}$$

The solution to this problem is given by the saddle point of the Lagrangian

$$\boldsymbol{\mathcal{L}}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \boldsymbol{w}^T \boldsymbol{P} \boldsymbol{S}_W \boldsymbol{P} \boldsymbol{w} - \sum_{i=1}^{N} \alpha_i [y_i(\boldsymbol{w}^T \boldsymbol{P} \boldsymbol{x}_i - b) - 1], \tag{9}$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots \alpha_N]^T$ is the vector of Lagrange multipliers. The Karush-Kuhn-Tucker (KKT) conditions [16] imply that for the saddle point

$$\begin{aligned}
\nabla_{\boldsymbol{w}} \boldsymbol{\mathcal{L}}(\boldsymbol{w}_o, b_o, \boldsymbol{\alpha}_o) = \boldsymbol{0} \Leftrightarrow \boldsymbol{P} \boldsymbol{S}_W \boldsymbol{P} \boldsymbol{w}_o &= \frac{1}{2} \sum_{i=1}^{N} \alpha_{i,o} y_i \boldsymbol{P} \boldsymbol{x}_i \\
\frac{\partial}{\partial b} \boldsymbol{\mathcal{L}}(\boldsymbol{w}_o, b_o, \boldsymbol{\alpha}_o) = 0 \Leftrightarrow \sum_{i=1}^{N} \alpha_{i,o} y_i &= 0 \\
y_i(\boldsymbol{w}_o^T \boldsymbol{P} \boldsymbol{x}_i - b_o) - 1 \geq 0, \quad i &= 1, \ldots, N \\
\alpha_{i,o} \geq 0, \quad i &= 1, \ldots, N \\
\alpha_{i,o}[y_i(\boldsymbol{w}_o^T \boldsymbol{P} \boldsymbol{x}_i - b_o) - 1] \geq 0, \quad i &= 1, \ldots, N,
\end{aligned} \tag{10}$$

where subscript o denotes the optimal solution.

The KKT conditions show that the weighting vector is a linear combination of the support vectors in the training set multiplied by the inverse of the 'projection' of matrix $\boldsymbol{S}_W$, that is $\boldsymbol{P} \boldsymbol{S}_W \boldsymbol{P}$. More specifically the optimal weighting vector normal to the separating hyperplane is given by

$$\boldsymbol{P} \boldsymbol{S}_W \boldsymbol{P} \boldsymbol{w}_o = \frac{1}{2} \sum_{i=1}^{N} \alpha_{i,o} y_i \boldsymbol{P} \boldsymbol{x}_i \Leftrightarrow \boldsymbol{w}_o = \frac{1}{2} (\boldsymbol{P} \boldsymbol{S}_W \boldsymbol{P})^{-1} \sum_{i=1}^{N} \alpha_{i,o} y_i \boldsymbol{P} \boldsymbol{x}_i. \tag{11}$$

By replacing (11) into (9) and using the KKT conditions, we obtain the Wolfe-dual problem

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{4} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{P} (\boldsymbol{P} \boldsymbol{S}_W \boldsymbol{P})^{-1} \boldsymbol{P} \boldsymbol{x}_j, \tag{12}$$

which is equivalent to the optimization problem

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{H}\boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\alpha} \\ &\text{subject to} \quad \alpha_i \geq 0, i = 1, \ldots, N \quad \text{and} \quad \sum_{i=1}^{N} \alpha_i y_i = 0, \end{aligned} \tag{13}$$

where $\boldsymbol{H}_{ij} = \frac{1}{2} y_i y_j \boldsymbol{x}_i^T \boldsymbol{P}(\boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P})^{-1}\boldsymbol{P}\boldsymbol{x}_j$ is the ijth element of the Hessian matrix.

The corresponding separating hyperplane is defined by

$$\begin{aligned} g(\boldsymbol{x}) &= sgn(\boldsymbol{w}^T\boldsymbol{P}\boldsymbol{x} + b) \\ &= sgn\left(\frac{1}{2}\sum_{i=1}^{N} \alpha_{i,o}\left(\boldsymbol{x}_i^T\boldsymbol{P}(\boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P})^{-1}\boldsymbol{P}\boldsymbol{x}\right) + b\right), \end{aligned} \tag{14}$$

where $b_o = \frac{1}{2}\boldsymbol{w}_o^T\boldsymbol{P}(\boldsymbol{x}_i + \boldsymbol{x}_j)$ for any pair of support vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ such that $y_i = 1$ and $y_j = -1$.

In the non-separable case [15], we relax the separability constraints (8) by introducing non-negative slack variables ξ_i, $i = 1, \ldots, N$. The new optimization problem is expressed as

$$\text{minimize} \quad \boldsymbol{w}^T\boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P}\boldsymbol{w} + C\sum_{i=1}^{N}\xi_i, \quad \boldsymbol{w}^T\boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P}\boldsymbol{w} > \mathbf{0} \tag{15}$$

subject to the separability constraints

$$y_i(\boldsymbol{w}^T\boldsymbol{P}\boldsymbol{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, N, \tag{16}$$

where by C we denote the cost of violating the constraints, i.e. the cost of misclassification.

The solution to this problem is given by the saddle point of the Lagrangian

$$\boldsymbol{\mathcal{L}}(\boldsymbol{w}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\xi}) = \boldsymbol{w}^T\boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P}\boldsymbol{w} + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[y_i(\boldsymbol{w}^T\boldsymbol{P}\boldsymbol{x}_i - b) - 1 + \xi_i] - \sum_{i=1}^{N}\beta_i\xi_i, \tag{17}$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots \alpha_N]^T$ and $\boldsymbol{\beta} = [\beta_1, \ldots \beta_N]^T$ are the vectors of Lagrange multipliers. The modified Karush-Kuhn-Tucker (KKT) conditions [16] for the non-separable case imply that for the saddle point

$$
\begin{aligned}
\nabla_{\boldsymbol{w}} \boldsymbol{\mathcal{L}}(\boldsymbol{w}_o, b_o, \boldsymbol{\alpha_o}, \boldsymbol{\beta_o}, \boldsymbol{\xi_o}) = \mathbf{0} &\Leftrightarrow \boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P}\boldsymbol{w}_o = \frac{1}{2}\sum_{i=1}^{N} \alpha_{i,o} y_i \boldsymbol{P}\boldsymbol{x}_i \\
\frac{\partial}{\partial b}\boldsymbol{\mathcal{L}}(\boldsymbol{w}_o, b_o, \boldsymbol{\alpha}_o, \boldsymbol{\beta}_o, \boldsymbol{\xi}_o) = 0 &\Leftrightarrow \sum_{i=1}^{N} \alpha_{i,o} y_i = 0 \\
\frac{\partial}{\partial \xi_i}\boldsymbol{\mathcal{L}}(\boldsymbol{w}_o, b_o, \boldsymbol{\alpha}_o, \boldsymbol{\beta}_o, \boldsymbol{\xi}_o) = 0 &\Leftrightarrow \beta_{i,o} = C - \alpha_{i,o} \\
y_i(\boldsymbol{w}_o^T \boldsymbol{P}\boldsymbol{x}_i - b_o) - 1 + \xi_{i,o} \geq 0, &\quad i = 1, \ldots, N \\
\beta_{i,o} \geq 0, 0 \leq \alpha_{i,o} \leq C, \xi_{i,o} \geq 0, \beta_{i,o}\xi_{i,o} = 0 &\quad i = 1, \ldots, N \\
\alpha_{i,o}[y_i(\boldsymbol{w}_o^T \boldsymbol{P}\boldsymbol{x}_i - b_o) - 1 + \xi_{i,o}] \geq 0, &\quad i = 1, \ldots, N,
\end{aligned}
\tag{18}
$$

The Wolfe-dual problem as well as the hyperplane are the same as in the separable case, i.e equations (12), (13) and (14), since the slack variables and their Lagrange multipliers do not appear in it.

2.4 Deflation Kernel

Instead of solving the optimization problem of the previous section for the deflated data in every iteration as our algorithm required in order to extract the desired knowledge, the formulation of the optimization problem in the previous section allows us to incorporate the deflation transformation of each iteration in the existing optimization problem. This is possible if we consider the deflation as a kernel function, which we define as

$$
K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{P}(\boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P})^{-1}\boldsymbol{P}\boldsymbol{x}_j \tag{19}
$$

and the feature map as

$$
\Phi(\boldsymbol{x}) = (\boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P})^{-1/2}\boldsymbol{P}\boldsymbol{x}. \tag{20}
$$

So

$$
\begin{aligned}
K(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \langle \Phi(\boldsymbol{x}_i)\Phi(\boldsymbol{x}_j) \rangle \\
&= \left((\boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P})^{-1/2}\boldsymbol{P}\boldsymbol{x}_i\right)^T \left((\boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P})^{-1/2}\boldsymbol{P}\boldsymbol{x}_j\right) \\
&= \boldsymbol{x}_i^T \boldsymbol{P}(\boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P})^{-1/2}(\boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P})^{-1/2}\boldsymbol{P}\boldsymbol{x}_j \\
K(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \boldsymbol{x}_i^T \boldsymbol{P}(\boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P})^{-1}\boldsymbol{P}\boldsymbol{x}_j
\end{aligned}
\tag{21}
$$

This notation gives us the advantage that the explicit computation of all the training samples is no longer needed, but we only need to compute the dot product of the vectors in the feature space, i.e. the Hessian matrix of (13), using the kernel trick. If we use matrix notation for the data instead of vectors, as defined in Sec.2.1, the above functions are expressed as $\Phi(\boldsymbol{X}) = (\boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P})^{-1/2}\boldsymbol{P}\boldsymbol{X}$ and $K(\boldsymbol{X}) = \boldsymbol{X}^T\boldsymbol{P}(\boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P})^{-1}\boldsymbol{P}\boldsymbol{X}$.

2.5 Feature Extraction

The method described in the previous sections consists a dimensionality reduction technique. In each iteration of the procedure we obtain a new weighting vector, orthogonal to all the previously obtained ones, which is used as a projection vector in the feature extraction schemes. The number of iterations and subsequently the number of obtained weighting vectors defines the number of the resulting dimensionality. The fact that the weighting vectors are pairwise orthogonal implies that in each iteration we acquire new discriminant information regarding our data. The extracted data consist of samples of which each feature is the projection of the initial vector onto the corresponding weighting vector, $f_{\boldsymbol{w}^k}(\boldsymbol{x}) = \boldsymbol{w}^{kT}\boldsymbol{x}$.

If we denote by $\boldsymbol{W}^k$ the augmented projection matrix which contains in its columns the weighting vector of each iteration, i.e. $\boldsymbol{W}^k = (\boldsymbol{w}^1, \boldsymbol{w}^2, \ldots, \boldsymbol{w}^k)$ we can express the feature extraction of the whole procedure in a compact way using the expression $\boldsymbol{X}_D^k = \boldsymbol{W}^{kT}\boldsymbol{X}$, where by $\boldsymbol{X}_D^k$ we denote the final matrix of the extracted data after k iterations and with a reduced dimensionality of k as well. The final form of the algorithm of WSVDA, which is implemented for the experiments in this paper is shown in Table 2.

Table 2. The implemented algorithm of WSVDA

input: training set data matrix $\boldsymbol{X}$ with N samples and corresponding labels y_i
output: extracted data matrix $\boldsymbol{X}_D^k$
1: compute the within scatter matrix $\boldsymbol{S}_W$ for the initial data
2: initial projection matrix $\boldsymbol{P} = \boldsymbol{I}_{d\times d}$
3: **for** k=1 to reduced dimensionality
4: check the condition number of $(\boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P})$, regularize by adding a small quantity to the diagonal elements if needed in order to achieve numerical stability
5: compute $(\boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P})^{-1}$
6: train SVMs using $\boldsymbol{H} = \boldsymbol{X}^T\boldsymbol{P}(\boldsymbol{P}\boldsymbol{S}_W\boldsymbol{P})^{-1}\boldsymbol{P}\boldsymbol{X}$
7: compute weighting vector $\boldsymbol{w}^k$ from (11)
8: normalize weighting vector $\boldsymbol{w}^k$
9: concatenate normalized $\boldsymbol{w}^k$ into $\boldsymbol{W}^T$
10: update projection matrix $\boldsymbol{P}$ using the normalized weighting vector of previous step, according to $\boldsymbol{P} = \boldsymbol{P}(\boldsymbol{I} - \boldsymbol{w}^k\boldsymbol{w}^{kT})$
11: **end**
12: use the normalized weighting vectors $\boldsymbol{w}$ for feature extraction, according to $\boldsymbol{X}_D^k = \boldsymbol{W}^{kT}\boldsymbol{X}$

3 Experimental Results

In this section we present the results of the experiments performed to assess the performance of WSVDA and compare it with the most commonly used techniques as PCA and LDA as well as state of the art methods as MMDA. After the dimensionality reduction of the datasets with the aforementioned techniques, classification is performed using KNN and NC algorithms. It was also considered valuable to compare these results with SVMs classification applied on the initial data.

In order to achieve higher credibility for our results we perform various instances of k-fold cross validation, with 1 fold being used as a training set and the rest $(k-1)$ folds being used as the test set. This approach offers the opportunity to use a small number of samples for the training phase, which is comparable or sometimes smaller than the number of features. That is, the dimensionality of the training set is higher than its cardinality, which is often the case for small sample size (SSS) problems. In such occasions we expect and we show that WSVDA has better performance. It is also important to mention that all the datasets were scaled uniformly to $[-1, 1]$.

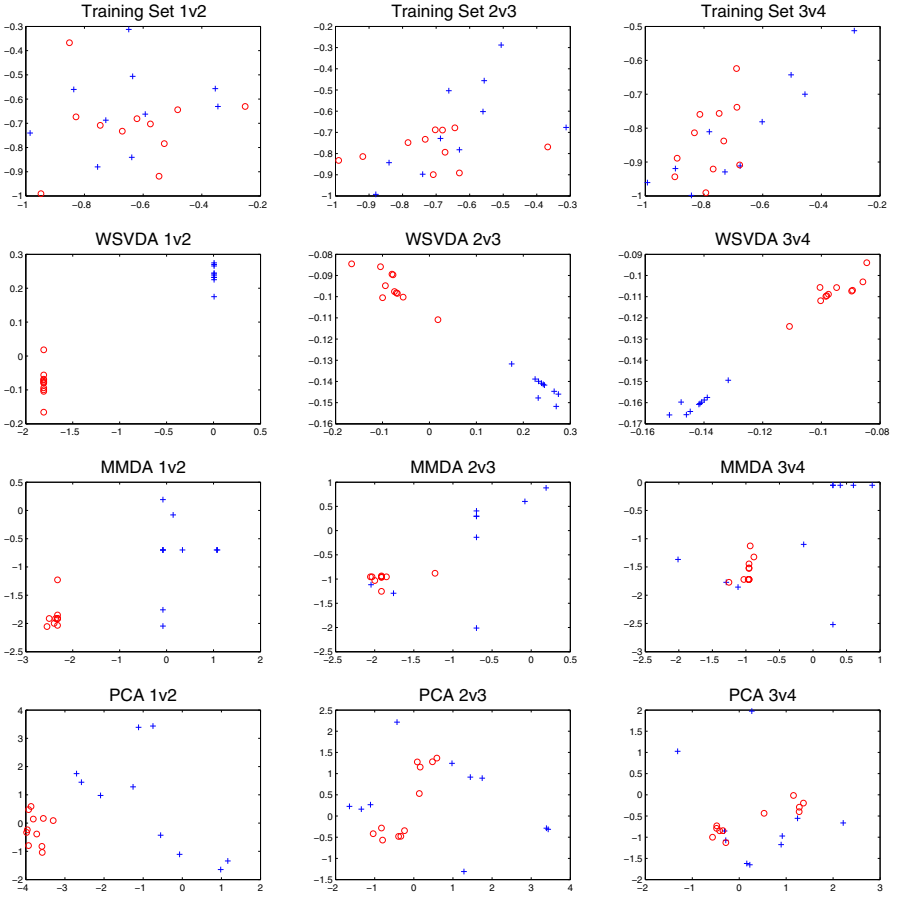


Fig. 2. Projection of pairs of features of Connectionist Bench (Sonar) dataset onto two-dimensional subspaces. In the first row three pairs of features of the initial data are shown. In the following three rows we can see the first three pairs of extracted features for WSVDA, MMDA and PCA. Best viewed in color.

3.1 Use of WSVDA for Visualization Purposes

One important attribute of WSVDA is the visualization capability, which is particularly useful for high-dimensional datasets. To demonstrate this attribute we use the Connectionist Bench dataset from the UCI Machine Learning Repository. In Fig. 2 the first three pairs of features are shown for the initial training data and after reducing their dimensionality with WSVDA, MMDA and PCA. Since the problem is binary LDA could not be used for two-dimensional visualization purposes.

We can observe that all three methods are capable of extracting discriminant information from the data and make the classification task easier compared to the initial data. However, it is important to note that only in the case of WSVDA the two classes are linearly separable for all the extracted features depicted on the figure. This means that except for the first extracted feature, the succeeding features provide additional and new discriminant information.

3.2 Dimensionality Reduction and Classification Results

In this section we present the experimental results for classification purposes using four different datasets. The experimental scenario includes a dimensionality reduction step using one of the following four techniques WSVDA, MMDA,

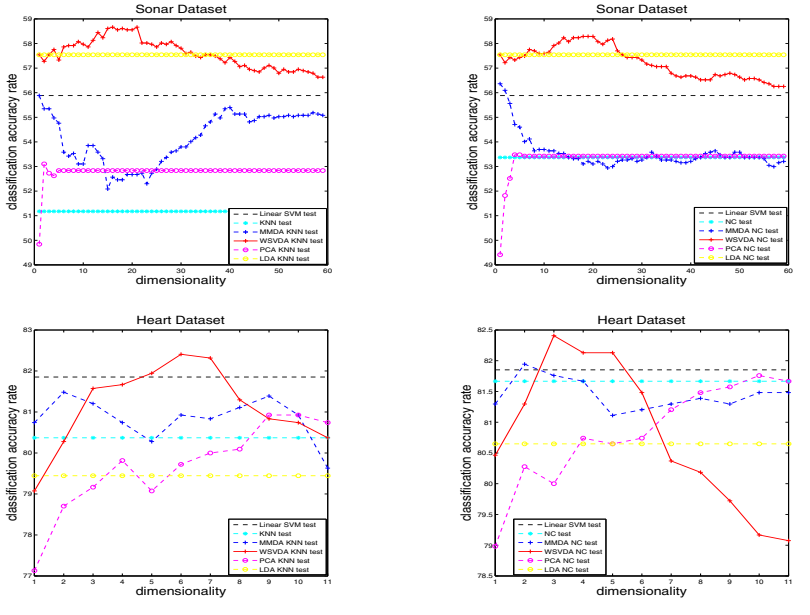


Fig. 3. Classification accuracy rates for Sonar and Heart datasets. WSVDA outperforms the other methods in all occasions in terms of classification accuracy rate and it is observable that it gains discriminant information from the successive subspaces where the optimization problem is solved. The number of neighbors for the KNN algorithm is 5 for this set of experiments, whereas the number of folds for the cross validation is 10 and 5 for Sonar and Heart datasets, respectively. Best viewed in color.

PCA, and LDA, and a classification step using either KNN or NC classification algorithms. Classification with Linear SVMs and KNN applied on the initial data is also performed, using the following datasets: 1. *Connectionist Bench (Sonar)* dataset contains 208 samples of 60 attributes that correspond to measurements of a sonar device for signals that are reflected on two different surfaces. 2. *Statlog (Heart)* dataset consists of 270 samples with 13 attributes that correspond to medical data related to heart. 3. *Wine Recognition Data* dataset contains 178 samples of 13 features which correspond to the chemical analysis of three varieties of wine. 4. *Splice-junction Gene Sequences* dataset consists of 3190 samples with 61 attributes that correspond to DNA sequences.

In Fig. 3 the average classification accuracy rates for the four dimensionality reduction techniques followed by k-Nearest Neighbor or Nearest Centroid classification are shown for Sonar and Heart datasets, first and second row respectively. On the horizontal axis we have the number of reduced dimensions. Since the problems are binary, LDA results to one-dimensional extracted data, so only one accuracy rate is available.

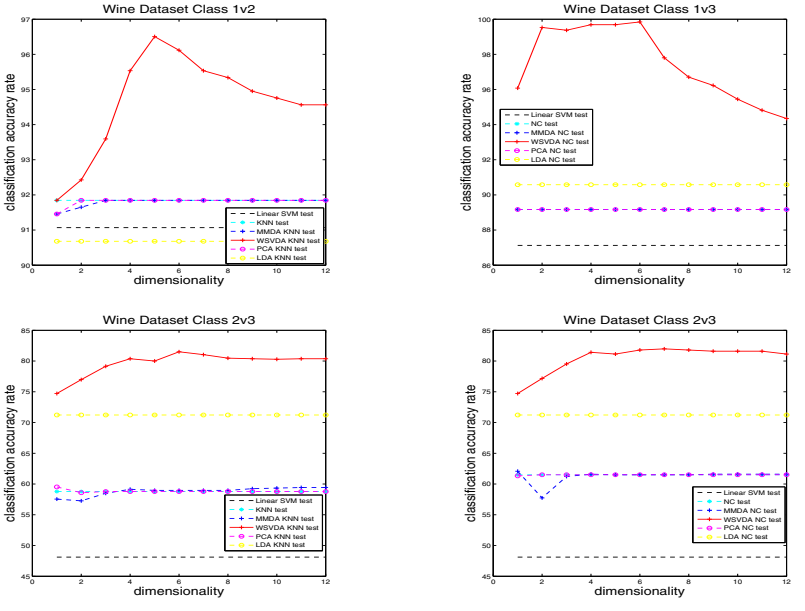


Fig. 4. Classification accuracy rates for Wine dataset. The top left subfigure corresponds to the binary problem between classes 1v2 for 5-NN classification and 5-fold cross-validation, whereas the top right subfigure corresponds to the binary problem between classes 1v3 for NC classification and 7-fold cross-validation. The second row corresponds to the binary problem between classes 2v3, which are more difficult to discriminate and due to 10-fold cross-validation, which results to very small training sample, we observe very low classification rates. WSVDA outperforms the other methods in all occasions in terms of classification accuracy rate showing that is a suitable technique for small sample size problems. Best viewed in color.

The comparison between the different methods highlights the superior performance of WSVDA in comparison to MMDA, PCA, LDA and classification with Linear SVMs and KNN or NC, applied on the initial data. In the left column we see the results for KNN and in the right one we see the results for NC. The results for the Wine Dataset are shown in Fig. 4.

Table 3 offers a detailed view over the classification results for all the datasets examined and all the approaches followed. The accuracy rates correspond to the highest value over the average accuracy rates of the cross validation and for every possible reduced dimensionality. They are instances from experiments with different parameters such as the number of folds, the number of nearest neighbors and the regularization, but same for each line of the table. This is the reason for big differences observed in classification rates. For the Wine dataset for example, the low classification rates for class 2 against 3 are due to the biggest overlapping between these classes in comparison to the other combination and due to the larger number of folds which has as a result a very small training set. This fact results to a more difficult classification task, for which WSVDA proves to be quite robust.

Table 3. Classification Results

Data sets	KNN	NC	SVM	PCA +KNN	LDA +KNN	MMDA +KNN	WSVDA +KNN	PCA +NC	LDA +NC	MMDA +NC	WSVDA +NC
Sonar	51.18	53.37	55.88	53.10	57.54	55.88	**58.66**	53.48	57.54	56.36	58.29
Heart	80.37	81.67	81.85	80.93	79.44	81.48	**82.41**	81.76	80.65	81.94	**82.41**
Wine											
1vs2	91.84	90.68	91.07	91.84	90.68	91.84	**96.50**	90.68	90.29	90.68	96.12
1vs3	89.32	89.17	87.13	89.32	90.42	89.32	**100**	89.17	90.58	89.17	99.84
2vs3	58.77	61.51	48.11	59.53	71.23	59.43	81.51	61.51	71.23	62.08	**81.98**
Splice	68.17	80.74	76.81	68.42	77.44	78.88	79.16	72.02	78.36	81.40	**81.43**

4 Conclusions

A novel dimensionality reduction method has been proposed that combines the minimization of the within class scatter matrix with the maximization of the margin between the classes in each projection. The proposed approach uses an iterative feature extraction with deflation kernels that transform the original data to perpendicular subspaces where a quadratic optimization problem is solved. Thus, the discriminant information that lie in the subspace which is perpendicular to the only dimension that standard SVM extract is exploited for better discriminality and classification. Experimental results on several datasets illustrate the superiority of the proposed approach against other popular dimensionality reduction methods.

Acknowledgments. This research has received funding from the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013) ERC Starting Grant agreement 204871-HUMANIS.

References

1. Cover, T.M., Hart, P.E.: Nearest Neighbor Pattern Classification. IEEE Transactions In Information Theory, 21–26 (1967)
2. Duda, O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley (2001)
3. Scholkopf, B., Smola, A.: Learning with Kernels. MIT, Cambridge (2002)
4. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer (2002)
5. Pearson, K.: On Lines and Planes of Closest Fit to Systems of Points in Space. Philosophical Magazine 2, 559–572 (1901)
6. Hotelling, H.: Analysis of a Complex of Statistical Variables into Principal Components. Journal of Educational Psychology 24, 417–441 (1933)
7. Scholkopf, B., Smola, A., Muller, K.R.: Nonlinear component analysis as a Kernel eigenvalue problem. Neural Computation 10, 1299–1319 (1998)
8. Alpaydin, E.: Introduction to Machine Learning. MIT Press (2004)
9. Juwei, L., Plataniotis, K.N., Venetsanopoulos, A.N.: Face recognition using Kernel direct discriminant analysis algorithms. IEEE Transactions on Neural Networks 14, 117–126 (2003)
10. Kocsor, A., Kovács, K., Szepesvári, C.: Margin Maximizing Discriminant Analysis. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 227–238. Springer, Heidelberg (2004)
11. Tefas, A., Kotropoulos, C., Pitas, I.: Using Support Vector Machines to Enhance the Performance of Elastic Graph Matching for Frontal Face Authentication. IEEE Transactions on Pattern Analalysis and Machine Intelligence 23(7), 735–746 (2001)
12. Zafeiriou, S., Tefas, A., Pitas, I.: Minimum Class Variance Support Vector Machines. IEEE Transactions on Image Processing 16(10), 2551–2564 (2007)
13. Kung, S.Y., Diamantaras, K.I.: Neural networks for extracting unsymmetric principal components. In: Neural Networks for Signal Processing, pp. 50–59. IEEE Press, New York (1991)
14. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. In: Data Minining Knowledge Discovery, vol. 2, pp. 121–167 (1998)
15. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
16. Fletcher, R.: Practical Methods of Optimization, 2nd edn. Wiley, New York (1987)

A New Learning Strategy of General BAMs

Hoa Thi Nong and The Duy Bui

Human Machine Interaction Laboratory
Vietnam National University, Hanoi
nongthihoa@gmail.com, duybt@vnu.edu.vn

Abstract. Bi-directional Associative Memory (BAM) is an artificial neural network that consists of two Hopfield networks. The most important advantage of BAM is the ability to recall a stored pattern from a noisy input, which depends on learning process. Between two learning types of iterative learning and non-iterative learning, the former allows better noise tolerance than the latter. However, interactive learning BAMs take longer to learn. In this paper, we propose a new learning strategy that assures our BAM converges in all states, which means that our BAM recalls perfectly all learning pairs. Moreover, our BAM learns faster, more flexibility and tolerates noise better. In order to prove the effectiveness of the model, we have compared our model to existing ones by theory and by experiments.

Keywords: Bi-directional Associative Memory, Multiple Training Strategy, Hopfield neural network.

1 Introduction

BAM is an associative memory that has two directions. Structure of BAM consists of two Hopfield neural networks with two ways of association, i.e. auto-association and hetero-association [7]. This class of neural networks is good for pattern recognition and artificial intelligence. The most important attribute of BAM is the ability to recall stored patterns from a noisy input. Output of recalling process directly depends on results of learning process. Learning process is performed by a learning strategy that can be divided into two types: non-iterative learning [12,21] and iterative learning [9,19]. Results of previous studies show that iterative learning BAMs recall better than non-interactive ones. Therefore, many iterative learning BAMs have been developed for recognition applications that manipulate noisy inputs.

Studies of BAMs focus on two main directions: improving math properties of models and creating new models. Some mathematicians showed output functions that assure conditions about exponential stability but not considering noisy level of input and learning process of BAMs [3,22,10,6]. Other studies proposed new models that improve the ability of storage and the ability of recall [16,17,9,19,12,13]. However, noise tolerance of these models is weak. Moreover speed of learning process is slow when the number of patterns is large.

In this paper, we propose a novel learning strategy for general BAMs. Our strategy performs iterative learning until we obtain the condition that guarantees the recall of all

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 213–221, 2012.

learning pairs, meaning that our novel model converges in all states. Updating connection weights is flexible by changing pair weights in an iteration of learning process. As a result, our BAM learns faster and recalls better. Moreover, we prove advantages of our novel model in theory and by experiments.

The rest of the paper is organized as follows. BAM models are described in Section 2. Related works are presented in Section 3. In section 4, we present our novel learning strategy and prove advantages. Section 5 shows our experiments and compares with other models.

2 Bidirectional Associative Memory Models

Structure of BAM

BAM is a two-layer hetero-associative feedback neural network model introduced by Kosko [7]. As shown in 1, the input layer F_A includes n binary valued neurons $(a_1, a_2, ..., a_n)$ and the output layer F_B comprises of m binary valued components $(b_1, b_2, ..., b_m)$. Now we have $A = \{0, 1\}^n$ and $B = \{0, 1\}^m$. BAM can be denoted as a bi-directional mapping in vector spaces $W : R_n \longleftrightarrow R_m$.

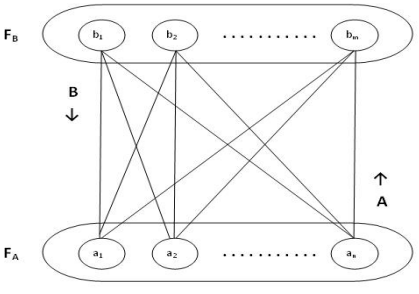


Fig. 1. Structure of Bidirectional Associative Memory

Learning Process

Assume that BAM learns N pattern pairs, $(A_1, B_1), (A_2, B_2), ..., (A_N, B_N)$.The learning pattern pairs are stored in the correlation matrix as follows [7]:

$$W_0 = \sum_{i=1}^{N} A_i^T B_i \ . \tag{1}$$

where A_i and B_i are the bipolar mode of the i^{th} learning pattern pair.

A learning rule of BAM shows the multiple training strategy [21]:

$$W = \sum_{i=1}^{N} q_i A_i^T B_i \ . \tag{2}$$

where q_i is pair weight of the i^{th} training pair.

Recalling Process
To retrieve one of the nearest (A_i, B_i) pairs from the network when any (α, β) pair is presented as an initial condition to the network. Starting with a value of (α, β) determine a finite sequence (α', β'), (α'', β''),... until an equilibrium point (α_F, β_F) is reached [7], where

$$\beta' = \phi(\alpha W) \tag{3}$$

$$\alpha' = \phi(\beta' W^T) \tag{4}$$

$$\phi(F) = G = (g_1, g_2, ..., g_n) \tag{5}$$

$$F = (f_1, f_2, ..., f_n) \tag{6}$$

$$g_i = \begin{cases} 1, f_i \geq 0 \\ 0, else \end{cases} \tag{7}$$

Energy Function
For any state (A_i, B_i) , an energy function is defined by [7]

$$E_i = -A_i W B_i \tag{8}$$

3 Related Works

3.1 Studies about BAMs

All studies of BAMs have been developed with two branches including building a proper output function and designing new models. The first branch focuses on math properties of models while the second one focuses on performance of models.

In the first branch, a series of papers proposed conditions for stability of BAM. Deqin Chen and Kelin Li [3] used delays and impulses in a continuous-time BAM to represent the dynamical behaviours of neurons. Yonggang Chen [22] added time-varying delays to show more clearly the dynamics of network in a discrete-time BAM. This class of BAMs is easy for analysis the stability and designing simulator in computer. Zuoan Li [10] added reaction-diffusion terms to describe the chaos of BAMs that work with fuzzy data. H.S.Shu [6] focused on effect of modelling errors in a uncertain stochastic BAMs by using uncertain parameters. However, these studies only ensure the ability of recall.

In the second branch, several new models were created to improve learning process and recalling process. Maria and Corlenio [12] proposed a new Alpha-Beta BAM that works with two binary operators α and β. Advantages of this model was recalling perfectly without iterations and stability problems. Similarly, A. Vzquez, Sossa, and A. Garro [13] showed a general BAM that recalls perfectly the whole fundamental set of learning patterns without an iterative algorithm. Chartier and Boukadoum [16,17] described a new model that has a self-convergent interactive learning rule and a new nonlinear output function to recall two outputs. Dan Shen, Jose and Cruz [14] expanded a

general BAM by choosing proper weights of pattern pairs to possess a maximum noise tolerance set. However, noise tolerance of these models is slow when inputs are high noisy.

3.2 Learning Process of BAMs

BAMs have two ways to learn including non-iterative learning and iterative learning. BAMs without iteration are developed earlier.

First, many previous studies have learning strategy without iteration because learning process and recalling process are quite simple in one iteration. A BAM worked well with true-color images by A. Vzquez [13]. Similarly, Maria [12] proposed a BAM that works with words in English-Spanish dictionary. Yeou-fang Wang et al proposed a new model with multiple training but learning process was performed in one iteration. Other researchers also designed Hopfield neural networks that learn in one iteration [15,2]. However, little data is retrieved from a high noisy input.

Second, several iterative learning BAMs were proposed to improve recalling process. Chartier and Boukadoum [16,17] obtained an on-line learning algorithm through changing an additional value based on input, output of BAM at a point of time. Leung [9] designed a multiple training algorithm that learns a pattern pair in each iteration. Tao Wang [19] showed a multiple training method that learns all pattern pairs in each iteration. Other researchers also proposed learning rules that new connection weights directly depend on old connection weights [11,5,8,18]. However, the complexity of learning process is high when increasing the number of learning patterns.

3.3 Given Learning Strategies

In the most of early researches on iterative learning, BAMs and Hopfield neural networks change connection weights according to the following way: a new connection weight is assigned to the sum of the old connection weight and an additional value. We can give some examples to illustrate.

The following equation was proposed in [11] to show Hebbian rule.

$$W_{ij}^{new} = W_{ij}^{old} + \gamma_v^+ x_i^D(t) A_j^t \ . \tag{9}$$

where W_{ij}^{new} and W_{ij}^{old} is new connection weight and old connection weight between neuron **i** and **j**, and γ is learning rate and x_i^D and A_j^t is output and input of neuron i at the time t.

Hebb [4] showed the Hebb rule in the following equation:

$$W(new) = W(old) + xy \ . \tag{10}$$

where x is input and y is output.

After Hebbian rule was improved to have delta rule by B. Widrow [1].

$$W(new) = W(old) + \alpha(d - y)x \ . \tag{11}$$

where d is desired output and α is learning rate.

In 1992, Kohonen [20] proposed a new rule for learning that is called Kohonens rule:

$$W(new) = W(old) + \alpha(x - W(old)). \tag{12}$$

where α is learning parameter.

Similarly, updating connection weights by Sylvain Chartier et al. [16,17]:

$$W_{[k+1]} = W_{[k]} + \eta(Y_{[0]} - Y_{[t]})(X_{[0]} - X_{[t]})^T \ . \tag{13}$$

where $W_{[k]}$ is weight matrix at the time **k** and η is learning parameter and y[0], y[t], x[0], x[t] are output and input of the first and t^{th} interaction in each trial learning.

Chi Sing Leung described a learning rule that was shown in [9]:

$$m_{ij}^{u+1} = m_{ij}^{u} + \Delta m_{ij}^{u}. \tag{14}$$

where m_{ij}^{u} is connection weight between neuron **i** and **j** after learning all sample pattern pairs from **1** to **u-1**, and Δm_{ij}^{u} ensures find one of the solution connection matrices.

In 1994, a similar learning rule was proposed by Tao Wang et al. [19]:

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij}. \tag{15}$$

where $W_{ij}(t+1)$ is connection weight between neuron **i** and **j** at the time **t+1**, and ΔW_{ij} presents updating weights base on the sum of signals come from two directions.

Through typical examples, we show that an additional value of connection weights depends on learning parameters and learning parameters unchanged in all the time of learning process. Thus, learning is inflexibility. Moreover, new connection weights directly depend on old connection weights. Therefore, networks converge more slowly if old connection weights are not near desired values. We propose a new learning strategy that learns flexibility and more quickly than given studies in the following section.

4 Our Approach

4.1 Our Learning Strategy

The goal of our learning strategy is to build a connection weight matrix that assures our model always converges in all states. That means our model can recall correctly all learning pairs.

Assume that we want to learn N pattern pairs, $(A_1, B_1), (A_2, B_2), ..., (A_N, B_N)$. Let q_i, E_i be pair weight and energy of i^{th} pair, ε be the threshold of energy to represent the stop condition of modifying pair weights, and δ is ratio of E_i to ε. Our learning strategy comprises of two steps that can be described as follows.

Step 1: sets up initial values for variables:

- Set the value of $\varepsilon = 10^{-5}$. ε is very small number and greater than 0. Meaning, energy in each state reachs to 0, which is the condition for convergence of Hopfield neural networks.
- Set the value of of $q_i = 1$ where i=1,..,N to get original connection weights.
- Formulate the original connection weights W_0 by the equation 1.

Step 2: performs weight updating iteratively:

- Formulate W according to the equation 2.
- Then, compute E_i with the equation 8.
- Based on value of E_i, update q_i.

Repeat Step 2 until all E_i reach the stop condition.

Rules for Updating q_i

R1: if $|E_i| \leq \varepsilon$, do not change q_i.
R2: if $|E_i| \geq \varepsilon$, change q_i.

q_i is computed with the following rule: $\delta = |E_i|/\varepsilon$ and $q_i \simeq q_i/\delta$.

4.2 Advantages of the New Learning Strategy

In our novel learning strategy, new connection weights indirectly depend on the old connection weights. New connection weights are changed by increasing or decreasing or unchanging of the learning parameter of each pattern pair in a iteration. Consequently, **updating connection weights is flexible**.

Learning process is performed faster by three reasons. Firstly, increasing or decreasing connection weights is made by multiplication or division, respectively, by addition or subtraction in previous studies [9,19,11,4,1,20,16,17]. Secondly, value of learning parameters is sharply decreased/increased after each iteration. As a result, value of each element in connection weight matrix drops/rises strongly according to Equation 2. Then, our model learns all pattern pairs in a iteration so the complex of learning process increase less when increasing the number of learning patterns.

In summary, our novel learning strategy learns faster and more flexibly. To test effectiveness of our novel learning strategy, we made some experiments with character images. Results show that noise tolerance of our model is better than previous studies.

5 Experiments

We use learning patterns which are 56 images. Each image has size of 8x5 and is converted to a vector of 40 values. Figure 2 presents sample images and each pair show a character in lower and upper correspond to each association stored in BAM. In all experiments, we randomly changed 40-50% pixels of the input image to obtain 105 noisy input. Five learning strategies are coded to prove the advantages in learning process and recalling process of our model.

To assess the noise tolerance of models, we use two new measurements, which are percent of input noise (n) and percent of successful recall (P). P and n are formulated by two following equations:

$$n = (t/N) * 100 \tag{16}$$

where N is number of pixels of an image (N=40) and t is the number of noisy pixels.

$$P = (k/N) * 100 \tag{17}$$

where k is the number of correctly-recalled pixels.

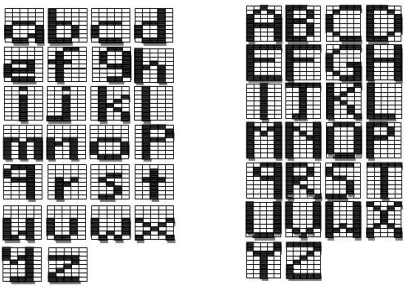


Fig. 2. Learning pattern pairs in experiments

5.1 Experiment 1 : Our Model Compared to Non-iterative Learning Models

We choose two typical models, including Alpha-Beta BAM (ABBAM) [12] and Wang's BAM (WBAM) [21] to compare the ability of noise tolerance . Results that obtained from three models are presented in Figure 3.

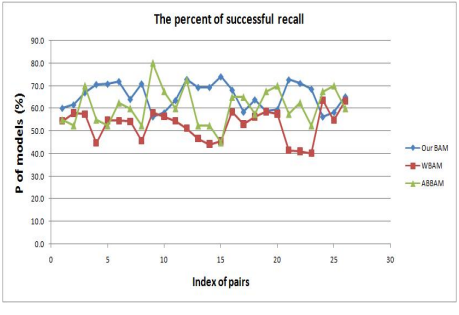


Fig. 3. The percent of successful recall of models in Experiment 1

The percentage of successful recalls of each pattern pair is shown. Lines from this graph show that our novel model recalls better than Alpha-Beta BAM with the most of pairs and Wang's one with all pairs except 24^{th} pair.

5.2 Experiment 2 : Our Model Compared to Iterative Learning Models

Our model is compared to two models of Leung (LBAM) [9] and Tao Wang (TBAM) [19]. We choose two models because they have quite optimal learning rule. Both learning process and recalling process are compared to prove completely the effectiveness of our model.

First, we consider speed of learning of three models. Our model learns all sample patterns in an iteration. Lung's BAM learns each pattern pair in a iteration according to the order of pattern pairs. Tao's BAM learns all pattern pairs in each iteration. Consequently, our model learns fastest and Tao's BAM learns slowest.

Second, we compare to the ability of noise tolerance. The percentage of successful recall of three models is shown in Figure 4. Figure 4 show that our model recalls better than Leung's BAM with the most of pairs and approximately equal to Tao's one.

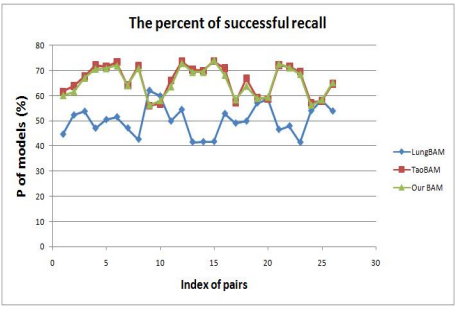


Fig. 4. Results of Experiment 2

To sum up, our model is better previous models both learning process and noise tolerance.

6 Conclusion

In this paper, we have proposed a new learning strategy for general BAMs. Our novel model modifies connection weight matrix until obtaining the condition of convergence. This condition assures BAMs recall correctly all learning pairs. Value of pair weights are changed in each iteration to modify connection weights. As a results, our model has important advantages in both learning process and recalling process. Our model learns more quickly and recall better. Moreover, we have proved experimentally the advantages of our model.

On-line learning strategy is an idea method that allow BAMs learn new learning pairs in any time. Thus, artificial neural networks can meet the need of real-time applications. Moreover, fuzzy data appears more popular and be used in many applications such as medicine, financial, control. It is a motivation to expand the ability of learning with with fuzzy data. We will investigate to develop an on-line learning strategy and expand the capacity of learning with fuzzy data in the future.

Acknowledgement. We would like thank to Vietnam National Foundation of Science and Technology Development. This work is also supported by Nafosted research project No. 102.02-2011.13.

References

1. Widrow, B., Hoff, M.E.: Adaptive switching circuits. IRE WESCON Conv. Rec. 4 (1960)
2. Pandey, B., Ranjan, S., Shukla, A., Tiwari, R.: Sentence Recognition Using Hopfield Neural Network. IJCSNS International Journal of Computer Science Issues 7(6) (2010)

3. Chen, D., Li, K.: Exponential Stability of BAM Neural Networks with Delays and Impulses. IJCSNS International Journal of Computer Science and Network Security 6(10), 94–99 (2006)
4. Hebb, D.O.: The Organization of Behavior: A Neuropsychological Theory. Wiley, New York (1949)
5. Costantini, G., Casali, D., Perfetti, R.: Neural Associative Memory Storing Gray-Coded Gray-Scale Images. IEEE Transactions on Neural Networks 14(3), 703–707 (2003)
6. Shu, H.S., Lv, Z.W., Wei, G.L.: Robust stability for stochastic bidirectional associative memory neural networks with time delays. Journal of Physics, Conference Series 96 012003 (2008)
7. Kosko, B.: Bidirectional associative memory. IEEE Transactions on on Systems, Man, and Cybernetic 18(1) (1988)
8. Lenze, B.: Improving Leungs Bidirectional Learning Rule for Associative Memories. IEEE Transactions on Neural Networks 12(5), 1222–1226 (2001)
9. Leung, C.S.: Optimum Learning for Bidirectional Associative Memory in the Sense of Capacity. IEEE Transactions on Neural Networks 24(5) (1994)
10. Li, Z.: Dynamics in BAM Fuzzy Neural Networks with Delays and Reaction-Diffusion Terms 1(20), 979 – 1000 (2008)
11. Ideguchi, M., Sato, N., Osana, Y.: Hetero Chaotic Associative Memory for Successive Learning with Give Up Function. In: 2005 International Symposium on Nonlinear Theory and its Applications, pp. 42–45 (2005)
12. Acevedo-mosqueda, M.E., Yanez-marquez, C., Lopez-yanez, I.: Alpha-Beta Bidirectional Associative Memories Based Translator. IJCSNS International Journal of Computer Science and Network Security 6(5), 190–194 (2006)
13. Vazquez, R.A., Sossa, H., Garro, B.A.: A New Bi-directional Associative Memory, 367–380 (2006)
14. Shen, D., Cruz Jr., J.B.: Encodding strategy for maximum noise tolerance bidiretional associative memory. IEEE Transactions on Neural Networks (2003)
15. Singh, T.: Performance analysis of Hopfield model of neural network with evolutionary approach for pattern recalling. International Journal of Engineering Science and Technology 2(4), 504–511 (2010)
16. Sylvain Chartier, M.B.: A Bidirectional Heteroassociative Memory for Binary and Grey-Level Patterns. IEEE Transactions on Neural Networks 17(2), 385–396 (2006)
17. Sylvain Chartier, M.B., Amiri, M.: BAM Learning of Nonlinearly Separable Tasks by Using an Asymmetrical Output Function and Reinforcement Learning. IEEE Transactions on Neural Networks 20(8), 1281–1292 (2009)
18. Wang, T., Zhuang, X., Xing, X.: Weighted Learning of Bidirectional Associative Memories by Global Minimization. IEEE Transactions on Neural Networks 3(6) (1992)
19. Wang, T., Zhuang, X., Xing, X.: Memories with Optimal Stability. IEEE Transactions on Systems, Man, and Cybernetic 24(5) (1994)
20. Kohonen, T.: Self-organization and Associative Memory. Springer, Berlin (1988)
21. Wang, Y.F., Cruz Jr., J.B., Mulligan Jr., J.H.: Guaranteed recall for all training patterns of Bidirectional Associative Memory. IEEE Transactions on Neural Networks 2(6) (1991)
22. Chen, Y., Bi, W., Wu, Y.: Delay-Dependent Exponential Stability for Discrete-Time BAM Neural Networks with Time-Varying Delays. In: Discrete Dynamics in Nature and Society 2008, pp. 3–15 (2008)

Proximity-Graph Instance-Based Learning, Support Vector Machines, and High Dimensionality: An Empirical Comparison

Godfried T. Toussaint[1] and Constantin Berzan[2]

[1] Faculty of Science, New York University Abu Dhabi, Abu Dhabi, United Arab Emirates
gt42@nyu.edu
[2] Department of Computer Science, Tufts University, Medford, MA 02155, USA
constantin.berzan@tufts.edu

Abstract. Previous experiments with low dimensional data sets have shown that Gabriel graph methods for instance-based learning are among the best machine learning algorithms for pattern classification applications. However, as the dimensionality of the data grows large, all data points in the training set tend to become Gabriel neighbors of each other, bringing the efficacy of this method into question. Indeed, it has been conjectured that for high-dimensional data, proximity graph methods that use sparser graphs, such as relative neighbor graphs (RNG) and minimum spanning trees (MST) would have to be employed in order to maintain their privileged status. Here the performance of proximity graph methods, in instance-based learning, that employ Gabriel graphs, relative neighborhood graphs, and minimum spanning trees, are compared experimentally on high-dimensional data sets. These methods are also compared empirically against the traditional *k*-NN rule and support vector machines (SVMs), the leading competitors of proximity graph methods.

Keywords: Instance-based learning, Gabriel graph, relative neighborhood graph (RNG), minimum spanning tree (MST), proximity graphs, support vector machines (SVM), sequential minimal optimization (SMO), machine learning.

1 Introduction

Instance-based learning algorithms are among the most attractive methods used today in many applications of machine learning to a wide variety of pattern recognition problems. The quintessential instance-based learning algorithm is the well-known *k*-Nearest Neighbor (*k*-NN) rule, whereby a new unknown pattern is classified into the class most heavily represented among its *k* nearest neighbors present in the training set. Two of the most attractive features of this method are immediately evident: (1) its simplicity, and (2) the fact that no knowledge is required about the underlying distribution of the training data. Nevertheless, unanswered questions about the rule's performance left doubts among its potential users. In 1967 Cover and Hart [4] showed

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 222–236, 2012.

that, under some continuity assumptions the number n of patterns in the training data becomes infinite, the asymptotic probability of misclassification of the 1-NN rule is at most twice the Bayes error. Furthermore Devroye showed that for all distributions, the asymptotic probability of misclassification of the k-NN rule approaches the Bayes error provided that k and n approach infinity, and the ratio k/n approaches zero [7-8]. These constraints are satisfied for example when $k = n^{1/2}$. Thus the ultimate classificatory power of the k-NN rule was finally firmly established. Despite this great news, resistance to using k-NN rules in practice persisted, fueled by several misconceptions, one being that all n points must be stored, thus requiring too much storage. It was pointed out in 1979 that the decision boundary of the 1-NN rule remains unchanged when the data points surrounded by Voronoi neighbors of the same class are discarded (in parallel) [23-24]. A second false claim frequently encountered is that in order to determine the nearest neighbor of an unknown query point X, the distances between X and *all* the n points in the training data must be computed. Today there exist a plethora of methods for avoiding such exhaustive search.

The above-mentioned false claims notwithstanding, in practice it is desired to reduce the size of the training set, or the concomitant memory of the data structure into which the training set is embedded, as much as possible, while maintaining a low probability of misclassification. Therefore much research has been devoted to this topic, yielding a cornucopia of different approaches [20]. One elegant and promising approach generalizes Voronoi (or Delaunay triangulation) editing [23] to incorporate more general proximity graphs [24]. In the latter approach, for any definition of proximity, data points with the property that all their proximity-graph neighbors belong to the same class are discarded (in parallel). Previous experiments with low-dimensional data sets have shown that proximity graph methods that used the Gabriel graph were among the best machine learning algorithms for pattern classification applications [2-3, 24, 28]. However, as the dimensionality of the data grows large, all data points in the training set tend to become Gabriel neighbors of each other, effectively forsaking the notion of proximity, and bringing the efficacy of this method into question [14]. We conjectured that for high-dimensional data, methods that use sparser graphs, such as relative neighbor graphs (RNG) or minimum spanning trees (MST), would yield better results by avoiding proximity-graphs with too many edges. In this paper we conduct experiments that confirm this hypothesis.

Here the performance of various proximity graph methods for instance-based learning is compared experimentally using Gabriel graphs, relative neighborhood graphs, and minimum spanning trees, on a group of high-dimensional data sets. These three graphs vary considerably in the number of neighbors they admit, thus allowing us to test the hypothesis. For completeness, these methods are also compared empirically against the traditional k-NN rule that does not condense the data, and the optimization-based support vector machine (SVM), a leading competitor of proximity graph methods that has also enjoyed widespread success in practice.

2 Proximity Graphs

Given a set S of $n \geq 3$ points in the plane, two points a, b are Gabriel neighbors if all other points in S lie in the exterior of the smallest circle that contains a and b, i.e., the circle with diameter determined by points a and b. The Gabriel graph of S is obtained by joining all pairs of points with an edge provided they are Gabriel neighbors of each other. Figure 1 (left) shows the Gabriel graph of a set of 42 points in the plane. Note that the graph is adaptive in the sense that the number of edges it contains (or alternately the number of bounded regions it encloses) may be large or small relative to the number of points in the set. This property allows proximity-graph decision rules to automatically vary the number and spatial location of the neighbors they utilize, as a function of the local density and structure of the data. The Gabriel graph in Figure 1 (left) contains 27 bounded regions.

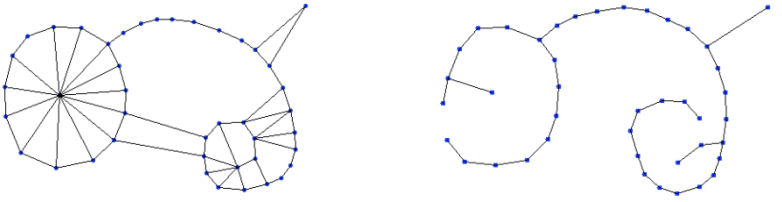


Fig. 1. The Gabriel graph (left) and the MST (right)

The properties of the Gabriel graph most relevant to the present research are that every minimum spanning tree (MST) of a set of points S is a sub-graph of the Gabriel graph of S, and that the Gabriel graph of S is a sub-graph of the Delaunay triangulation of S [21]. Devroye [6] has shown that for almost all probability density functions, as the number of points grows to infinity, the expected number of edges in the Gabriel graph approaches $2^{d-1}n$, where d is the dimensionality of the space.

The minimum spanning tree of a set of points is the sparsest connected proximity graph in terms of the number of edges it contains, whereas the Delaunay triangulation is much denser. Figure 1 (right) shows the MST for a similar set of points. Naturally, since this proximity graph is a tree it contains no bounded regions whatsoever. Figure 2 (left) shows the Delaunay triangulation of a similar set of points along with its dual, the Voronoi diagram.

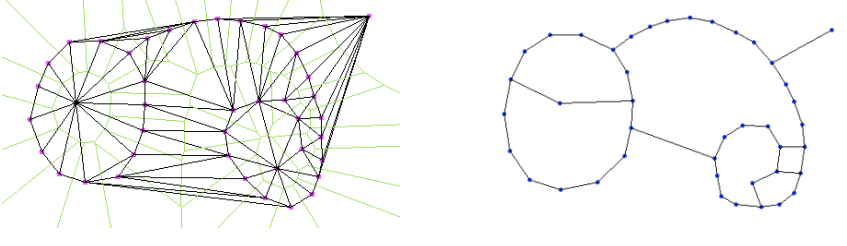


Fig. 2. The Delaunay triangulation with Voronoi diagram (left) and the RNG (right)

Toussaint [21] defined the relative neighborhood graph (RNG) of a set S of n points as the graph obtained by adding an edge between every pair of points a and b provided that $d(a, b) \leq \max [d(a, x), d(b, x)]$ for all x in S. The RNG is generally (for random configurations of points) much sparser than the Gabriel graph. Figure 2 (right) shows the RNG of 42 points arranged much like the points in Figure 1, but here the number of bounded regions is only 6. All four graphs constitute snapshots of a continuum of proximity graphs called β-skeletons that vary from the dense Delaunay triangulation to the sparse minimum spanning tree, [15]. These proximity graphs have been applied to a wide variety of problems in different fields [14].

3 Reducing the Size of the Training Data

The first published algorithm for reducing the size of the training data for use in nearest neighbor decision rules was the *condensed* nearest neighbor rule [12]. The purpose of this algorithm was to discard as much data as possible by removing data points that were "far" from the decision boundary, in a heuristic fashion, without knowing precisely where the decision boundary lay. However, although the condensed training set discarded many points (one of its best features), and obtained zero misclassification errors when classifying the original full training set, a property referred to as *training-set consistency*, the performance on separate testing data was modest.

In 1972, Wilson [26] proposed an algorithm that did the exact opposite of Hart's algorithm, by discarding the data that were "near" the decision boundary, also in a heuristic manner. This had the effect of "smoothing" the decision boundary. In the pre-processing stage of Wilson's edited nearest neighbor rule, all the data are first classified using the k-NN rule in a leave-one-out setting, and all the data misclassified are then discarded. In the decision stage new data are classified using the 1-NN rule with the resulting (smaller) edited set. As would be expected, although this algorithm does not remove much of the training data, its redeeming feature is that for a large class of problems the asymptotic performance of the edited nearest neighbor rule is difficult to differentiate from that of the optimal Bayes decision rule. In effect, Wilson's editing scheme makes the 1-NN rule perform like the k-NN rule with the optimal value of k. This edited nearest neighbor decision rule has come to be known in the literature as *Wilson editing*, and has left a lasting impact on subsequent instance-based learning algorithms. Indeed, almost all algorithms proposed since 1972 use Wilson editing as one of their initial steps, with the primary goal of lowering the final overall probability of misclassification.

One of the best algorithms in the literature for considerably reducing the size of the training set without degrading its classification performance is the iterative case-filtering algorithm (ICF) of Brighton and Mellish [1]. This algorithm consists of a two-step procedure for discarding data. The first step is Wilson editing, in which the k-NN rule is used to classify the data, in this case with $k = 3$. The second step is an original condensing step that makes use of novel notions of *reachable* sets and *coverage* sets. New query points are then classified using the 1-NN rule with the resulting condensed set.

An omnipresent concern with the k-NN rule has been the selection of the value of k that will yield the lowest probability of misclassification for the problem at hand. Furthermore, in the traditional k-NN rule, once the value of k has been chosen it remains fixed for all subsequent classification decisions. Therefore, in this setting the k-NN rule does not adapt its value of k to either the local density or the spatial distribution of the neighbors of a query point. Proximity graphs provide an approach that not only takes care of these issues, but does so in a fully automatic way without having to tune any parameters. Instead of taking a majority vote from among the k nearest neighbors of a query point X, the proximity graph decision rule takes a majority vote from among *all* the *graph* neighbors of X determined by a suitable proximity graph of the training data. As a concrete example, consider the Delaunay triangulation of the set of points in Figure 2 (left). It is evident that the number of Delaunay neighbors of a point varies greatly depending on the point's location. The leftmost point for example has only three neighbors, whereas the point in the center of the circularly arranged group of points has thirteen neighbors. Similarly, the point in the upper right has eleven Delaunay neighbors, and furthermore these eleven do not even correspond to the eleven nearest neighbors. Proximity graph methods may then replace the k-NN rule in Wilson editing, thus dispensing with the problem of determining a suitable value of k.

Proximity graph methods may also be used for discarding points that are far from the decision boundary, in condensing algorithms. The first condensing algorithm that used proximity graphs employed the Delaunay triangulation [23]. In this algorithm a data point X is first marked if all its Delaunay neighbors belong to the same class as that of X. Then all marked points are deleted. The strength of this condensing method lies in the fact that the 1-NN decision boundary of the original full training set is not changed after the data are deleted, a property referred to as *decision-boundary consistency*. All the discarded data are completely redundant and hence do not play a role in determining the decision boundary of the original larger training set. Initial experiments with Delaunay graph condensing, performed with cervical cancer data collected at McGill University that consisted of two thousand cells in four dimensions [18], revealed that the amount of data discarded was not particularly impressive [24]. An effect of the curse-of-dimensionality is that even in a space of only four dimensions, two thousand points are sufficiently scattered to allow points to have many Delaunay neighbors. Clearly, a point with a greater number of neighbors has a higher probability that one of its neighbors belongs to a different class, especially if it lies near the decision boundary, and thus a smaller probability that it will be discarded. This observation prompted the application of proximity graphs that have fewer edges than the Delaunay triangulation. Furthermore, in order to minimize the distortion of the decision boundary of the entire set it was advocated that the proximity graphs employed should be sub-graphs of the Delaunay triangulation, such as Gabriel graphs and relative neighborhood graphs [24]. It was found experimentally that Gabriel graphs discarded a significantly greater number of data points without degrading the performance, whereas relative neighbor graphs (RNGs) significantly increased the probability of misclassification. For this reason minimum spanning trees (which are sparser than RNGs) were not even tried in those experiments.

The application of Gabriel and Relative Neighborhood graphs to editing as well as condensing methods was experimentally investigated using two data sets (Image data and Vowel data) by Sánchez, Pla, and Ferri [22]. For the Image data the RNG gave the best recognition accuracy if only editing was used, but the Gabriel graph did best when editing was followed by condensing. On the other hand, with the Vowel data the simple 1-NN rule gave the highest accuracy. The data sets used in these experiments were composed of low-dimensional feature vectors: 5 for the Vowel data, and 2 for the Image data. It is thus difficult, from this study, to make conclusions about other data sets in general, and high-dimensional data in particular.

The promising results obtained independently by two very different approaches, namely, the iterative case-filtering (ICF) algorithm of Brighton and Mellish [1] and the Gabriel graph methods [22, 24], provided sufficient motivation to concatenate these approaches into a hybrid algorithm called GSASH that honed the best of both worlds [2-3]. GSASH employed Wilson-type editing but using a Gabriel decision rule, followed by Gabriel neighbor condensing and iterative-case filtering. The final decisions in queries were also made using the Gabriel decision rule.

A word is in order concerning the acronym GSASH appended to the title of this hybrid algorithm. No practically efficient algorithm exists for computing the Gabriel graph of very large training data sets. The fastest algorithm available is essentially the brute-force method that runs in $O(dn^3)$ time, where d is the dimension, although at least one algorithm has been proposed to speed up the average computation time by a constant factor [24]. In order to obtain a truly efficient algorithm one must resort to computing approximate Gabriel graphs with a suitable data structure, in the spirit of SASH, an approximate nearest neighbor method originally proposed by Houle [13]. GSASH is a modification of SASH to handle Gabriel rather than nearest neighbors. It allows the data structure containing the training data to be computed in $O(dn \log n)$ time using $O(dn)$ memory, so that the k approximate Gabriel neighbors of a query point may be computed in $O(dk \log n)$ time.

Experiments with the 25 data sets available at the time in the UCI Repository of Machine Learning Database [16] demonstrated that the Gabriel condensed set, using the approximate Gabriel graph, preserved quite faithfully the original decision boundary based on the exact Gabriel graph [2]. It was also experimentally observed that the ICF algorithm was overly zealous in discarding data, resulting in a slight decrease in recognition accuracy. The Hybrid GSASH algorithm on the other hand tends to incorporate the best of both worlds: preserve the decision boundary, thus maintaining the recognition accuracy, and reduce significantly the storage space required. Nevertheless, the sizes and dimensionalities of the 25 data sets used in the experiments were not particularly large compared to those of the data sets that have been added more recently to the UCI repository. Only two of the data sets used in [2] had more than 4000 patterns; the maximum was 5000, the minimum 150, and the average 944. In addition, only 6 data sets had dimensions greater than 19, the maximum dimension was 69, the minimum 3, and the average 16. Only two data sets had a number of classes greater than 9. Therefore one goal of the present study was to investigate how well proximity graph methods perform with larger data sets in higher dimensions, and how their performance scales with respect to the sparseness of the proximity graphs

utilized. A second goal was to compare proximity-graph methods with the traditional *k*-NN rules and support vector machine (SVM) algorithms, a class of completely different methods for designing classifiers, based on optimization techniques, that also has a history of superlative performance [9, 28]. Support vector machines have received widespread attention in the machine learning literature [5, 25]. Zhang and King [28] compared the performance of support vector machines with methods that employ the Gabriel graph. Indeed, it is conjectured that the Gabriel thinned set contains the support vectors [28].

4 Experiments and Results

The primary goal of this research project was to test the efficacy of the proximity graphs and their scalability with respect to their edge-density, rather than the running time of the algorithms used. For this reason, the *k*-NN, Gabriel, RNG, and MST decision rules were implemented using exact rather than approximate methods such as GSASH. Another drawback of using GSASH is that it puts a constraint on the maximum number of approximate neighbors it can return (to achieve computational efficiency). In contrast, our goal was to investigate the true error rates that the proximity graphs can deliver, as well as the number of neighbors required. For this, an exact approach was needed. In the following, the terms "point" and "instance" are used interchangeably.

For the *k*-NN, Gabriel, RNG, and MST decision rules, the voting neighbors of a query point were first calculated. Then the query point was classified by taking an unweighted majority vote of the voting neighbors' class memberships. Ties were broken arbitrarily by selecting the class with the smallest index value. The distances between pairs of instances were computed using the Hybrid Value Difference Metric (HVDM) described by Wilson and Martinez [27]. This metric allows the handling of both real-valued and categorical attributes, and provides approximate normalization between the different attributes.

The algorithm for the *k*-NN rule simply found the *k* nearest neighbors of the query point, by computing the distance to every point in the data set, and keeping the *k* smallest distances seen so far. No fancy data structure for computing neighbors more efficiently was used. In our implementation the time complexity of finding the *k* nearest neighbors of each point was $O((k+d)n)$, where n is the number of points, and d is the dimensionality. The algorithm for the Gabriel neighbor rule found the Gabriel neighbors of a query point q by trying each point p in the data set as a potential neighbor, and checking that no other point o lies in the sphere with diameter determined by q and p. A speedup for avoiding unnecessary checks [24] was used, but the worst-case time complexity for each query point was still $O(dn^2)$, where n is the number of points in the data set, and d is the dimensionality. The algorithm for the RNG neighbor rule operated exactly like the Gabriel algorithm, except that the distance check was different (lunes were used instead of diametral spheres). The algorithm for the MST neighbor rule computed the minimum spanning tree of the set of training points, into which the query point was first inserted. It then returned the neighbors of

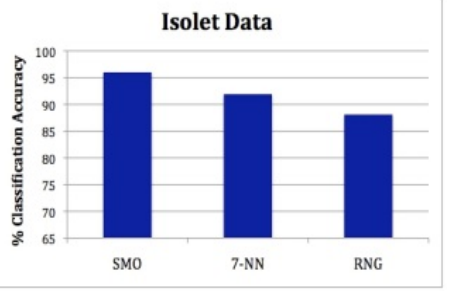
the query point in the resulting MST. Source code used for computing geometric minimum spanning trees in arbitrary dimensions was provided by Giri Narasimhan [17]. In this code, the (un-normalized) Euclidean metric was used as the distance between instances, instead of HVDM.

The algorithms for the *k*-NN, Gabriel, and RNG neighbor rules were implemented in C++. The imported geometric MST code was also written in C++. For the support-vector machine classifier, the Sequential Minimal Optimization (SMO) [19] algorithm from the off-the-shelf Weka data mining package [29] was used. The algorithms were tested on a 64-bit Linux machine with a 2.26GHz CPU and 3GB of RAM. No editing or condensing was done in the experiments presented below, unless stated otherwise. The experiments used four data sets, including three from the UCI Machine Learning Repository [10] (Isolet, Dermatology, and Segmentation).

Isolet Data

The Isolet speech data set has 26 classes (the spoken name of each letter of the alphabet), 6238 training instances, and 1559 testing instances, each with 617 real-valued features consisting of spectral coefficients, contour features, sonorant features, pre-sonorant features, and post-sonorant features. The suggested split between training and testing subsets as indicated by the data set was used, rather than performing cross-validation. This data set is truly massive in terms of both size and dimensionality.

The accuracy (rate of correct classification) of the *k*-NN rule as a function of *k* for the Isolet data is shown in Figure 3. The maximum accuracy was 91.9% for *k* = 7. The RNG accuracy was slightly lower at 88.1%. The SMO classifier (with parameters: complexity = 1.5, polynomial kernel exponent = 1.5) reached an accuracy of 96.0%, surpassing the other instance-based methods. These results are summarized in the chart on the right.



No results were obtained with the Gabriel and MST classifiers because the algorithms were too slow to run to completion. In the MST classifier, every test instance required re-computing the MST. In the Gabriel classifier, there were simply too many neighbors. This underscores the fact that despite their similar theoretical worst-case time complexity, the actual running time required by the RNG and Gabriel algorithms can be vastly different. In this data set, each point has very many Gabriel neighbors, but only a few RNG neighbors. Verifying that two points are Gabriel (or RNG) neighbors requires checking that no other point lies in their diametral sphere (or lune). This test uses O(*dn*) time, an expensive operation in this context. For the RNG, most pairs of points are not neighbors. Thus most neighbors are discarded by promptly finding a counterexample. On the other hand, for the Gabriel graph most pairs of nodes are neighbors. Thus, the linear-time check will be required for almost every pair of nodes.

From the theoretical point of view Devroye [6] has shown that for most probability density functions governing the data, the asymptotic expected number of edges in the Gabriel graph is essentially $2^{d-1}n$, where n is the number of instances in the training set, and thus the number of Gabriel neighbors of a point grows exponentially in terms of the dimension. To obtain some practical intuition for this curse of dimensionality, the Gabriel graph was computed for only 35 test instances, and the average number of neighbors already reached 4588. This result provided empirical evidence to support the hypothesis that the Gabriel graph, like its denser parent Delaunay triangulation, quickly becomes extremely dense in high-dimensional spaces. Furthermore, once the number of neighbors reaches a significant fraction of the entire data set, the resulting decisions are primarily based on the prior probability of the classes, effectively ignoring much of the feature information, and resulting in a natural degradation of performance.

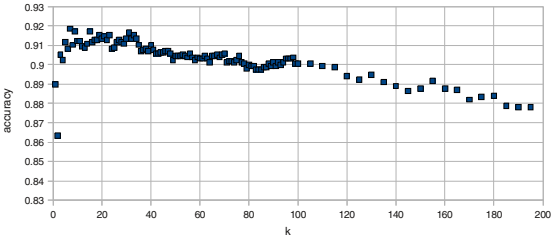


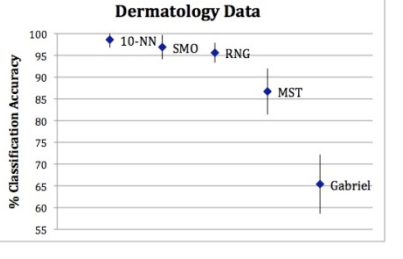
Fig. 3. The classification accuracy of the k-NN rule with the Isolet data

Dermatology Data

The Dermatology data set has six classes (skin diseases) and 366 instances, each with 33 categorical attributes and one real-valued attribute. The total dimensionality of the feature space is therefore 34. Eight instances contained missing data, and were removed from the data set. In these experiments, randomly selected subsets consisting of 20% of the data were used for the testing sets, leaving the remaining 80% for training. All the accuracy results were averaged over 10 trials.

The classification accuracy of the k-NN rule as a function of k for the Dermatology data is shown in Figure 4. The k-NN rule does very well, achieving a maximum value of 98.6% for both $k = 10$ and $k = 15$, although it does pretty well for all the values of k tried. The following table and graph summarize the results, where the error bars indicate ± one standard deviation.

Classifier	Mean Accuracy (%)	Standard Deviation
10-NN	98.6	1.8
SMO	96.9	2.8
RNG	95.6	2.3
MST	86.7	5.3
Gabriel	65.4	6.8



The SMO algorithm performs slightly worse than 10-NN, obtaining a top mean accuracy of 96.9% (for parameters: complexity = 2.0, polynomial kernel exponent = 3.0). However, as the error bars show, the results with 10-NN, SMO, and RNG are not significantly different from each other. On the other hand, these three classifiers yield results significantly better than those with the MST, which in turn are significantly better than those obtained with the Gabriel graph.

Interestingly, this is the only data set for which an instance-based method appears to do better than the SMO (although not significantly). The *k*-NN rule is significantly better than all three proximity graph methods. The MST appears to be too sparse to capture the requisite discrimination information.

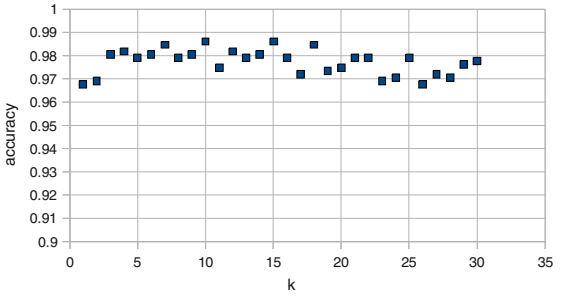


Fig. 4. The classification accuracy for the *k*-NN rule with the Dermatology data

Since the number of Gabriel graph neighbors grows so fast with *d*, it seemed conceivable that capping might still capture local information better than the *k*-NN rule. Accordingly, an experiment was performed that looked at only the *k* closest Gabriel neighbors. The mean classification accuracy for this decision rule with the Dermatology data is shown in Figure 5. For all values of *k* up to 30, the mean accuracy is better than all three proximity graph methods, and for $k = 22$ the value of 98.6% matches the accuracy of *k*-NN for $k = 10$ and $k = 15$. These results however do not appear to provide any additional insight into the workings of the Gabriel graph, since the values of *k* that yield results comparable to the *k*-NN rule are greater, and therefore the *k* closest Gabriel neighbors may in fact include the *k* nearest neighbors.

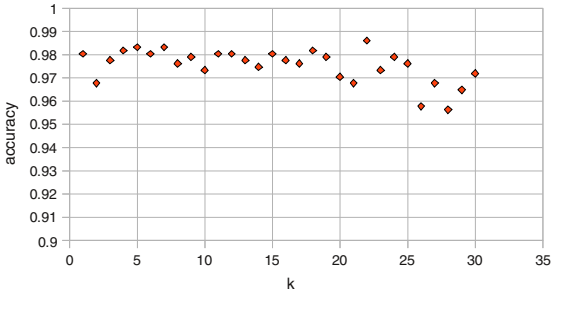


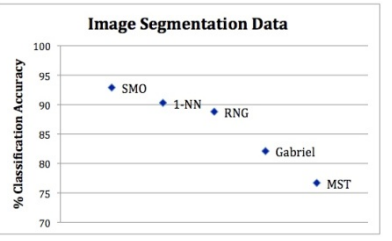
Fig. 5. The mean classification accuracy for the *closest k Gabriel neighbors* rule with the Dermatology data

Image Segmentation Data

The Image Segmentation data set has 7 classes, 210 train instances, and 2100 test instances, each with 18 real-valued attributes. We used the suggested train/test split given by the data set, rather than performing cross-validation (see Figure 6).

The 1-NN rule does best, and the SMO algorithm (with complexity = 3.0, polynomial kernel exponent = 5.0) surpasses all methods (see table and chart below). Again, the RNG is superior to the Gabriel graph, and the MST does not perform as well.

Classifier	Classification Accuracy (%)
SMO	92.9
1-NN	90.3
RNG	88.8
Gabriel	82.1
MST	76.7



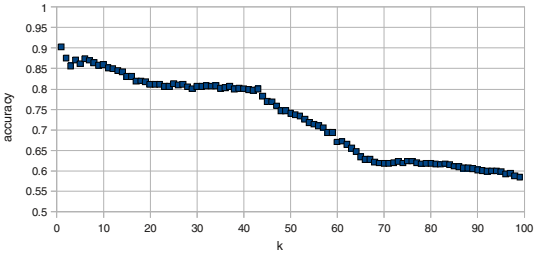


Fig. 6. The classification accuracy for the *k*-NN rule with the Image Segmentation data

Figure 7 shows the results, and the amount of storage used (as a fraction of the initial training set) for different k used in Wilson editing. Storage was reduced by 15%, at the expense of accuracy. A top accuracy of 85.4% was achieved with $k = 6$.

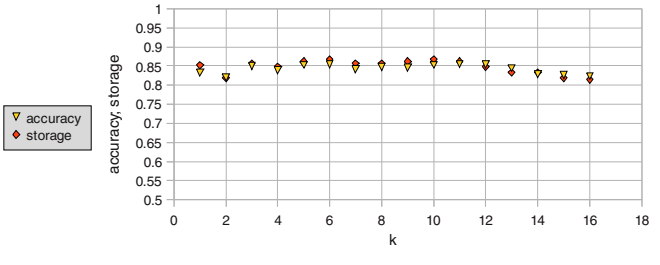


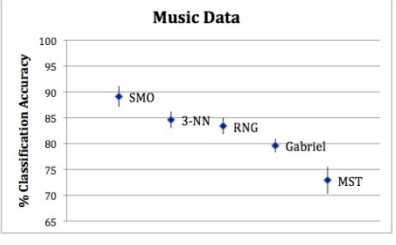
Fig. 7. The classification accuracy and storage used when applying Wilson editing to the Image Segmentation data

Music Data

The Music data set has 2 classes (Western and Non-Western music) and 1641 instances, each with 41 real-valued attributes, of which 1 was constant [11]. The total dimensionality of the feature space is therefore 40. In the experiments, 20% of the data was randomly set aside as the testing set, and the remaining 80% was used for training. All accuracy results were averaged over 10 trials.

The classification accuracy of the *k*-NN rule as a function of *k* for the Music data is shown in Figure 8. The *k*-NN rule achieves a top mean accuracy of 84.6% for $k = 3$. The SMO support-vector machine classifier achieves an accuracy of 86.5% according to Gomez and Herrera [11], but they reported no standard deviations. Replicating their experiment using the same parameters (complexity = 1.5, polynomial kernel exponent = 1.5), a fairly significantly higher mean accuracy of 89.1% was obtained. The results are summarized in the following table and graph:

Classifier	Mean Accuracy (%)	Standard Deviation
SMO	89.1	2.0
3-NN	84.6	1.6
RNG	83.4	1.6
Gabriel	79.6	1.3
MST	72.9	2.6



The RNG results are significantly better than those of the Gabriel graph, which are in turn significantly better than the MST results. It is worth noting that for the Gabriel rule, the test instances had an average number of 303 voting neighbors, further illustrating that the Gabriel graph becomes dense in high dimensions. With the RNG classifier, the average number of neighbors was only 4.2. The 3-NN rule is superior to all three proximity graph methods, although not statistically significantly better than the RNG. The SMO classifier yields results statistically significantly better than all the other classifiers, with a mean accuracy of 89.1%.

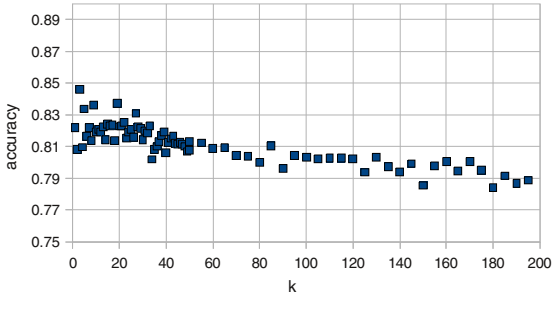


Fig. 8. The classification accuracy for the *k*-NN rule with the Music data

5 Concluding Remarks

In theory it is known that for almost all probability density functions governing the training data, the expected number of edges in the Gabriel graph grows exponentially with the dimensionality of the space in which the data are embedded [6]. The results of the experiments obtained here provide empirical confirmation of the theory.

In previous research with data sets having small-dimensionality d, it has been frequently found that the Gabriel graph performed best among the proximity graph methods. It was hypothesized that with higher d a sparser graph would give better results. The empirical results obtained here confirm this hypothesis. In all the experiments the relative neighbor graph (RNG) performed significantly better than the Gabriel graph, probably as a result of the explosive growth of the number of Gabriel graph neighbors as d increases. To determine the degree to which the sparseness of proximity graphs can help, experiments were performed with the (connected) proximity graph that has the fewest possible number of edges, the minimum spanning tree (MST). The experimental results showed conclusively that the MST is too sparse to capture sufficient discrimination information to achieve good performance, and thus it appears to be useless for this application in instance-based learning, thus settling a long-standing speculation.

The traditional k-NN decision rule has a long recorded history of yielding high classification accuracy in practice. Its drawbacks of high memory and computation requirements motivated the introduction of proximity graph methods for reducing the size of the training data. For low values of d it appeared that little would be lost in terms of performance, by resorting to proximity graphs. However, the results obtained here with large d clearly indicate that k-NN methods are superior, thus challenging proximity graph methods, if performance is the only issue. In all the experiments the k-NN rule performed statistically significantly better than the Gabriel graph, and although it also typically achieved better results than the RNG these were not statistically significant.

It is known that in theory the k-NN rules are asymptotically Bayes optimal, and that therefore there should not exist any other classifier that gives strictly better average classification accuracy [7-8]. Nevertheless, as the experiments reported here demonstrate, even for large real-world data sets, the SMO support vector machine yields statistically significantly better results than k-NN. This suggests that in practice SVMs should be the classifiers of choice, other things being equal. The drawback of traditional implementations of SVMs is high computation in the design stage of the classifier, although the sequential minimal optimization (SMO) version offers improvements in this regard. Their advantage is fast classification of new query data. Proximity graph decision rules on the other hand are very slow for this task unless approximate methods such as GSASH are used. Therefore the results of this study suggest that the most fruitful approach for classification is to use SMO in order to obtain the best classification performance, if the computation time spent on classifier design can be significantly reduced. Furthermore, since this computation time depends to a large extent on the size of the training data, it is worthwhile reducing the number of instances in a computationally

inexpensive manner before subjecting them to an SMO support vector machine training algorithm. This appears to be a more appropriate role for proximity graphs to play than to act as neighbor filters in decision rules.

References

1. Brighton, H., Mellish, C.S.: Advances in Instance Selection for Instance Based Learning Algorithms. Data Mining and Knowledge Discovery 6, 153–172 (2002)
2. Bhattacharya, B., Mukherjee, K., Toussaint, G.T.: Geometric Decision Rules for Instance-Based Learning Problems. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 60–69. Springer, Heidelberg (2005)
3. Bhattacharya, B., Mukherjee, K., Toussaint, G.T.: Geometric Decision Rules for High Dimensions. In: Proc. 55th Session of the International Statistics Institute, Sydney, Australia, April 5-12 (2005)
4. Cover, T.M., Hart, P.E.: Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory 13, 21–27 (1967)
5. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning 20, September 1-25 (1995)
6. Devroye, L.: The Exptected Size of Some Graphs in Computational Geometry. Computers and Mathematics with Applications 15, 53–64 (1988)
7. Devroye, L.: On the Inequality of Cover and Hart in Nearest Neighbor Discrimination. IEEE Transactions on Pattern Analysis and Machine Intelligence 3, 75–78 (1981)
8. Devroye, L., Györfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition. Springer (1996)
9. Duan, K.-B., Keerthi, S.S.: Which Is the Best Multiclass SVM Method? An Empirical Study. In: Oza, N.C., Polikar, R., Kittler, J., Roli, F. (eds.) MCS 2005. LNCS, vol. 3541, pp. 278–285. Springer, Heidelberg (2005)
10. Frank, A., Asuncion, A.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA (2010), http://archive.ics.uci.edu/ml
11. Gomez, E., Herrera, P.: Comparative Analysis of Music Recordings from Western and Non-Western traditions by Automatic Tonal Feature Extraction. Empirical Musicology Review 3 (2008)
12. Hart, P.E.: The Condensed Nearest Neighbor Rule. IEEE Transactions on In-formation Theory 14, 515–516 (1968)
13. Houle, M.: SASH: A Spatial Approximation Sample Hierarchy for Similarity Search. Tech. Report RT-0517, IBM Tokyo Research Lab (2003)
14. Jaromczyk, J.W., Toussaint, G.T.: Relative Neighborhood Graphs and their Relatives. Proceedings of the IEEE 80, 1502–1517 (1992)
15. 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)
16. Merz, C.J., Murphy, P.M.: UCI Repository of Machine Learning Database, Department of Information and Computer Science, University of California, Internet, http://www.ics.uci.edu/mlearn/MLRepository.html
17. Narasimhan, G., Zhu, J., Zachariasen, M.: Experiments with Computing Geometric Minimum Spanning Trees. In: Proceedings of Algorithm Engineering and Experiments (ALENEX 2000). LNCS, pp. 183–196. Springer, Heidelberg (2000)

18. Oliver, L.H., Poulsen, R.S., Toussaint, G.T.: Estimating False Positive and False Negative Error Rates in Cervical Cell Classification. J. Histochemistry and Cytochemistry 25, 696–701 (1977)
19. Platt, J.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In: Schoelkopf, B., et al. (eds.) Advances in Kernel Methods - Support Vector Learning. MIT Press (1988)
20. Toussaint, G.T.: Geometric Proximity Graphs for Improving Nearest Neighbor Methods in Instance-Based Learning and Data Mining. International J. Computational Geometry and Applications 15, 101–150 (2005)
21. Toussaint, G.T.: The Relative Neighborhood Graph of a Finite Planar Set. Pattern Recognition 12, 261–268 (1980)
22. Sánchez, J.S., Pla, F., Ferri, F.J.: Prototype Selection for the Nearest Neighbor Rule through Proximity Graphs. Pattern Recognition Letters 18, 507–513 (1997)
23. Toussaint, G.T., Poulsen, R.S.: Some New Algorithms and Software Implementation Methods for Pattern Recognition Research. In: Proc. Third International Computer Software and Applications Conference, pp. 55–63. IEEE Computer Society (1979)
24. Toussaint, G.T., Bhattacharya, B.K., Poulsen, R.S.: The Application of Voronoi Diagrams to Nonparametric Decision Rules. In: Proc. Computer Science and Statistics: 16th Symposium on the Interface, pp. 97–108. North-Holland, Amsterdam (1985)
25. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)
26. Wilson, D.L.: Asymptotic Properties of Nearest Neighbor Rules Using Edited-Data. IEEE Transactions on Systems, Man, and Cybernetics 2, 408–421 (1973)
27. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. Machine Learning 38, 257–286 (2000)
28. Zhang, W., King, I.: A Study of the Relationship Between Support Vector Machine and Gabriel Graph. In: Proc. IEEE International Joint Conference on Neural Networks, IJCNN 2002, Honolulu, vol. 1, pp. 239–244 (2002)
29. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11 (2009)

Semi Supervised Clustering: A Pareto Approach

Javid Ebrahimi and Mohammad Saniee Abadeh

Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran
{j.ebrahimi,saniee}@modares.ac.ir

Abstract. In this paper we present a Pareto based multi objective algorithm for semi supervised clustering (PSC). Semi-supervised clustering uses a small amount of supervised data known as constraints, to assist unsupervised learning. Instead of modifying the clustering objective function, we add another objective function to satisfy specified constraints. We use a lexicographically ordered cluster assignment step to direct the search and a Pareto based multi objective evolutionary algorithm to maintain diversity in the population. Two objectives are considered: one that minimizes the intra cluster variance and another that minimizes the number of constraint violations. Experiments show the superiority of the method over a greedy algorithm (PCK-means) and a genetic algorithm (COP-HGA).

Keywords: semi supervised clustering, multi objective evolutionary algorithms, none dominance ranking, must link, cannot link.

1 Introduction

Clustering is the task of meaningful grouping of similar data points in the same group while separating dissimilar points in distinctive groups [5-12]. The unsupervised nature of clustering makes it hard to evaluate the output because multiple interpretations make different partitionings acceptable [1]. But in most real world application, there is some background knowledge about the data, which could guide the clustering process and pave the way for a unique interpretation. The background knowledge could be formulated as either labels or pair wise constraints, which are more naturally available; because the user may not know the exact categories of the data but can, provide feedback i.e. denote which instances must be grouped together and which cannot [4]. Numerous works have shown significant improvements to clustering accuracy by employing constraints [2-4, 18-19]. Some of the constraints are inherent properties of the data like must link or cannot link constraints while some of them are imposed by application e.g. size constraints in an energy efficient sensor network [4]. K-means is the most famous clustering algorithm that suffers from sub-optimality problem and can be trapped in local optima. Depending on the choice of initial centroids, K-means can give different partitionings [6]; moreover constraints may split the search space of the optimal clustering solution into pieces, thus significantly complicate the search task [12]. Bio inspired meta-heuristic algorithms,

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 237–251, 2012.

can explore solution space exhaustively and are therefore good candidates to find globally optimum solution for clustering problem [7]. Numerous Evolutionary and stigmergic algorithms including particle swarm optimization (PSO) [8, 16], ant colony optimization (ACO) [9] and differential evolution (DE) [5, 10]; among many others have been employed to tackle the sub-optimality problem of classical clustering algorithms. Genetic algorithms (GA) have been successfully applied to clustering on several problems [6, 11-12, 25]. The advantage of these algorithms is their ability to deal with local optima by mutating and recombining several candidate solutions at the same time [7]. Suitable guided or unguided evolutionary operators are defined for any specific encoding of the problem while a fitness function is being optimized [25].Fitness functions are based on an objective function that measures the goodness of a partitioning; several measures have been used in literature including Minimum squared error, silhouette coefficient, Davies-Bouldin [14] etc. These are (internal) validity functions that prefer a minimized intra cluster variation or maximized inter cluster distance. However, when the ground truth of the data exists, and is partially used to guide the process, clustering output is validated by labels through semi supervised (external) validity metrics, e.g. Rand Index, normalized mutual information [13] and fitness functions should be changed accordingly. PCK-means [3] is a state of the art semi supervised algorithm that greedily optimizes a modified version of K-means objective function. Our contribution is in solving semi supervised clustering algorithm by a multi objective evolutionary algorithm and improving the results of traditional PCK-means [3] and GA based COP-HGA [12]. We employ a lexicographical-type cluster assignment step and NSGAII [21] known for its maintenance of quality and diversity. Our algorithm has a desirable property that the clustering objective function is not modified and no subsequent weighting is needed.

2 Related Works

2.1 Clustering

Problem of clustering has already been investigated extensively in machine learning literature [1, 28-30]. Clustering algorithms could roughly be divided to two major categories: partitional and hierarchical algorithms, e.g. K-means, and single link respectively [1]. In partitional algorithms a global objective function is optimized throughout the clustering process, while in hierarchical approaches clusters are built incrementally by either dividing or agglomerating cluster(s) in the previous iterations [1, 5]. Other approaches to clustering include graph based methods [26-27, 29-30] and model based e.g. Expectation Maximization (EM) methods [28]. Many researchers have used evolutionary algorithms for clustering in different domains; e.g. document clustering [5, 8, 11], Image segmentation [10, 15-16], social networks [26- 27] and gene expression data [17].

2.2 Semi Supervised Clustering

Recently researchers have included partial supervision in clustering process to guide the algorithm to a better result. Semi supervised (constrained) clustering employs a small amount of data to help clustering task. Utilization of supervised data in this task is done in two primary ways. 1) constraint-based methods that guide the clustering algorithm towards a better partitioning of the data, and 2) distance-function learning methods that adapt to the underlying similarity measure used by the clustering algorithm [19].In the first approach, the clustering objective function is modified to conform to constraints set by the user [2].While in the second approach, a distance metric is first trained on the labeled data set and is then used to cluster unlabeled data set [19]. Cop-K-means [18] is a modified version of k-means that uses backtracking to satisfy all constraints. When there is a large number of a constraint, this rigid clustering make the algorithm more biased toward constrained data points. A more relaxed version of constraint conformance is PCK-means [2] where the K-means objective function is changed to one that considers a penalty for constraint violation, it too suffers from falling in local optima traps , but does not need backtracking. In COP-HGA [12] a genetic algorithm was used that optimizes the same objective function as in PCK-means. Their algorithm was based on an evolutionary strategy operator that guided the clustering process. In [20] a Cop-K-means-like probabilistic genetic algorithm is introduced that builds a Bayesian network based on given constraints; all clustering solutions are sampled from that network, guaranteeing all constraints. Assuming that instances are independent their algorithm estimates the density function of the population. We propose a new approach for semi supervised clustering where the objectives are regarded independently. Like PCK-means, it allows constraints to be violated, but the difference is that instead of modifying the K-means objective function, we add another one and treat the two objectives independently. The objectives are: one that minimizes the intra cluster variance and another that minimizes the number of constraint violations; they are optimized in a multi objective setting. There exists three main approaches for multi objective optimization[22]: 1) transforming objectives into one weighted single one, in this approach objectives should be normalized, and weighted by the user which might be far from trivial. 2) Lexicographical approaches where objectives are prioritized by an expert and are optimized accordingly, which may not always be applicable 3) The other approach that needs the least amount of tuning is through finding a set of Pareto points; in which the final solution contains a number of Pareto-optimal solutions none of which can be improved on any objective without degrading it in another.

3 Problem Definition

Clustering aims to classify a set of unlabeled instances into groups such that instances in the same group are more similar to each other, while they are more different in different groups. The main characteristic of most partitional clustering algorithms is that they use a global criterion function whose optimization drives the entire clustering process [5]. For these algorithms, the clustering problem can be stated as

computing a clustering solution such that the value of a particular function is optimized. Below is the objective function of K-means.

$$\sum_{j=1}^{k} \sum_{d_i \in c_j} \left\| x - \mu_j \right\|^2 \tag{1}$$

Where *d*i is the *ith* data in *jth* cluster and μ_i is the centroid of *ith* cluster. For the constrained clustering version, we follow the formulation in [3]. Let *M* be the set of *must link* pairs such that if *(di, dj)* $\in$ *M*, then d*i* and *dj* must be assigned to the same cluster and let *C* be the set of *cannot link* pairs such that if *(di, dj)* $\in$ *C*, then d*i* and *dj* must be assigned to different clusters. Let *lj* be the cluster assignment of a point *di*, where *lj* $\in$ *{1, 2.., k}*, then the above objective function is modified as:

$$\sum_{j=1}^{k} \sum_{d_i \in c_j} \left\| x - \mu_j \right\|^2 + \lambda\Big(\sum_{(d_i, d_j) \in M} \mathrm{I}\left[l_i \neq l_j \right] + \sum_{(d_i, d_j) \in C} \mathrm{I}\left[l_i = l_j \right] \Big) \tag{2}$$

Where I is the indicator function, Here we assume that the cost of violating *must link* and *cannot link* are the same, and λ is a parameter set by user that determines the weight of the penalty added to the within cluster variation objective. λ works as a tuning knob, the higher it gets, the more emphasis on constraints and the lower it gets, the more emphasis would be on intra cluster distance; this weighting scheme differs from one data set to another and may need exhaustive tunings. PCK-means is similar to K-means and alternates between two assignment and update steps. In the beginning, based on *must link* constraints, a set of initial centroids are generated, in assignment step, each data point *d* is assigned to the best cluster h^* as indicated in (3). Based on the new partitioning, centroids are update and the process continues till it reaches a local minimum.

$$h^* = \arg\min_h \left\| x - \mu_h \right\|^2 + \lambda\Big(\sum_{(d_i, d) \in M} \mathrm{I}\left[l_i \neq h \right] + \sum_{(d_i, d) \in C} \mathrm{I}\left[l_i = h \right] \Big) \tag{3}$$

However, we can't easily treat the objective function in (2) as a multi objective optimization problem, since data points are added incrementally to clusters and an assignment, effects future decisions. Therefore the two objectives are not separable until the last data point is assigned to a cluster, all clusters are formed and we have sum of intra cluster distance and total number of constraint violations.

4 Clustering Algorithm

Genetic algorithms have been successfully applied for clustering on several problems [6, 11-12, 17].One major issue in these algorithms is the choice of chromosome encoding. We have used integer encoding where a chromosome is an integer vector of *N* positions, where *N* is the number of data points that can take values from 1 to *k*, where k is the number of clusters. As in every genetic based algorithm, at first an initial

population is generated and evolves through an elitist strategy where the fittest individuals among parents and Offspring replace the previous population and the process goes on until convergence. Selection and replacement is done based on ranking scheme of NSGAII. In the next subsection we describe the assignment and update steps that are performed on every individual. Without this assignment step, the search would be impossible, since the search space is too vast for an unguided GA to explore.

4.1 Centroid Update and Assignment

Centroids of the clusters are updated by averaging on data points assigned to each cluster.

$$\mu_j = \frac{\sum_{d_i \in c_j} d_i}{\left| c_j \right|} \tag{4}$$

Instances are reassigned to a cluster in lexicographical steps: 1) they are assigned to a cluster where the number of violated instance-level constraints is minimal:

$$h^* = \arg\min_h \sum_{(d_i, d) \in M} \mathrm{I}\left[l_i \neq h\right] + \sum_{(d_i, d) \in C} \mathrm{I}\left[l_i = h\right] \tag{5}$$

If there is a tie 2) they are assigned to the one which has the closest centroid:

$$h^{**} = \arg\min_{h^*} \left\| x - \mu_{h^*} \right\|^2 \tag{6}$$

The reason for the name is the resemblance between the step and lexicographical optimization where the two objectives are optimized one by one. Finally by summing over the intra cluster distance and constraint violation of all data points, the two objective functions for an individual is calculated.

$$\sum_{(d_i, d_j) \in M} \mathrm{I}\left[l_i \neq l_j\right] + \sum_{(d_i, d_j) \in C} \mathrm{I}\left[l_i = l_j\right] \tag{7}$$

$$\sum_{j=1}^{k} \sum_{d_i \in c_j} \left\| x - \mu_j \right\|^2 \tag{8}$$

These two measures help us explore the search space; using NSGAII ranking scheme we maintain solutions from the most compact partitioning (7) to the most constraint bound ones (8). By using Pareto fronts we are more likely to find the optimum as opposed to when we directly minimize the single objective function (2). In other words by lexicographical assignments of data points, the search process is directed and by maintaining useful solutions in diverse Pareto fronts, the exploited region is explored more effectively. Furthermore we are now exempt of setting the best value of λ for each data set. Figure1.

4.2 Multi Objective Evolutionary Algorithms

There are many applications in which there may be several objectives to be optimized simultaneously. Often It is the case that there is not a single point in solution space that optimizes all objectives at the same time so instead we look for non-dominated solutions (solutions that are not dominated in every objective by another solution) [22]. The multi objective optimization is formally stated as follows

Find the vector $X^* = \left[x_1^*, x_2^*, ..., x_n^*\right]^T$ to optimize

$$F(X) = \left[f_1(X), f_2(X), ..., f_k(X)\right]^T \tag{9}$$

Subject to *m* inequality constraints

$$gi(X) \leq 0 \quad , \quad \text{i} = 1 \text{ to m} \tag{10}$$

And *p* equality constraints

$$h_j(X) = 0 \quad , \quad \text{j} = 1 \text{ to p} \tag{11}$$

Where $X^* \in \Re^n$ is the vector of decision variables, and $F(X) \in \Re^k$ is the vector of objective functions which each of the indices is either minimized or maximized. (Without loss of generality, it is assumed that all objective functions are to be minimized.). A decision vector X^* is said to Pareto optimal if and only if there is no X that dominates X^* i.e.

$$\forall i \in \{1,2,...,k\}\ f_i(X^*) \leq f_i(X) \ \wedge \ \exists j \in \{1,2,...,k\}\ \ f_j(X^*) < f_j(X) \tag{12}$$

NSGAII [21] is a state of the art multi objective evolutionary algorithm that has several desirable properties that are briefly discussed. Firstly, In NSGAII during selection and replacement phases, individuals are sorted based on Pareto dominance, in which individuals are divided into different ranks. The first rank includes individuals that are not dominated by any other individual; the second rank includes those individuals only dominated by the first rank members and so forth. Therefore two remote individuals may have equal ranks because they get dominated by the other in one criterion. This is useful in our problem since we don't discriminate among objectives. Secondly, NSGAII is pro-diversity; the crowding distance mechanism that is used in its sorting individuals in the same rank, gives priority to individuals in more sparse regions in order to maintain a more scattered set of individuals in population. Diversity for its sake is not an issue; instead we need to maintain appropriate diversity [24], we posit that modeling semi supervised clustering as a multi objective problem provides us useful population diversity that will allow us for more effective exploration of the search space. Below is a summary of PSC algorithm.

Algorithm: PSC

Input: Set of data points D

Set of *must link* constraints, M

Set of *cannot link* constraints, C

Number of clusters k

Output: Pareto optimal solutions such that number of violated constraints and intra cluster distance are minimized.
Method:

1 Initialize population:

1a. randomly assign data points *D* to a cluster {1, 2... *k*} in *integer encoding* format

Repeat until convergence

2a.**Selection:** use tournament selection based on *Pareto dominance* on objectives (7) and (8)
2b.**Evolutionary operators:** perform single point crossover (probability of 0.8) and bit flipping mutation (probability of 0.01).

3 Perform on each individual:

3a. **Centroid update:** set the mean of the data points *D* of each cluster as the centroid μ of that cluster *c*.

$$\mu_j = \frac{\sum_{d_i \in c_j} d_i}{\left| c_j \right|}$$

3b. **Lexicographical-type assignment**: assign each data point *d* to the cluster *h** that violates the least number of constraints, if there is a tie; assign it to the closest cluster.

$$h^* = \arg\min_h \sum_{(d_i, d) \in M} \mathrm{I}\left[l_i \neq h\right] + \sum_{(d_i, d) \in C} \mathrm{I}\left[l_i = h\right]$$

If *Exists_tie*

$$h^{**} = \arg\min_{h^*} \left\| x - \mu_{h^*} \right\|^2$$

4. **Replacement:** use *Pareto dominance* based on objectives (7) and (8)

Fig. 1. PSC Algorithm

5 Experiments

For every data set, we experimented on 100, 200, 300, 400 and 500 constraints. For each of these numbers, 10 random sets were generated that each were run 10 times independently and finally their average results were reported. Thus we ran the semi supervised clustering algorithms 500 times for each data set. K-means is an unsupervised algorithm and its accuracy does not change by the number of constraints, so it was run

100 times for each data set and its average was reported. Constraints were generated through randomly selecting two instances from data sets and checking their labels. If their labels were the same, a *must link* otherwise *cannot link* constraint was generated. The probabilities of crossover and bit flip mutation were set to 0.8 and 0.01 respectively. The population size and maximum generations were set 100 and 50 respectively.

5.1 Data Sets

We tested our algorithm on one UCI [31] data set namely, glass and three text data sets derived from 20 Newsgroups data set; News-Similar-3 which has 3 newsgroups on similar topics (comp.graphics, comp.os.ms-windows, comp.windows.x), News-Related-3 that consists of 3 newsgroups on related topics (talk.politics.misc, talk.politics.guns, and talk.politics.mideast) and News-Different-3 that has articles in 3 newsgroups that cover different topics (alt.atheism, rec.sport.baseball, sci.space).Text data sets are available from repository of information on semi-supervised clustering, [32]. See Table I for a description of data sets.

5.2 Clustering Evaluation Measure

We use two external validity measures, Rand index.

$$RandIndex = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

Where a true positive (TP) decision assigns two similar data to the same cluster, a true-negative (TN) decision assigns two dissimilar data to different clusters. A false-positive (FP) decision assigns two dissimilar data to the same cluster. A false-negative (FN) decision assigns two similar data to different clusters [17].

Table 1. Data sets descriptions

Data set	*#Instances*	*#Features*	*#Classes*
Glass	214	9	7
News-Similar	288	3225	3
News-Same	295	1864	3
News-Different	300	3251	3

5.3 Analysis

It can be seen from Figures 2-5 that virtually in every comparison, our algorithm (PSC) is superior to PCK-means [2] and genetically semi supervised clustering COP-HGA [12]; more evidently in text data sets which are more complex, it results in significantly better results. We can see as the number of constraints increase the significance of the improvement decreases, since the more constraints are available the more supervised the methods get similar results are obtained by all three algorithms.

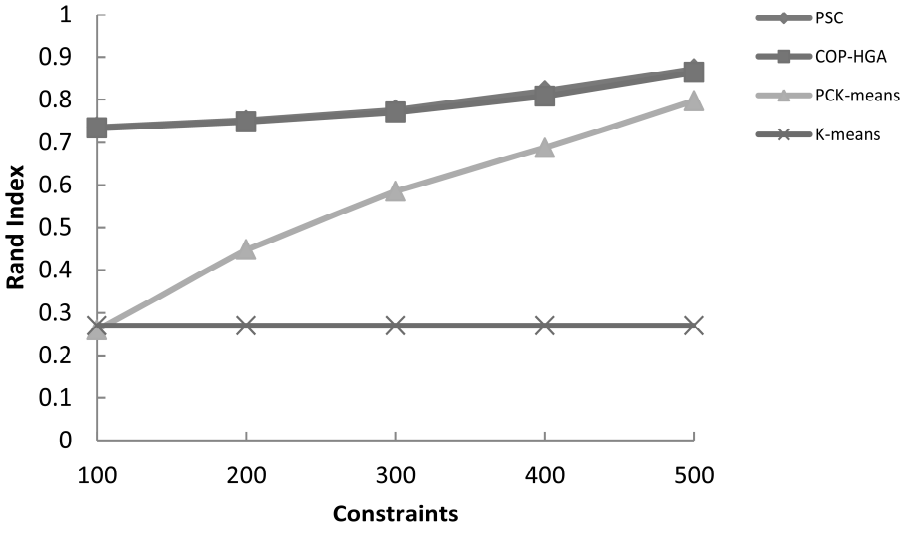


Fig. 2. Glass data set

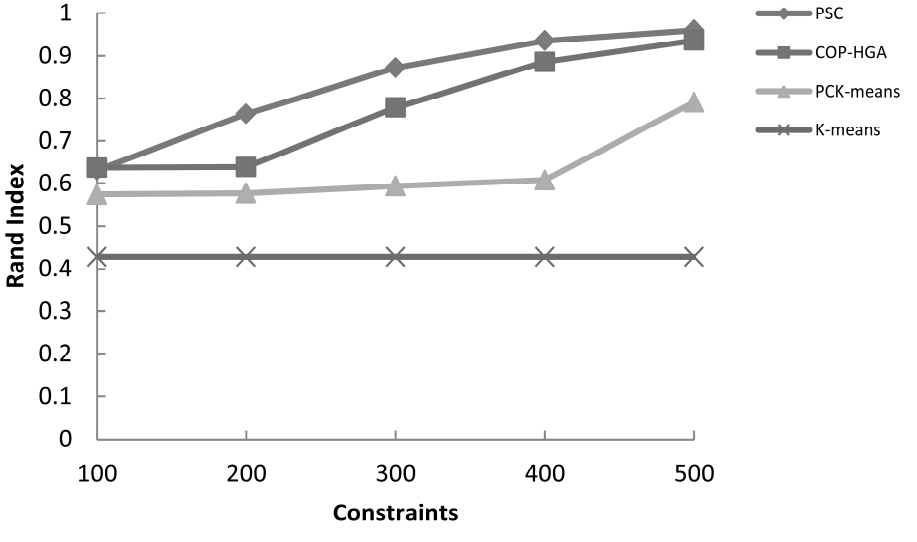


Fig. 3. News-different data set

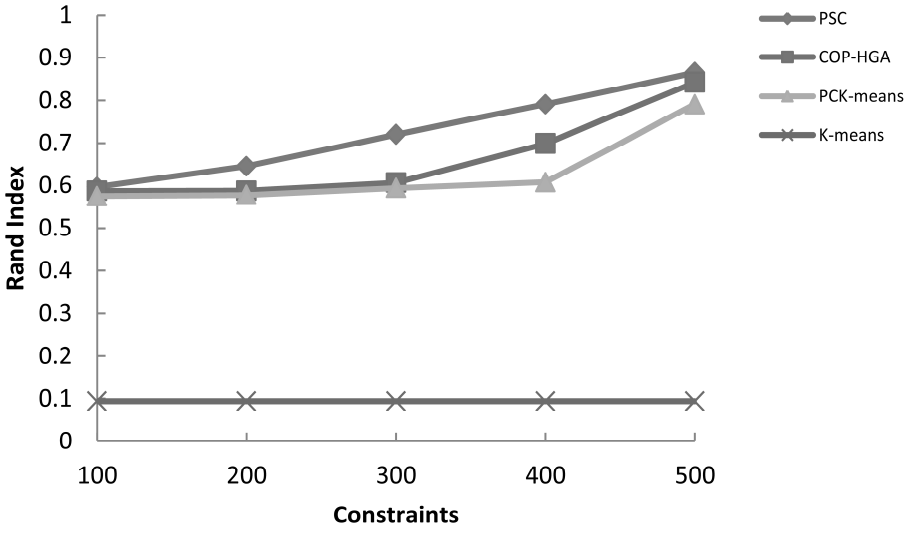


Fig. 4. News-related data set

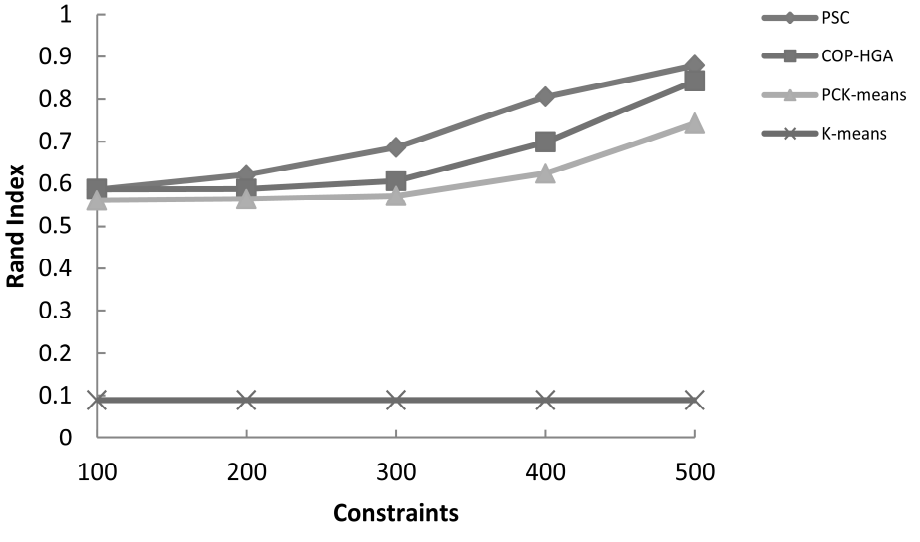


Fig. 5. News-similar data set

As was explained in the previous section, maintaining diverse population is the reason why the algorithm works better. To furthermore demonstrate it, we use population distribution diagrams of the two evolutionary algorithms, PSC and COP-HGA based on two variables: number of violated constraints and intra cluster distance, Figures 6-7. To save space, we have only shown the final distribution for the data set News-related when constrained by a random 400, 500 constraints. The squares show the population of single weighted genetic algorithm (COP-HGA) and the Diamonds show the multi objective (PSC) population.

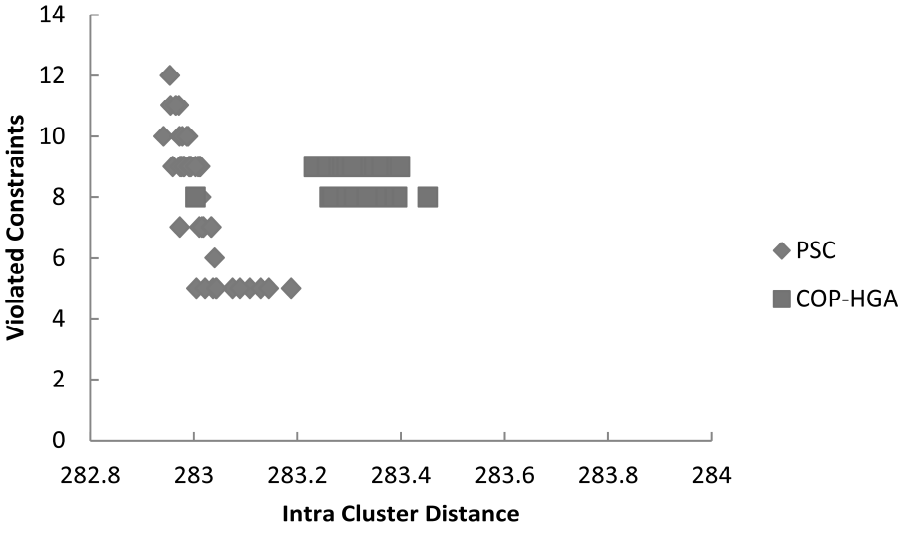


Fig. 6. Distribution of population for 400 random constraints on News-related data set

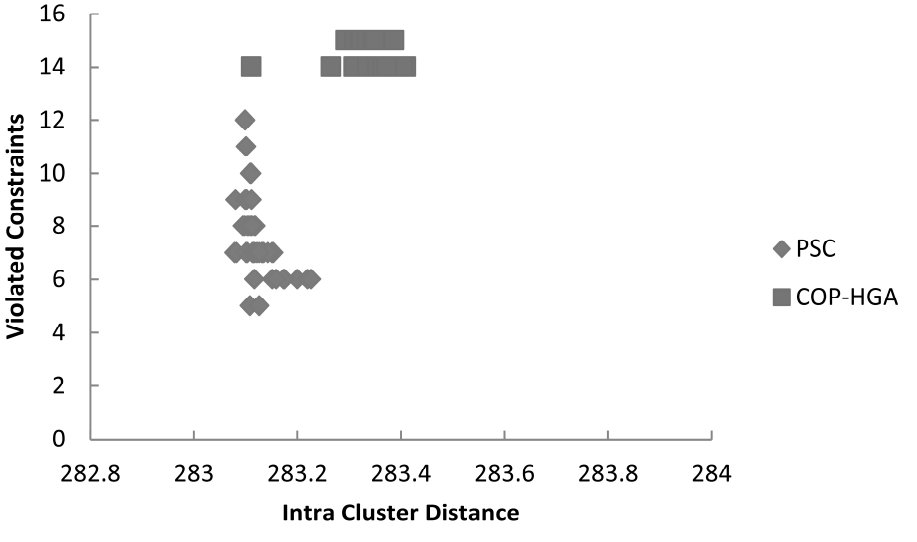


Fig. 7. Distribution of population for 500 random constraints on News-related data set

We have intentionally selected sets of constraints that demonstrate the difference better. It is clear that in PSC the population is more diverse and also exploits a better region. In other words by minimizing the two objectives independently, they are both optimized more effectively and chances are that we find better solutions. Figure 8-10 depict it from another perspective, for text data sets, we show the *Rand Index* of the best individual in population at every generation. To save space we only show the

case where a random 400 constraints are given. COP-HGA narrows down the search space quickly which results in premature convergence while in PSC the *Rand Index* is gradually increased. Note that like the previous observation; here we have also intentionally selected the set of constraints that show the difference more clearly. Another reason for PSC strength lies in crossover and mutation operators whereas COP-HGA uses an evolutionary strategy that lacks the exploration power of crossover operators.

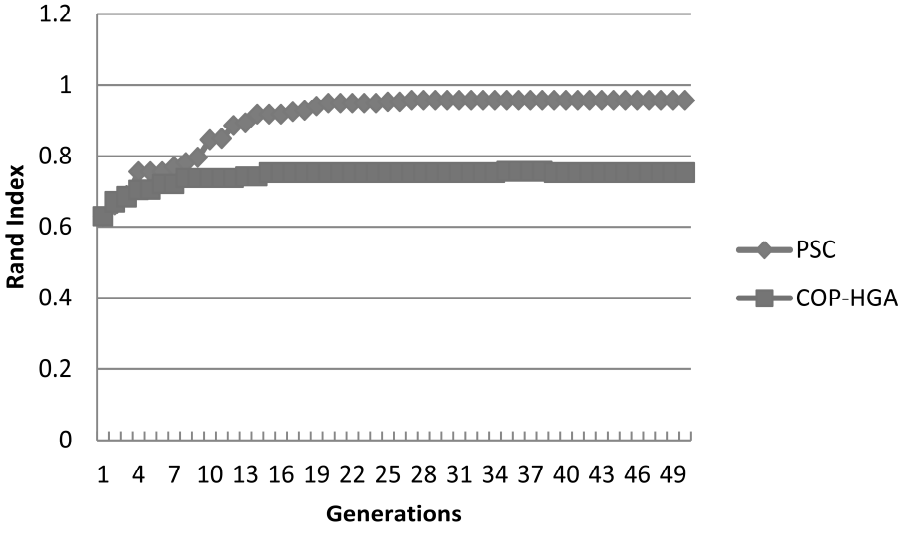


Fig. 8. Best Rand Index in each generation for 400 random constraints on News-different data set

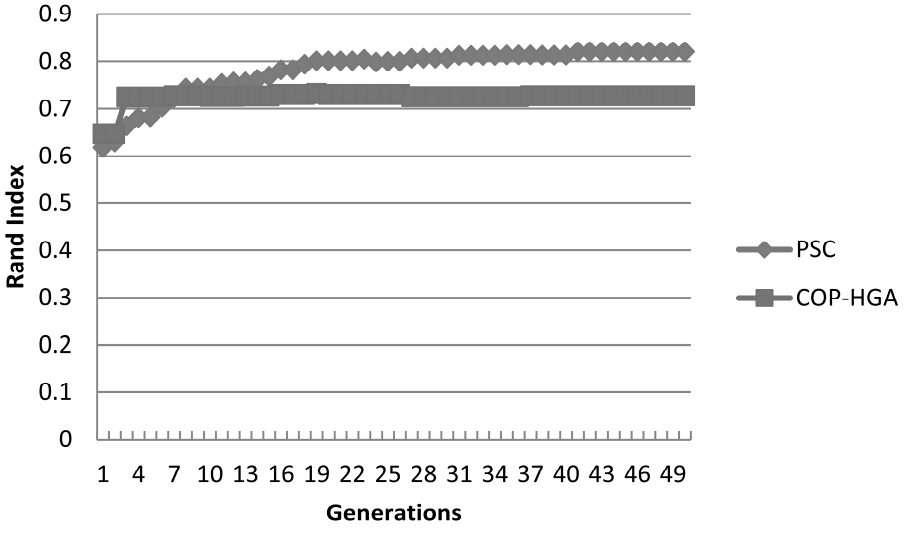


Fig. 9. Best Rand Index in each generation for 400 random constraints on News-related data set

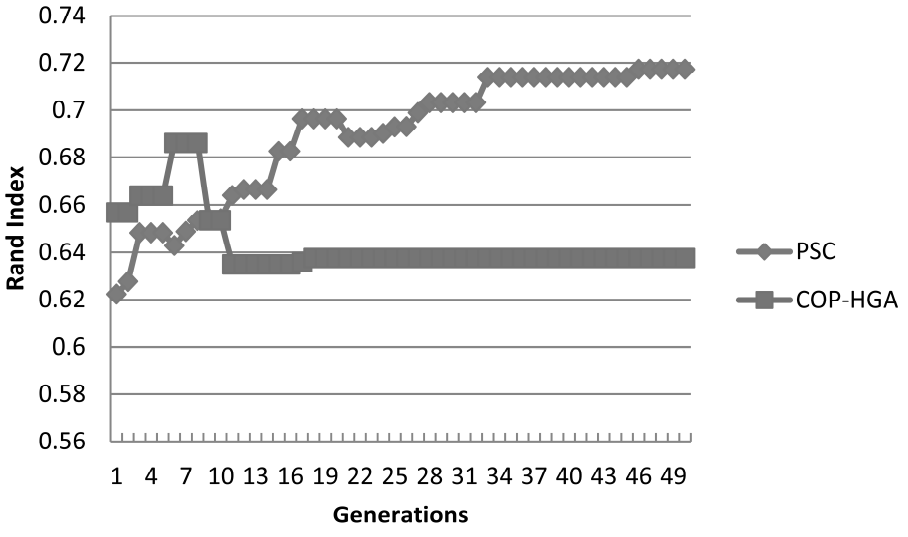


Fig. 10. Best Rand Index in each generation for 400 random constraints on News-similar data set

6 Conclusion

Recently search based Meta heuristic methods have been employed to improve heuristic based methods e.g. K-means and probabilistic methods e.g. EM, since they only guarantee local optimum. In this paper, semi supervised clustering was investigated by a Pareto optimization method. A new assignment step that guided the search, along with NSGAII was used to explore the search space more effectively. Our algorithm PSC proved better than greedy PCK-means and GA-based COP-HGA. Our observations shows that search based meta-heuristic algorithms are a promising candidate for solving semi supervised clustering problems. Also more analysis needs to be done to investigate the properties of objectives and their latent relation.

References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. ACM Comput. Surv., 264–323 (1999)
2. Basu, S., Banerjee, A., Mooney, R.J.: Active Semi-Supervision for Pairwise Constrained Clustering. In: SDM (2004)
3. Basu, S., Bilenko, M., Mooney, R.J.: A probabilistic framework for semi-supervised clustering. In: KDD, pp. 59–68 (2004)
4. Basu, S., Davidson, I., Wagstaff, K.L.: Constrained Clustering: Advances in Algorithms, Theory, and Applications, 1st edn. Chapman and Hall/CRC (2008)
5. Aliguliyev, R.M.: Clustering of document collection - A weighting approach. Expert Syst. Appl, 7904–7916 (2009)

6. Maulik, U., Bandyopadhyay, S.: Genetic algorithm-based clustering technique. Pattern Recognition, 1455–1465 (2000)
7. Das, S., Abraham, A., Konar, A.: Metaheuristic Clustering. SCI, vol. 178. Springer, Heidelberg (2009)
8. Cui, X., Palathingal, P., Potok, P.: Document Clustering using Particle Swarm Optimization. In: IEEE Swarm Intelligence Symposium 2005, Pasadena, California, pp. 185–191 (2005)
9. Handl, J., Meyer, B.: Ant-based and swarm-based clustering. Swarm Intelligence, 95–113 (2007)
10. Das, S., Konar, A.: Automatic image pixel clustering with an improved differential evolution. Appl. Soft Comput., 226–236 (2009)
11. Song, W., Choi, L.C., Park, S.C., Ding, X.F.: Fuzzy evolutionary optimization modeling and its applications to unsupervised categorization and extractive summarization. Expert Syst. Appl., 9112–9121 (2011)
12. Hong, Y., Kwong, S., Xiong, H., Ren, Q.: Genetic-guided semi-supervised clustering algorithm with instance-level constraints. In: GECCO, pp. 1381–1388 (2008)
13. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval, 1st edn. Cambridge University Press (2008)
14. Davies, D., Bouldin, D.: A cluster separationmeasure. IEEE Trans. Pattern Anal. Mach. Intell. 1, 224–227 (1979)
15. Coelho, G.P., França, F.O.D., Zuben, F.J.V.: Multi-Objective Biclustering: When Non-dominated Solutions are not Enough. J. Math. Model. Algorithms, 175–202 (2009)
16. Maitra, M., Chatterjee, A.: A hybrid cooperative-comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding. Expert Syst. Appl., 1341–1350 (2008)
17. Mitra, S., Banka, H.: Multi-objective evolutionary biclustering of gene expression data. Pattern Recognition, 2464–2477 (2006)
18. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained K-means Clustering with Background Knowledge. In: ICML, pp. 577–584 (2001)
19. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: ICML (2004)
20. Hong, Y., Kwong, S., Wang, H., Ren, Q., Chang, Y.: Probabilistic and Graphical Model based Genetic Algorithm Driven Clustering with Instance-level Constraints. In: IEEE Congress on Evolutionary Computation, pp. 322–329 (2008)
21. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimisation: NSGA-II. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000)
22. Coello, C.A.C.: A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. Knowl. Inf. Syst., 129–156 (1999)
23. Sindhya, K., Deb, K., Miettinen, K.: A Local Search Based Evolutionary Multi-objective Optimization Approach for Fast and Accurate Convergence. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 815–824. Springer, Heidelberg (2008)
24. Mahfoud, S.: Niching Methods for Genetic Algorithms. PhD thesis, University of Illinois at Urbana Champaign (1995)
25. Hruschka, E.R., Campello, R.J.G.B., Freitas, A.A., Carvalho, A.C.P.L.F.D.: A Survey of Evolutionary Algorithms for Clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 133–155 (2009)

26. Pizzuti, C.: GA-Net: A Genetic Algorithm for Community Detection in Social Networks. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 1081–1090. Springer, Heidelberg (2008)
27. Firat, A., Chatterjee, S., Yilmaz, M.: Genetic clustering of social networks using random walks. Computational Statistics & Data Analysis, 6285–6294 (2007)
28. Mitchell, T.M.: Machine learning. McGraw Hill series in computer science, pp. 1–414 (1997)
29. Shi, J., Malik, J.: Normalized Cuts and Image Segmentation. IEEE Trans. Pattern Anal. Mach. Intell., 888–905 (2000)
30. Ng, A.Y., Jordan, M.I., Weiss, Y.: On Spectral Clustering: Analysis and an algorithm. In: NIPS, pp. 849–856 (2001)
31. UCI Machine Learning Repository, http://www.ics.uci.edu/~mlearn/MLRepository.html
32. Repository of information on semi-supervised clustering, University of Texas at Austin, http://www.cs.utexas.edu/users/ml/risc/

Semi-supervised Clustering: A Case Study

Andreia Silva and Cláudia Antunes

Department of Computer Science and Engineering
Instituto Superior Técnico – Technical University of Lisbon
Lisbon, Portugal
{andreia.silva,claudia.antunes}@ist.utl.pt

Abstract. The exploration of domain knowledge to improve the mining process begins to give its first results. For example, the use of domain-driven constraints allows the focusing of the discovery process on more useful patterns, from the user's point of view. Semi-supervised clustering is a technique that partitions unlabeled data by making use of domain knowledge, usually expressed as pairwise constraints among instances or just as an additional set of labeled instances. This work aims for studying the efficacy of semi-supervised clustering, on the problem of determining if some movie will achieve or not an award, just based on the movies characteristics and on ratings given by spectators. Experimental results show that, in general, semi-supervised clustering achieves better accuracy than unsupervised methods.

1 Introduction

Both clustering and classification aim for creating a model able to distinguish instances from different categories. The main difference between them is that, in the first case, the categories are not known in advance, neither their number in most cases. In unsupervised clustering, an unlabeled dataset is partitioned into groups of similar examples, typically by optimizing an objective function that characterizes good partitions. On the opposite side, in supervised classification there is a known, fixed set of categories, and labeled training data is used to induce a classification model.

Semi-supervised learning combines labeled and unlabeled data to improve performance, and is applicable to both clustering and classification. In semi-supervised clustering, some labeled data or other types of constraints are used along with the unlabeled data to obtain better clustering. Semi-supervised classification uses some unlabeled data to train the classifier. The main difference between these two approaches is that, unlike semi-supervised classification, in semi-supervised clustering the data categories can be extended and modified as needed to reflect other regularities in the data. This is very important in domains where knowledge of the relevant categories is incomplete or where labeled data does not contain examples of all categories.

In this work, we will focus on semi-supervised clustering, and apply it to a movies domain. We first collected and joined two different movie datasets

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 252–263, 2012.

(described in section 3), in order to gather information not only about movies' characteristics, like genre, director, awards, etc., but also about the ratings given by costumers.

Our goal is to improve clustering prediction accuracy in the absence of enough labeled data, and predict which movies have or do not have at least one award. In this sense, we run some existing semi-supervised clustering algorithms[1] with several fractions of labeled data, and compare their performance with unsupervised clustering, to study the effects of seeding, and with supervised classification, to have an upper bound corresponding to the complete knowledge of the categories and all labeled data for training.

Our goal is also to analyze the efficacy of the algorithms, not only in terms of their final accuracy, but also in terms of other measures such as precision, recall and specificity.

The algorithms used are described next, and a case study on the movies domain is presented in section 3. Section 4 concludes the work, by discussing the advantages and disadvantages of semi-supervised clustering.

2 Semi-supervised Clustering

Semi-supervised clustering and its applications have been the focus of several recent projects. The most common and simplest semi-supervised clustering algorithms studied nowadays are modifications of the K-Means algorithm[7] to incorporate domain knowledge, typically through a set of labeled data points (called seeds) or pairwise constraints between the instances. This knowledge is being used to change the objective function so that it includes the satisfaction of constraints, to force some assignments to clusters[14,1] or to initialize cluster centroids[1] There are also some metric-based algorithms that employ a distance metric, trained in the beginning of the algorithm or in each iteration to satisfy the existing labels or constraints.[6,4]

There has been an effort to incorporate constraints into a more complex algorithm, Expectation Maximization (EM),[5,12] and also an effort to use other types of constraints.[3]

The used algorithms are described below.

2.1 Unsupervised Clustering with K-Means

K-Means[7] is the simplest and best known unsupervised clustering algorithm, commonly used as a baseline to compare with other clustering algorithms. It partitions a dataset into K clusters, locally minimizing the squared Euclidean distance between the data points and cluster centroids. It starts with random initial centroids and iteratively refines the clustering by assigning each instance to the nearest centroid and then recomputing each centroid as the mean of the instances of each cluster, until convergence.

[1] Available in the WEKAUT machine learning toolkit, an open source tool: http://www.cs.utexas.edu/users/ml/risc/code

Let $\chi = \{x_i\}_{i=1}^{N}$, with $x_i \in \mathbb{R}^d$ be a set of N data points, where the i^{th} data point is a vector represented by x_i with d dimensions. $\{\mu_h\}_{h=1}^{K}$ represents the K cluster centroids and $l_i \in \{1, ..., K\}$ is the cluster label of the point x_i. K-Means iteratively creates a K-partitioning $\{\chi_h\}_{h=1}^{K}$ of χ so that the objective function $J_{kmeans} = \sum_{x_i \in \chi} \|x_i - \mu_{l_i}\|^2$ is locally minimized.

2.2 Unsupervised Clustering with Expectation Maximization

The EM algorithm[5] is not specific for clustering, being first proposed for parameter estimation with missing data. EM is an iterative procedure that estimates the Maximum Likelihood: $\hat{\Theta}_{ML} = \arg\max_{\Theta} L(\chi|\Theta) = \arg\max_{\Theta} P(\chi, Y|\Theta)$ of the missing data Y (cluster assignments, in this case) for which the observed data χ is the most likely.

Each iteration of EM consists of two steps, repeated until convergence – Expectation (E-step) and Maximization (M-step):

$$\text{E-step:} \qquad Q(\Theta, \hat{\Theta}^{(t)}) = \mathbb{E}_{Y|\chi}[logP(\chi, Y|\Theta)|\chi, \hat{\Theta}^{(t)}]$$

$$\text{M-step:} \qquad \hat{\Theta}^{(t+1)} = \arg\max_{\Theta} Q(\Theta, \hat{\Theta}^{(t)})$$

In the E-step, given the observed data and current estimate of the model parameters, the missing data is estimated using its conditional expectation. In the M-step, under the assumption that the missing data is known, the likelihood is maximized.

It was shown in [1] that the K-Means algorithm is essentially an EM algorithm on a mixture of K Gaussians under assumptions of identity covariance of the Gaussians, uniform mixture component priors and expectation under a particular type of conditional distribution (which can be provided by semi-supervision). Under the assumption of identity covariance, the parameters to estimate are just the means of the Gaussians, i.e. the centroids of the K clusters, and the squared Euclidean distance between a point x_i and its corresponding cluster centroid μ_l is $\|x_i - \mu_{l_i}\|^2 = (x_i - \mu_{l_i})^T (x_i - \mu_{l_i})$.

2.3 Semi-supervised Clustering by Seeding

Let $S \subseteq \chi$ be a subset of data points, called the *seed set*, where for each $x_i \in S$ we have the label l of the partition χ_l to which it belongs. It is assumed that there is, at least, one seed point for each partition. The algorithms receive, therefore, a disjoint K-partition $\{S_l\}_{l=1}^{K}$ (*the seed clustering*) of the seed set S, so that all $x_i \in S_l$ belongs to χ_l, according to the supervision.

Seeded KMeans[1]. In this algorithm, the seed clustering is only used to initialize the KMeans algorithm, and it is not used in the following steps. Instead of initializing KMeans from K random means, the mean of the l^{th} cluster is initialized with the mean of the l^{th} partition of S_l of the seed set. In Seeded KMeans, the labels specified in the seed data may change in the course of the algorithm. Therefore, it is appropriate in the presence of noisy seeds.

Constrained KMeans[1]. Like Seeded KMeans, it uses the seed clustering to initialize the centroids in KMeans. However, in subsequent steps, cluster labels of seed data are kept unchanged in the cluster assignment steps, and only the labels of the non-seed data are re-estimated. It is appropriate when the initial seed labeling is noise-free, or if the user does not want the labels of the seed data to change.

2.4 Semi-supervised Clustering with Pairwise Constraints

Let us define two types of pairwise constraints, introduced by [14], that provide *a priori* knowledge about which instances should be grouped or not. Let M be a set of must-link pairs, where $(x_i, x_j) \in M$ implies x_i and x_j should be in the same cluster, and C be a set of cannot-link pairs where $(x_i, x_j) \in C$ implies x_i and x_j should be in different clusters. Let also $W = \{w_{ij}\}$ and $\overline{W} = \{\overline{w}_{ij}\}$ be penalty costs for violating the constraints in M and C respectively.

PCKMeans[4]. Pairwise Constrained KMeans uses soft pairwise constraints, i.e. it allows violation of constraints if it leads to a more cohesive clustering, and uses them to seed the initial cluster centroids and to influence the clustering.

The goal of PCKMeans is to minimize a combined objective function, defined as the sum of the total squared distances between the points and their cluster centroids, and the cost incurred by violating any pairwise constraint:

$$J_{pckmeans} = \sum_{x_i \in \chi} \|x_i - \mu_{l_i}\|^2 + \sum_{(x_i,x_j)\in M} w_{i,j} \mathbb{1}_{l_i \neq l_j} + \sum_{(x_i,x_j)\in C} \overline{w}_{i,j} \mathbb{1}_{l_i = l_j}$$

where point x_i is assigned to the partition χ_l with centroid μ_{l_i}, and where $\mathbb{1}$ is the indicator function: $\mathbb{1}_{true} = 1$ and $\mathbb{1}_{false} = 0$.

MPCKMeans[4]. Metric PCKMeans is like PCKMeans, with a metric learner component. It performs distance-metric training in each clustering iteration, making use of both unlabeled data and pairwise constraints. The Euclidean distance between two points is defined using a symmetric positive-definite weigh matrix A: $\|x_i - \mu_{l_i}\|_A = \sqrt{(x_i - \mu_{l_i})^T A (x_i - \mu_{l_i})}$. And by using a separate weight matrix A_l for each cluster l, the algorithm is capable of learning individual metrics for each cluster, which allows clusters of different shapes.

The goal of MPCKMeans is also to minimize a combined objective function like PCKMeans, but now including the weight matrices A_h and allowing different costs of constraint violations, via a function f, so that *distant* and *nearby* points can be treated differently:

$$J_{mpckmeans} = \sum_{x_i \in \chi} (\|x_i - \mu_{l_i}\|^2_{A_{l_i}} - log(det(A_{l_l}))) + \sum_{(x_i,x_j)\in M} w_{i,j} f_M(x_i, x_j) \mathbb{1}_{l_i \neq l_j} + \sum_{(x_i,x_j)\in C} \overline{w}_{i,j} f_C(x_i, x_j) \mathbb{1}_{l_i = l_j}$$

where $f_M(x_i, x_j) = \frac{1}{2}\|x_i - x_j\|^2_{A_{l_i}} + \frac{1}{2}\|x_i - x_j\|^2_{A_{l_j}}$ is the cost violation of a must-link (i.e. if $i \neq j$), and $f_C(x_i, x_j) = \|x'_{l_i} - x''_{l_i}\|^2_{A_{l_i}} - \|x_i - x_j\|^2_{A_{l_i}}$, with (x'_{l_i}, x''_{l_i}) the maximally separated pair of points in the dataset according to l_i^{th} metric, is the cost violation of a cannot-link (i.e. if $i = j$).

MPCKMeans uses EM: the E-step consists of assigning each point to the cluster that minimizes $J_{mpckmeans}$, based on the previous assignments of points to clusters; the M-step consists of two parts: re-estimating the cluster centroids given the E-step cluster assignments, like KMeans; and re-learning the metric matrices $\{A_h\}_{h=1}^K$ to decrease $J_{mpckmeans}$. Each updated matrix is obtained by taking the partial derivative $\frac{\partial Jmpckmeans}{\partial A_h}$ and setting it to zero, resulting in:

$$A_h = |\chi_h|\Big(\sum_{x_i \in \chi_h} (x_i - \mu_h)(x_i - \mu_h)^T + \sum_{(x_i,x_j)\in M_h} \frac{1}{2} w_{ij}(x_i - x_j)(x_i - x_j)^T \mathbb{1}_{l_i \neq l_j} + \sum_{(x_i,x_j)\in C_h} \overline{w}_{ij}((x'_h - x''_h)(x'_h - x''_h)^T - (x_i - x_j)(x_i - x_j)^T \mathbb{1}_{l_i = l_j})^{-1}$$

3 Case Study

In this case study, we decided to join two movie datasets, in order to predict whether a movie has (or will have) an award, based not only on the ratings given by customers, but also based on other characteristics of the movie, such as its genre, studio, director, etc.

This study compares the efficacy obtained by applying semi-supervised clustering techniques versus unsupervised clustering. We also run a supervised classification algorithm just to have an upper bound, corresponding to the complete knowledge of the categories and all labeled data for training.

All the experiments were conducted using implementations incorporated into the WEKAUT machine learning toolkit.[2] The applied algorithms were: KMeans [7] and EM [5] for unsupervised clustering; Seeded-KMeans and Constrained-KMeans [1], PCKMeans and MPCKMeans [4] for semi-supervised clustering; and C4.5 [9] for classification.

For all algorithms, we have generated learning curves with 10-fold cross-validation and, in each fold, 10% of the dataset is set aside as the test set, and the remaining is used as the training set. Since the results on semi-supervised clustering can diverge with the size of the given seeds, we experimented with different fractions of seeds, generated randomly from the training set. For constrained-based algorithms, we used the selected fraction of seeds to build the respective must-links and cannot-links randomly (50% of each pairwise constraints). Unit constraint costs were used for both constraints, since the dataset did not provide individual weights.

[2] WEKAUT is a Data Mining open source tool available online at `http://www.cs.utexas.edu/users/ml/risc/code`

For each algorithm, we evaluate the accuracy of the results using the *Rand Index* metric,[10] which compares the resulted clusters with the true labels available, calculating the fraction of correctly predicted movies. We also computed the *F-measure* of the results, which makes a balance between their precision (the fraction of truly awarded movies among those that the algorithm believes to have an award) and recall (the fraction of positive awarded movies correctly predicted).[8]

Clustering algorithms were run on the whole dataset, but the measures were calculated only on the test set. After running, each category ("have" or "do not have" an award) is assigned to the cluster with more instances of that category, so that we can evaluate the efficacy of the results. Results were averaged over the 10 folds.

We first describe the datasets and the preprocessing needed. Then, the experimental results are presented and discussed.

3.1 Data Description

Netflix Dataset. The Netflix dataset used in these experiments was constructed specially for the Netflix Prize.[3]

It contains over 100 million ratings from 480 thousand randomly-chosen, anonymous Netflix customers, over 17 thousand movie titles. The data were collected between October, 1998 and December, 2005 and reflect the distribution of all ratings received during this period. The ratings are on a scale from 1 to 5 (integral) stars. The year of release (from 1890 to 2005) and title of each movie are also provided.

It is known that, in most cases, movies with awards are those that many people like, and therefore those with higher ratings. Since this dataset has no other information about the movies, except their year of release (not interesting when we want to create a model to predict the future) and their title (also not interesting because in most cases, each movie has a different one), we decided to use another dataset, described below.

Gio Dataset. This dataset was extracted from a movies database donated by Gio Widerhold,[15] which collects data about more than 10 thousand movies, released between 1891 and 1999. It contains characteristics such as the genre, directors, actors, studios and awards received for each movie.

All this extra information may help predicting if a movie has an award. For example, we know some movie directors, like James Cameron and others, who have already received several awards for most of their work. We also know some actors that tend to receive awards.

Final Dataset. To have both the ratings and other characteristics of the movies in the same dataset, we joined the movies of those two datasets with the equivalent title and the same year of release. There are 2800 movies in common. Of these, 34.5% have an award and the remaining 65.5% do not have. Only these

[3] See http://www.netflixprize.com for details.

common movies were used in the mining process, in order to evaluate the accuracy of the clustering algorithms.

Table 1 shows the attributes chosen for the final table, as well as their meaning. The attribute year was discarded, since we want to build a model that predicts if new movies will have awards (indeed, using this attribute, C4.5 produces a tree which decides first by the year of the movie, and therefore, based on the past, which is what we want to avoid). And since the total number of ratings per movie has a wide range (from one to more than 200 thousand, with a median of just 561 ratings per movie), we used the percentage of customers that voted in a movie and gave each rating.

Table 1. Attributes in the final dataset

Attribute	**Type**	**Missing**	**Description**
Rate 1 Rate 2 Rate 3 Rate 4 Rate 5	Real: [0 ; 1]	0%	Percentage of customers that voted in the movie and gave the star 1, 2, 3, 4 or 5, respectively
Med Rate	Integer: 1 to 5	0%	The medium rate of the movie
Director	Nominal: 1190 values	0%	Director's name
Actor 1	Nominal: 1808 values	10%	First main actor's name
Actor 2	Nominal: 1772 values	13%	Second main actor's name
Studio Name	Nominal: 314 values	35%	Studio's name
Studio Country	Nominal: 10 values	57%	Country where the movie was made
Genre	Nominal: 12 values	0%	Genre of the movie
Awards	Boolean	0%	True if the movie has an award; False otherwise

To deal with the missing values of an attribute, the implemented algorithms adopt different strategies. K-Means replaces them by the mode or mean of the attribute, while EM ignores them, and the semi-supervised algorithms available are not able to deal with them. In the context of actors and studios, which has many possible values, with completely different meanings, it makes no sense to replace missing values of an attribute by its mode. Therefore, they were replaced by a non-existing value, "zero", representing the lack of value.

3.2 Experimental Results

We first run the classification algorithm to observe its performance, and which attributes were used to build the model. Indeed, the final tree has attributes from both datasets: in a first level of decision nodes, it uses the attribute "studio name"; In the second and third levels, it uses the ratings or the genres of the movies. As an example, for a movie produced in the studio *Gaumont*:

if $StudioName = Gaumont$
  if $Med <= 3 : false$
  or if $Med > 3$
    if $Rate3 <= 0.25 : false$
    or if $Rate3 > 0.25 : true$

if the average ratings ("*Med*") is less than or equal to 3, it means the movie has no award. Otherwise, even with an average higher than 3, it only has an award if more than 25% of customers rated the movie with 3.

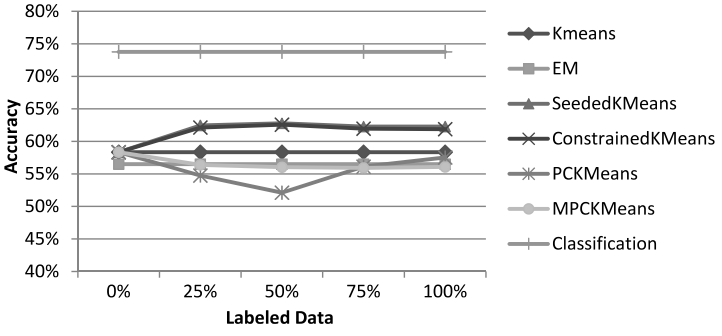


Fig. 1. Accuracy results

Figure 1 shows the accuracy (or Rand Index) of the algorithms, for different fractions of labeled instances incorporated. As we can see, semi-supervised clustering by seeding (Seeded and Constrained KMeans) always performed better than unsupervised clustering (KMeans and EM), even with a small number of seeds (% of labeled data). On the other hand, semi-supervised clustering by constraints (PC and MPC KMeans) seems have worst accuracy results. PCKMeans only started getting improvements with more than 50% of constraints, but it does not achieve better accuracy results than unsupervised clustering algorithms. Note that the labeled data is selected only from the train set, and the results are computed from the test set.

Although seeding algorithms achieve a better accuracy, it does not show where the correct decisions come from (positive awarded – also called true positives, or negative awarded movies – called true negatives). At figure 2 we analyze just that, by showing the recall and the specificity of the algorithms. The recall is the fraction of positive awarded movies correctly predicted (also known as True Positive Rate or sensitivity) and specificity is the fraction of non awarded movies correctly predicted as such (True Negative Rate).

In the chart we can see that semi-supervised algorithms were much better to correctly predict which movies do not have an award, than predicting which movies have (very high specificity versus very low recall). Only MPCKMeans had the recall and specificity at the same level as the unsupervised EM and KMeans. Seeded and Constrained KMeans are very similar to each other, and have a very low recall. However, their specificity is much higher than all other algorithms, even classification.

Finally, figure 3 presents the precision and the f-measure of the models built. Precision is the fraction of truly awarded movies among those that the algorithm

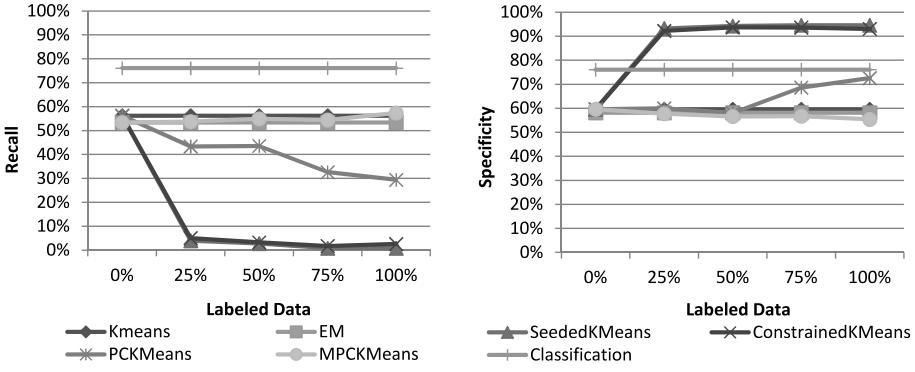


Fig. 2. Recall (TP Rate) and Specificity (TN Rate) results

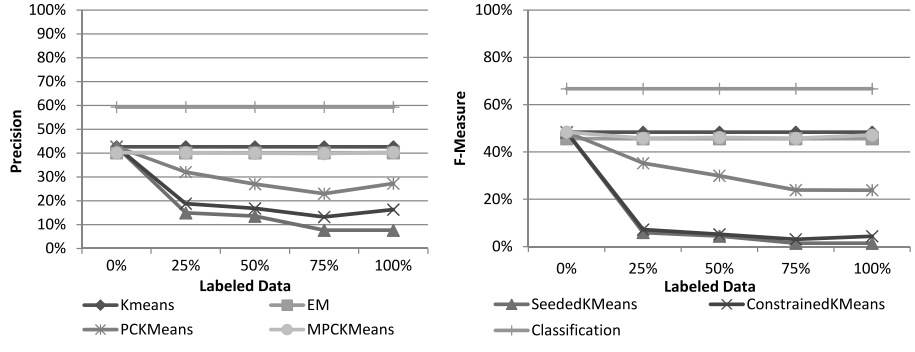


Fig. 3. Precision and F-Measure results

believes to have an award, and f-measure is a balance between precision and recall.

By analyzing the precision, we confirm that semi-supervised methods have difficulties in creating the cluster for positive awarded movies, since a lower precision indicates the presence of a lot of non-awarded movies in the cluster that represents the awarded ones. Although similar, Seeded KMeans proved to be worse than Constrained KMeans. Only MPCKMeans behaves like unsupervised clustering, maintaining its behavior. The F-measure culminates these results, showing that the best balance between the precision and recall is MPCKMeans, and that seeded algorithms are not good in none of them.

3.3 Discussion

As results showed, semi-supervised clustering does not always bring improvements over unsupervised clustering accuracy. However, the seeding algorithms can achieve a better overall accuracy.

The bad results in recall and very good ones in specificity can be explained by the fact that the dataset is unbalanced, with 34.5% of awarded movies, versus 65.5% of non awarded movies. This problem, also known as skewed class distribution, may cause the learning algorithms have difficulty in learning the concepts related to the minority class. In this case, seeded clustering has difficulties in clustering awarded movies (low recall), but can cluster non awarded movies very well (high specificity). This problem can be addressed by forms of sampling and other techniques that transform the dataset into a more balanced one.[11] However, any real dataset for this kind of analysis, used to predict if a movie has (or will have) an award, will always have much more movies without an award than with one. Therefore, the results obtained for an extracted balanced dataset may not be significant for a real dataset.

Another problem that can also explain the low results in recall is related to the choice of seeds and constraints, and is the fact that we do not really know how they will influence the results. Researchers have already studied this problem, and explain it with two properties of seed or constraint sets: *informativeness* and *coherence*.[13] Informativeness refers to "the amount of information in the set that the algorithm cannot determine on its own". Coherence is "the amount of agreement between the elements in the set, given a distance metric". They recommend the selection of sets with higher informativeness and coherence values, to avoid situations where the constraints negatively impact the performance. There are already some good works that already exploit this combination.[2]

Other interesting fact we can observe is the constant behavior of Seeded and Constrained KMeans. We see that more seeds do not change their results, meaning that new seeds do not bring new relevant information for cluster initialization.They had also very similar results. This may indicate that the initial seeds are not noisy, and that the clusters have a wide overlapping (and in fact the accuracy is above 50%, which indicates that the dataset is not easily separable).

In global terms, we can conclude that semi-supervised clustering based on seeds achieves better accuracy than unsupervised clustering, even with few fractions of seeds. They may have problems predicting which movies have an award, but they can predict, better than the others, which movies do not have an award, and therefore they can help, for example, limiting the number of candidates to awards, or eliminating a larger set of non awarded movies before another algorithm run.

4 Conclusions

Unlike unsupervised clustering, semi-supervised clustering uses some labeled data to aid the search and bias the partitioning of unlabeled data, and therefore, they can generally learn better models and achieve better accuracy results. However, the good accuracy may stem from just one or some correct decisions and not from all.

With our case study, we can conclude that semi-supervised clustering by seeding proved to be better in accuracy than approaches that use pairwise constraints and unsupervised ones, with good results even for a few fraction of seeds.

However, there are some questions we have to consider, for example:

Is the dataset unbalanced? Many real datasets are. If so, it is expected the algorithm to learn better how to cluster the instances of the most common class, and have problems with the minority class, as shown here;

What is the goal of the learning task? We cannot just look to the accuracy of an algorithm, we should analyze other measures, like recall, specificity, precision and/or f-measure, accordingly to our goal;

To what extent the seeds help? As explained before, seeds not always help. We should try to select seed sets or constraints with higher informativeness and coherence values. Future work could go through a study testing several sets of seeds and constraints.

Acknowledgment. This work is partially supported by FCT – Fundação para a Ciência e a Tecnologia, under research project D2PM (PTDC/EIA-EIA/110074/2009) and PhD grant SFRH/BD/64108/2009.

References

1. Basu, S., Banerjee, A., Mooney, R.: Semi–supervised clustering by seeding. In: Proceedings of 19th International Conference on Machine Learning, ICML (2002)
2. Basu, S., Banerjee, A., Mooney, R.J.: Active semi-supervision for pairwise constrained clustering. In: Proceedings of the 2004 SIAM International Conference on Data Mining (SDM), pp. 333–344 (2004)
3. Basu, S., Davidson, I., Wagstaff, K. (eds.): Constrained Clustering: Advances in Algorithms, Theory, and Applications. Chapman and Hall/CRC (2008)
4. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: Proceedings of 21st International Conference on Machine Learning, ICML (2004)
5. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1), 1–38 (1977)
6. Klein, D., Kamvar, S.D., Manning, C.D.: From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In: Proceedings of 19th International Conference on Machine Learning (ICML), pp. 307–314 (2002)
7. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297 (1967)
8. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval. Cambridge University Press (2008)
9. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
10. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66(336), 846–850 (1971)
11. Scholz, M.: Sampling-based sequential subgroup mining. In: Knowledge Discovery and Data Mining, pp. 265–274 (2005)

12. Shental, N., Bar-Hillel, A., Hertz, T., Weinshall, D.: Computing gaussian mixture models with em using equivalence constraints. In: Advances in Neural Information Processing Systems (NIPS) 16 (2003)
13. Wagstaff, K., Basu, S., Davidson, I.: When is constrained clustering beneficial, and why. In: AAAI (2006)
14. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: Proceedings of 18th International Conference on Machine Learning (ICML), pp. 577–584 (2001)
15. Wiederhold, G.: Movies database documentation (1989)

SOStream: Self Organizing Density-Based Clustering over Data Stream*

Charlie Isaksson, Margaret H. Dunham, and Michael Hahsler

Department of Computer Science and Engineering,
Southern Methodist University, Dallas, Texas, USA
charlie.isaksson@tektronix.com, {mhd,mhahsler}@lyle.smu.edu

Abstract. In this paper we propose a data stream clustering algorithm, called Self Organizing density based clustering over data Stream (SOStream). This algorithm has several novel features. Instead of using a fixed, user defined similarity threshold or a static grid, SOStream detects structure within fast evolving data streams by automatically adapting the threshold for density-based clustering. It also employs a novel cluster updating strategy which is inspired by competitive learning techniques developed for Self Organizing Maps (SOMs). In addition, SOStream has built-in online functionality to support advanced stream clustering operations including merging and fading. This makes SOStream completely online with no separate offline components. Experiments performed on KDD Cup'99 and artificial datasets indicate that SOStream is an effective and superior algorithm in creating clusters of higher purity while having lower space and time requirements compared to previous stream clustering algorithms.

Keywords: Adaptive Threshold, Data Stream Clustering, Density-Based Clustering, Self Organizing Maps.

1 Introduction

Data stream mining has recently captured an enormous amount of attention. Stream mining can be defined as the process of finding complex structure within a large volume of data where the data evolves over time and arrives in an unbounded stream. A data stream is a sequence of continuously arriving data which imposes a single pass restriction where random access to the data is not feasible. Moreover, it is impractical to store all the arriving data. In this case, cluster features or synopses that typically include descriptive statistics for a cluster are used. In many cases, data stream algorithms have to observe space and time constraints. Stream clustering algorithms are used to group events based on similarity between features. Data arriving in streams often contain noise and outliers. Thus, data stream clustering should be able to detect, distinguish and filter this data prior to clustering.

Inspired by both DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [1] and SOM (Self Organizing Maps) [2], we propose a new data stream clustering algorithm, Self Organizing density-based clustering over data Stream (SOStream).

* This work was supported in part by the U.S. National Science Foundation NSF-IIS-0948893.

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 264–278, 2012.

SOStream is a density-based clustering algorithm that can adapt its threshold to the data stream. It uses an exponential fading function to reduce the impact of old data whose relevance diminishes over time. SOStream has the following novel features:

- Setting a threshold manually for density-based clustering (similarity threshold, grid size, etc.) is difficult and if this parameter is set to an unsuitable value, then the algorithm will suffer from overfitting, while at the other extreme the clustering is unstable. SOStream addresses this problem by using a dynamically learned threshold value for each cluster based on the idea of building neighborhoods with a minimum number of points.
- SOStream employs a novel cluster updating strategy which is inspired by competitive learning techniques developed for Self Organizing Maps (SOMs) [2] and CURE (Clustering Using REpresentatives) [3]. CURE utilizes a unique shrinking strategy that encouraged us to implement the same methodology for SOStream. The micro-clusters that are formed after shrinking are used as a representative of the global cluster. The shrinking procedure also helps to correctly identify highly-overlapped clusters (See Figure 1). As a result, the clusters become less sensitive to outliers.
- All aspects of SOStream (including deletion, addition, merging, and fading of clusters) are performed online.

We conduct experiments using both a synthetic and the KDD Cup'99 datasets [4], and demonstrate that SOStream outperforms the state-of-the-art algorithms MR-Stream [5] and D-Stream [6] without the use of an offline component. Moreover, SOStream dynamically adapts its similarity threshold. SOStream achieves better clustering quality in terms of cluster purity and utilizes less memory which is a key advantage for any data stream algorithms.

Throughout this paper, we use threshold and radius interchangeably to specifically refer to a value that is used to cluster a new point into suitable micro-cluster or to find the neighborhood of the winning micro-cluster.

The remainder of this paper is organized in the following manner: Section 2 surveys related work; Section 3 presents the SOStream framework; Section 4 presents results of experiments evaluating the performance of SOStream; and Section 5 concludes the paper.

2 Related Work

We first review the most important data stream clustering algorithms to highlight the novel features of SOStream.

E-Stream [7] starts empty and for every new point either a new cluster is created around the incoming data point or the point is mapped into one of the existing clusters based on a radius threshold. Any cluster not meeting a predefined density level is considered inactive and remains isolated until achieving a desired weight. Cluster weights decrease over time to reduce the influence of older data points. Clusters not active for a certain time period may be deleted from the data space. Also, for each step, two clusters may be merged because the overlap is sufficiently large (or the maximum cluster limit is

Table 1. Features of different data stream clustering algorithms

Algorithms	new cluster	remove	merge	fade	split
SOStream	✓	✓	✓	✓	x
E-Stream	✓	✓	✓	✓	✓
CluStream	✓	✓	offline	x	x
DenStream	✓	✓	offline	✓	x
OpticsStream	✓	✓	offline	✓	x
HPStream	✓	✓	x	✓	x
WSTREAM	✓	✓	✓	✓	x
D-Stream	✓	✓	offline	✓	x
MR-Stream	✓	✓	offline	✓	x

reached) or one cluster may be split into two sub-clusters if internal data is too diverse. The data is summarized into an α-bin histogram, and the split is made if a deep valley between two significant peaks is found.

CluStream [8] uses an online micro-clustering component to periodically store detailed summary statistics in a fast data stream while an offline macro-clustering component uses the summary statistics in conjunction with other user input to provide the user with a quick understanding of the clusters whenever required.

DenStream [9] maintains two lists: one with potential micro-clusters and the other with outlier micro-clusters. As each new point arrives, an attempt is made to merge it into one of the nearest potential micro-clusters. If the radius of the resulting micro-cluster is larger than specified, the merge is omitted, and an attempt is made to merge the point with the nearest outlier micro-cluster. If this resulting radius is larger than specified, the merge is omitted and a new outlier micro-cluster is created centered at the point. If any of the outlier micro-clusters exceeds a specified weight, they are moved into the potential micro-clusters list.

OpticsStream [10] is an online visualization algorithm producing a map representing the clustering structure where each valley represents a cluster. It adds the ordering technique from OPTICS [11] (not suitable for data stream) on top of a density-based algorithm (such as DenStream) in order to manage the cluster dynamics.

HPStream [12] is an online algorithm that discovers well-defined clusters based on a different subset of the dimensions of d-dimensional data points. For each cluster a d-dimensional vector is maintained that indicates which of the dimensions are included for continuous assignment of incoming data points to an appropriate cluster. The algorithm first assigns the received streaming data point to each of the existing clusters, computes the radii, selects the dimensions with the smallest radii, and creates a d-dimensional vector for each cluster. Next, the Manhattan distance is computed between the incoming data point and the centroid of each existing cluster. The winner is found by returning the largest average distance along the included dimensions, and the radius is computed for the winning cluster and compared to the winning distance. Then, either a new cluster is created centered at incoming data point or the incoming data point is added to the

winning cluster. Clusters are removed if the number of clusters exceeds the user defined threshold or if they contain zero dimensions.

WSTREAM [13] is density-based and discovers cluster structure by maintaining a list of rectangular windows that are incrementally adjusted over time. Each window moves based on the centroid of the cluster which is incrementally recomputed whenever new data points are inserted. It incrementally contracts or expands based on the approximated kernel density and the user defined bandwidth matrix. If two windows overlap, the proportion of intersecting data points to the remaining points in each window is computed, and, upon meeting a user defined threshold, the windows are merged. Periodically, the weights of the stored windows are checked, and a window is removed if its weight is less than the defined minimum threshold (the window is considered to be an outlier).

D-Stream [6] is density-based and works on the basis of a time step model which starts by initializing an empty hash table grid list. An online component reads the incoming raw data record, and this record is mapped to the grid list or inserted into the grid list if it does not exist. After the insertion, the characteristic vector (containing all the information about the grid) is updated. Thus, the online component partitions the data into many corresponding density grids forming grid clusters while the offline component dynamically adjusts the clusters every gap time. The gap is the key decision factor for inspecting each grid and adjusting the cluster. If the grid is empty or receives no new value for a long period of time then it is removed.

MR-Stream [5] finds clusters at versatile granularities by recursively partitioning the data space into well-defined cells by using a tree data structure. MR-Stream facilitates both online and offline components. Table 1 summarizes features of these stream clustering algorithms and shows those possessed by our new SOStream algorithm. Although not contained in the SOStream algorithm presented in this paper, we have developed a splitting function which is easily incorporated into the SOStream algorithm presented later in this paper. This will be reported in subsequent publications.

3 SOStream Framework

A key issue for clustering stream data is the online constraint, which imposes a single pass restriction over the incoming data points. Although many previously proposed stream clustering algorithms have an offline component, this is neither desirable nor necessary. In this section we introduce Self Organizing density-based clustering over data Stream (SOStream) and highlight some of its novel features.

3.1 SOStream Overview

We assume that the data stream consists of a sequence of d-dimensional input vectors where $v(t)$ is used to indicate the input vector at time t, where $t = (1, 2, 3...)$. For every time step t, SOStream is represented by a set of micro-clusters $M(t) = \{ N_1, N_2, ..., N_k \}$, where for each cluster a tuple with three elements $N_i = (n_i, r_i, C_i)$ is stored. n_i is the number of data points assigned to N_i, r_i is the cluster's radius and C_i is the centroid. The tuple is a form of synopsis or cluster feature (CF) vector. Cluster feature vectors were introduced in the non-data stream clustering algorithm BIRCH [14].

As these values change over time we include in the following description the time point where needed to identify the value at a particular time. Thus, $C_i(t)$ indicates the centroid for cluster N_i at time t. SOStream uses a centroid to describe the cluster as a d-dimensional vector. The number of clusters varies over time and depends upon the complexity of the input data. SOStream has built-in stream clustering operations to dynamically create, merge, and remove clusters in an online manner. In addition, an exponential fading function can be used to gradually reduce the impact of historical data.

SOStream uses competitive learning as introduced for SOMs where a winner influences its immediate neighborhood [2]. For each new input vector, $v(t)$, the winning cluster is determined by measuring the distance (e.g. Euclidean distance) between each existing cluster centroid and the current input vector:

$$N_{win}(t) = \underset{N_i \in M(t)}{\operatorname{argmin}} \{\, d(v(t), C_i), C_i \in N_i \,\} \tag{1}$$

If the winning cluster $N_{win}(t)$ is close enough (distance is below a dynamically determined threshold), then $v(t)$ is placed in that cluster otherwise a new cluster is created. Thus we assume a simple nearest neighbor algorithm is used. In our discussions we assume that the Euclidean distance metric is used for clustering, but any distance or similarity metric could be used. In addition, any other technique could be used to determine the winning cluster. The salient feature of SOStream is the weighted density concept described in the rest of this paper.

In the following subsections we examine the algorithm in more detail.

3.2 Density-Based Centroid

SOStream uses a centroid to identify each cluster. However the manner in which the centroid is calculated is not just a simple arithmetic mean applied to all points, $v(t)$, in the cluster. The way we calculate the centroid is inspired by the technique to update weights for the winning competitive node in a Kohonen Network [2]. In our case the winner is a cluster and the weights are associated with neighboring clusters. The centroid of a cluster is updated in several ways:

- When an input vector is added to a cluster the centroid is updated using a traditional arithmetic mean approach.
- Centroids of clusters sufficiently close to the winning clusters have their centroids modified to be closer to the winning cluster's centroid. This approach is used to aid in merging similar clusters and increasing separation between different clusters.
- Fading also adjusts the centroid values. This will be discussed later in the paper.

As the first of these techniques is straightforward, we concentrate on the second one.

As described earlier, the winning cluster is the one that is closest to the input vector. Updates are performed to the centroids of clusters that are within the neighborhood of the winning cluster. This brings clusters in the neighborhood of the winner closer to the incoming data in a similar way as the neighbors of a winning competitive node in a SOM have their weights adjusted to be closer to the winner.

We define the neighborhood of the winner based on the idea of a $MinPts$ distance given by a minimum number of neighboring objects [1,9]. This distance is found by computing the Euclidean distance from any existing clusters to the winning cluster. Then all the distances are ordered in ascending order and the maximum of the first MinPts minimum distinct distances is chosen and used to represent the radius of the winning cluster. Thus, every cluster whose distance from the winning cluster is less then the computed radius is considered to be a neighbor of the winning cluster. Note that the efficiency of this calculation can be improved using a min heap type data structure.

Motivated by Kohonen's work [2], we propose that the centroid C_i of each cluster N_i that is within the neighborhood of the winning cluster N_{win} is modified to resemble the winner:

$$C_i(t+1) = C_i(t) + \alpha\beta(C_{win}(t) - C_i(t)) \tag{2}$$

α is a scaling factor and β is a weight which represents the amount of influence of the winner on a cluster. We define β as

$$\beta = e^{-\frac{d(C_i, C_{win})}{2(r^2_{win})}} \tag{3}$$

where r_{win} denotes the radius of the winner. The definition of β ensures that $0 < \beta \leq 1$.

Next we need to prove that updating a centroid moves the cluster closer to the winning cluster, i.e.

$$d(C_i(t+1), C_{win}(t)) \leq d(C_i(t), C_{win}(t))$$

By the definition of Euclidean distance we have

$$d(C_i(t), C_{win}(t)) = \sqrt{(v_1 - z_1)^2 + (v_2 - z_2)^2 +, ..., +(v_n - z_n)^2}$$

where $C_i(t) = \langle v_1, v_2, ..., v_n\rangle$; $C_{win}(t) = \langle z_1, z_2, ..., z_n\rangle$ and $C_i(t+1) = \langle v'_1, v'_2, ..., v'_n\rangle$. If we can show that $0 < \alpha \leq 2$ is a necessary condition for $(v'_i - z_i)^2 \leq (v_i - z_i)^2$, then one can easily show that $d(C_i(t+1), C_{win}(t)) \leq d(C_i(t), C_{win}(t))$ when $0 < \alpha \leq 2$ by the definition of Euclidean distance.

Given $0 \leq \alpha \leq 2$ then $0 \leq \alpha\beta \leq 2$ provided that $0 < \beta \leq 1$, we have:

$$\begin{aligned}
&-2 \leq -\alpha\beta \leq 0 \\
&-1 \leq 1 - \alpha\beta \leq 1 \\
&|1 - \alpha\beta| \leq 1 \\
&|v_i - z_i||1 - \alpha\beta| \leq |v_i - z_i| \quad \text{where} \quad (|v_i - z_i| > 0) \\
&|v_i - z_i| \geq |(v_i - z_i)(1 - \alpha\beta)| \\
&= |v_i - \alpha\beta v_i - z_i + z_i\alpha\beta| \\
&= |\alpha\beta(z_i - v_i) + v_i - z_i|
\end{aligned}$$

Algorithm 1. SOStream(DS, α, $MinPts$)

```
1  SOStream ← NULL;
2  foreach v_t ∈ DS do
3      win ← minDist{ v_t, M(t) };
4      if |M(t)| ≥ MinPts then
5          winN ← findNeighbors(win, MinPts);
6          if d(v_t, win) ≤ win.Radius then
7              updateCluster(win, v_t, α, winN);
8          else
9              newCluster(v_t);
10         overlap ← findOverlap(win, winN);
11         if |overlap| > 0 then
12             mergeClusters(win, overlap);
13     else
14         newCluster(v_t);
```

by Equation 2, $\alpha\beta(z_i - v_i) + v_i = v_i'$. Then we have:

$$|v_i - z_i| \geq |v_i' - z_i|, \quad \text{which implies :}$$
$$0 < \alpha \leq 2 \Rightarrow |v_i - z_i| \geq |v_i' - z_i|$$
$$(v_i - z_i)^2 \geq (v_i' - z_i)^2$$

This shows that if $0 < \alpha\beta \leq 2$ then each dimension in the modified cluster centroid will move closer to the winner centroid.

3.3 SOStream Algorithm

We are finally ready to discuss the SOStream algorithm in more detail. Here we decompose SOStream into seven basic algorithms. Algorithm 1 is the main algorithm and performs its main loop (step 2) for each input data point in the original data stream (DS). When a new input vector is obtained, the winner cluster is identified, its neighbors are found and either clusters are merged, the winning cluster is updated, or a new cluster is created.

Algorithm 2 returns all the neighbors of the winning cluster as well as its computed radius (threshold) at line 8. If the size of the neighborhood satisfies MinPts then, Algorithm 3 is called to find clusters that overlap with the winner. For each overlapping cluster its distance is calculated to the winning cluster. Any clusters with a distance less than that of the merge-threshold will be merged with the winner. This process can be triggered at regular intervals or if there is any shortage of memory.

Algorithm 5 is called to update an existing cluster. If the neighborhood of the winning cluster does not have a sufficient number of nearest neighbors or the input data $v(t)$ does not lie within the radius of the winning cluster then, Algorithm 6 is called to create a new cluster and add it to the model $M(t) \leftarrow M(t) \cup \{ v(t) \}$. Over time if this cluster does not succeed in attracting enough neighbors, then it will fade and we can remove

Algorithm 2. findNeighbors(win, $MinPts$)

```
if |M(t)| ≥ MinPts then
    foreach N_i ∈ M(t) do
        //Determine the distance from any
        //cluster N_i to the winner.
        winDistN ← winDistN ∪ { d(win, N_i) };
    Sort winDistN distances in ascending order;
    //kDist: represent the radius (threshold) of the winning cluster.
    kDist ← winDistN[MinPts - 1];
    win.setRadius(kDist);
    //Find the nearest neighbors for winner.
    foreach d_i ∈ winDistN do
        if d_i ≤ kDist then
            winNN ← winNN ∪ { N_i };
    return winNN;
else
    return ∅;
```

Algorithm 3. findOverlap(win, $winN$)

```
overlap ← ∅;
foreach N_i ∈ winN do
    if (win.ID()! = N_i.ID()) then
        if d(win, N_i) - (win.Radius + N_i.Radius) < 0 then
            overlap ← overlap ∪ { N_i };
return overlap;
```

it. Fading of cluster structure is used to discount the influence of historical data points. SOStream can adapt to changes in data over time, by using a decay decreasing function associated with each cluster:

$$f(t) = 2^{\lambda t} \tag{4}$$

where, λ define the rate of decay of the weight over time and $t = (t_c - t_0)$ where, t_c denote the current time and t_0 is the creation time of the cluster.

SOStream uses centroid clustering to represent the cluster center and does not store data points. The frequency count n determines the weight of each cluster. Aging is accomplished by reducing the count over time:

$$n_{i+1} = n_i 2^{\lambda t} \tag{5}$$

SOStream checks for clusters that are fading and removes any cluster that reaches a defined weight. See Algorithm 7. We do not explicitly show the fading function in Algorithm 1 as its use is optional. The fading function can be explicitly called or called at a regular time intervals which gives the flexibility and efficiency of using fading.

Algorithm 4 merges two clusters and set the new created cluster as the new winner and continue to test other clusters for merge. Based on experiments we expect the

Algorithm 4. mergeClusters(win, overlap)

```
foreach N_i ∈ overlap do
    if d(N_i, win) < mergeThreshold then
        //Equation 7 and 8 are used to merge two clusters.
        merge(N_i, win);
```

Algorithm 5. updateCluster(win, v_t, α, $winN$)

```
//This method incrementally update the centroid of the winner.
win.updateCentroid(v_t);
//Frequency-counter is incremented.
win.counter++;
//Modify the winning neighborhood to resemble winning cluster. The method
//adjustCentroid is computed using equation 2.
winRadius ← win.Radius;
foreach N_i ∈ winN do
    widthN ← (winRadius)^2;
    influence ← exp(- d(N_i, win) / (2 widthN));
    N_i.adjustCentroid(win.getCentroid(), α, influence);
```

number of clusters to be small (relative to the number of data points) as the merge/fade is happening online.

3.4 Online Merging

With data stream clustering, the creation of clusters in a quick online manner may result in many small micro-clusters. As a result, many earlier stream clustering algorithms created special offline components to perform a merging of similar micro-clusters into larger clusters. In SOStream, merging is efficiently performed online at each time step as an integral part of the algorithm by only considering the neighborhood of the winning cluster.

Centroid clustering is a well known clustering technique, where the centroid is the mean of all the points within the cluster. In data stream, incoming data points are incrementally clustered with the centroid of the nearest cluster. Over time the clusters change their original position and may result in overlapping with other clusters. As a result, clusters may be merged into one cluster. Recall that each cluster N_i has a radius r_i associated with it. Two clusters are said to overlap if the spheres in d-dimensional space defined by the radius of each cluster overlap. We merge clusters if they overlap with a distance that is less than the merge-threshold. Hence, the threshold value is a determining factor for the number of clusters. The impact of this threshold will be analyzed in section 4.

Merging procedure: Let S represent a set of clusters from the neighborhood of the wining cluster $S = \{ N_1, N_2, ..., N_k \}$. Two clusters in the neighborhood S are said to overlap if

$$d(C_i, C_j) - (r_i + r_j) < 0 \tag{6}$$

Algorithm 6. newCluster(v_t)

```
1 //Set v_t as a centroid for the new cluster.
2 N_{t+1} ← new_Cluster(v_t);
3 //Set the frequency-count to 1.
4 N_{t+1}.counter ← 1;
5 //Initialize the radius with 0. The radius gets computed only for winning clusters.
  N_{t+1}.Radius ← 0;
6 M(t+1) ← M(t) ∪ { N_{t+1} };
```

Algorithm 7. fadingAll()

```
1 foreach N_i ∈ M(t) do
2     N_i.fading(t, λ);
3     if N_i.counter < fadeThreshold then
4         remove N_i;
```

These clusters will be merged only if the distance between the two centroids is less than or equal to the threshold value. By merging, a new cluster N_y is created by first, finding the weight of each cluster and then computing the weighted centroid for the new cluster. This is achieved by

$$N_y = (w_i a_i + w_j b_i)/(w_i + w_j) \tag{7}$$

where, w_i, w_j are the number of points within cluster N_i, N_j and a_i, b_i are the i^{th} dimension of the weighted centroids.

We compute the new cluster's radius r_y dynamically by selecting the larger of both sums

$$r_y = max\{\, d(C_y, C_i) + r_i, d(C_y, C_j) + r_j \,\}. \tag{8}$$

We choose the largest radius to avoid losing any data point within the clusters.

4 Experiments

In this section we compare the performance of SOStream with two recent data stream clustering algorithms namely, MR-Stream and D-Stream. Our experiments were performed using synthetic datasets and the KDD Cup'99 dataset which was also used for evaluation in [8,9,6,5]. SOStream is implemented in C++ and the experiments are conducted on a machine with an Intel Centrino Duo 2.2 GHz processor and Linux Ubuntu 9.10 ($x86\_64$) as the operating system. For our test we selected the input parameters α, λ and $MinPts$ where SOStream provided the best results. In the following subsections we evaluate the ability of SOStream to detect clusters in evolving data streams and the resulting cluster quality.

4.1 Synthetic Data

In this test we generate a synthetic data stream to demonstrate SOStream's capacity of distinguishing overlapping clusters.

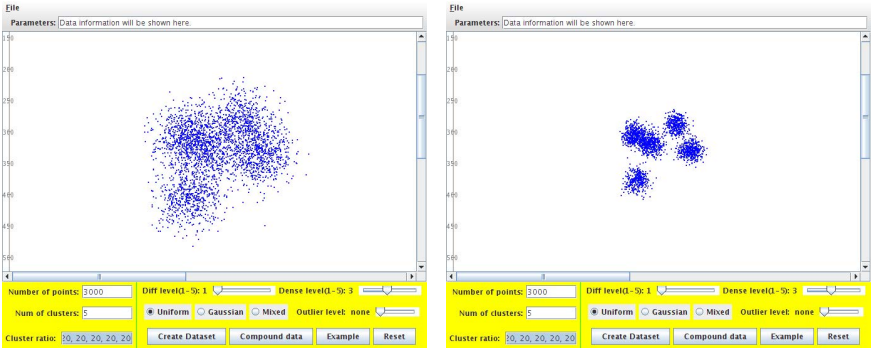


Fig. 1. (a) Data points of stream with 5 overlapping clusters and (b) Show SOStream capability to distinguish overlapped clusters. For visualizing cluster structure, we do not utilize Fading or Merging.

We used the dataset generator described in [15]. This generator is written in Java and we can control the density of the cluster, data size and noise (outlier) level. In Figure 1, the synthetic dataset has 3000 data points with no noise (outliers) added. It contains five convex-shaped clusters that overlap. It can be noted from Figure 1(b) that SOStream is able to detect the five clusters and also we can observe that the clusters are clearly separated. This is due to the use of CURE and SOM-like updating of centroids which is the most significant innovation of SOStream versus previous data stream clustering algorithms. Based on experiments we selected $\alpha = 0.1$ and $MinPts = 2$.

4.2 Real-World Dataset

Many recent streaming algorithms, such as [8,9,6,5] have been tested with KDD CUP'99 dataset to evaluate its performance. We use the same dataset to evaluate the clustering quality of SOStream algorithm. This dataset was developed with an effort to examine Network Intrusion Detection System in the Air Force base network [4]. It embeds realistic attacks into the normal network traffic. In this dataset there are a total of 42 available attributes out of which only 34 continuous attributes were considered. We compare SOStream with MR-Stream as it has been shown that MR-Stream outperformed D-Stream and CluStream. We are using the same clustering quality comparison method used by [5] which is an evaluation method that computes the average purity defined by [9,6]:

$$\text{purity} = \frac{\sum_{i=1}^{K} \frac{|N_i^d|}{|N_i|}}{K} 100\% \tag{9}$$

where K is the number of real clusters, $|N_i^d|$ is the number of points that dominate the cluster label within each cluster, and $|N_i|$ is the total number of points in each cluster.

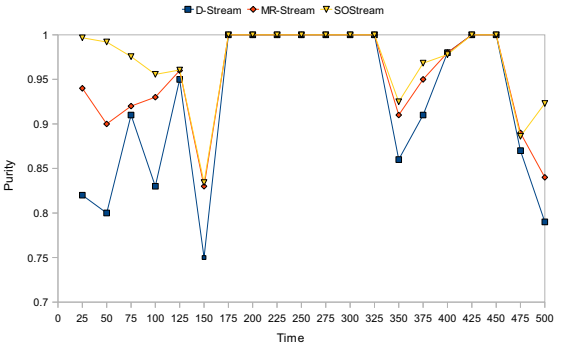


Fig. 2. SOStream clustering quality horizon = 1K, Stream speed = 1K. The quality Evaluation for MR-Stream and D-Stream is retrieved from [5].

Cluster structure fades with time which means some clusters may be deleted. This will effect the computation of purity. In order to efficiently compute the purity of the arriving data points, a predefined window (known as horizon) is used [9].

The dataset consists of 494,000 records. Using the same experimental setup as [5], the speed of the stream is set to 1000 points per second and the horizon is set to 1000 data points. There are approximately 500 time instances which are divided into intervals of 25 time instances where each instance shows an average purity value. Figure 2 shows the clustering quality comparison between SOStream, and the results of MR-Stream and D-Stream. These three algorithms all achieve an average purity of 1 between the time instance 196 and 320 since there is only one cluster appearing in one window. It can also be noticed from the graph that SOStream is able to maintain a better average purity than MR-Stream and D-Stream. The three approaches show the same pattern, this is related to the particular data set used and not to the different methods used. The overall average purity was above 95% for SOStream. In this experiment, the parameters used were again $\alpha = 0.1$, $\lambda = 0.1$ and $MinPts = 2$.

4.3 Parameter Analysis

One might consider a vector $(\alpha_1, ..., \alpha_n)$ in order to optimize the scaling factor. Because this paper focuses on a new stream clustering algorithm, we leave the optimization for future work. Table 2 shows how the parameter MinPts affects the behavior of SOStream and the average purity at different intervals of data points. For this test we used the KDD Cup'99 dataset with scaling factor, $\alpha = 0.1$. For Table 3 we changed the scaling factor parameter to $\alpha = 0.3$. On this particular test, SOStreams capability to cluster at a perfect purity level is evident for MinPts = 3. However, in order to convey and contrast SOStream improved performance against other stream clustering algorithms we used an average purity measure of different MinPts. To compare SOStream, we have derived the values for MR-Stream and D-Stream from [5]. In Table 4 the same dataset is used by MR-Stream, D-Stream and SOStream which allows us to determine the improvement

Table 2. Comparing average purity for different MinPts for $\alpha = 0.1$

	$\alpha = 0.1$			
Data Points	MinPts = 3	MinPts = 5	MinPts = 10	Mean
25000	0.983	0.990	0.921	0.965
75000	0.917	0.982	0.968	0.955
125000	0.907	0.973	1.000	0.960
175000	0.876	0.974	0.937	0.929
225000	0.876	0.974	0.937	0.929
275000	0.876	0.974	0.937	0.929
325000	0.876	0.974	0.937	0.929
375000	0.895	0.975	0.919	0.929
425000	0.907	0.975	0.963	0.949
475000	0.934	0.977	0.935	0.949
Mean	0.899	0.976	0.932	0.936

Table 3. Comparing average purity for different MinPts for $\alpha = 0.3$

	$\alpha = 0.3$			
Data Points	MinPts = 3	MinPts = 5	MinPts = 10	Mean
25000	0.999	0.938	0.914	0.950
75000	0.998	0.996	0.962	0.985
125000	0.998	0.997	0.890	0.961
175000	0.995	0.993	1.000	0.996
225000	0.995	0.993	1.000	0.996
275000	0.995	0.993	1.000	0.996
325000	0.995	0.993	1.000	0.996
375000	0.996	0.991	0.877	0.955
425000	0.996	0.992	0.941	0.977
475000	0.997	0.993	0.946	0.979
Mean	0.996	0.991	0.943	0.977

Table 4. Highlight the improvement SOStream compared to MR-Stream and D-Stream

Data Points	SOStream ($\alpha = 0.1$)	SOStream ($\alpha = 0.3$)	MR-Stream	Improvement to MR-Stream%	D-Stream	Improvement to D-Stream%
25000	0.965	0.950	0.94	2.592	0.82	15.027
75000	0.955	0.985	0.92	6.646	0.91	7.661
125000	0.960	0.961	0.96	0.000	0.95	1.182
175000	0.929	0.996	1	0	1	0
225000	0.929	0.996	1	0	1	0
275000	0.929	0.996	1	0	1	0
325000	0.929	0.996	1	0	1	0
375000	0.929	0.955	0.95	0.000	0.91	4.688
425000	0.949	0.977	1.00	-2.387	1.00	-2.387
475000	0.949	0.979	0.89	9.056	0.87	11.100
Mean	0.936	0.977	0.96	2.081	0.93	5.020

percentage between an average purity for SOStream and the results of MR-Stream and D-Stream. Over D-Stream, SOStream improves by an average of 5.0% and over MR-Stream it improves by 2.1%.

To test the merging threshold we used both quality evaluation and memory cost test on the same dataset with the matching parameters. We can observe from Figure 3 that the number of clusters residing in memory is low compared to the opposing algorithms. However, SOStream obtained a high purity (see Figure 2). This is due to the merging procedure that was presented earlier. From this study we have observed that a large merge threshold value causes the clusters to collapse and may result in only one cluster. On the other hand, a small threshold value will result in high memory cost.

4.4 Scalability and Complexity of SOStream

SOStream achieves high efficiency by storing the data structures in memory, where the updates of the stored synopses occur frequently in order to cope with the data stream. Using the same testing criteria as MR-Stream, we chose the high dimensional KDD CUP99 dataset. As we mentioned earlier, the data stream contains 494000 data points with 34 numerical attributes. We sample the memory cost every 25K records.

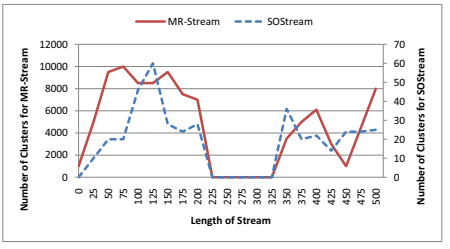


Fig. 3. SOStream memory cost over the length of the data stream. The Memory Evaluation for MR-Stream is retrieved from [5].

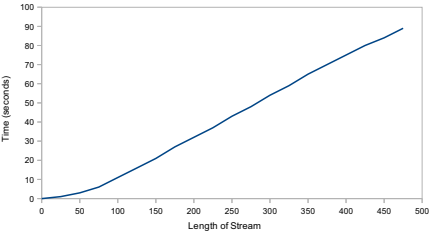


Fig. 4. SOStream execute time using high dimensional KDD CUP99 dataset with 34 numerical attributes. The sampling data rate is every 25K points.

To evaluate the memory usage of SOStream, we considered the total number of clusters in the memory. As can be seen from the Figure 3, our proposed algorithm demonstrated its low cost of memory by merging overlapping clusters which reduces the amount of space and time. In this experiment, the parameters used are $\alpha = 0.1$, MinPts = 2, fading and merging threshold = 0.1.

Figure 3 shows the memory utilized by SOStream and MR-Stream in terms of total number of clusters currently created. Our result indicates that SOStream utilizes less memory than MR-Stream. Also from Figure 3 we observe that the memory utilization profile is similar to our result. Between 0 and 200k data points, the memory utilization increases with different clusters which depreciate the clustering purity for all three algorithms. Between 200k and 350k data points, the data stream consisted mostly of one cluster, which explains why SOStream and MR-Stream consumed almost no memory. For both SOStream and MR-Stream the memory slightly increased between 350k and 400k data points and then, decreased around 450k.

Finally, we analyze the execution time and complexity of the SOStream. One appropriate data structure for Algorithm 2 is the min-heap data structure. The computation of the radius and neighbors of the winning micro-cluster takes $O(k \log k)$. With n points, SOStream complexity is $O(nk \log k)$, where k is the number of clusters. In the worst case $k = n$. In this case SOStream is $O(n^2 \log n)$. However, most data stream clustering algorithms make sure that k does not increase unbounded which reduces the more expensive operation. Other clustering operations such as remove, update and merge take $O(k)$. As shown in Figure 4, the algorithm shows that the execution time for clustering increases linearly with respect to time and number of data points.

5 Conclusion

In this paper, we proposed SOStream, which is an efficient streaming algorithm that is capable of distinguishing overlapping cluster in an online manner. The novel features of SOStream are the use of density based centroids, and an adaptive threshold. In addition, everything needed for stream clustering operations are included in a simple online algorithm. Our results show that SOStream outperformed MR-Stream and D-Stream in

terms of purity and memory utilization. We are currently working on the development of a split clusters algorithm and creating outlier detection techniques based on SOStream.

References

1. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. of 2nd International Conference on Knowledge Discovery and, pp. 226–231 (1996)
2. Kohonen, T.: Self-Organized Formation of Topologically Correct Feature Maps. Biological Cybernetics 43, 59–69 (1982)
3. Guha, S., Rastogi, R., Shim, K.: Cure: an efficient clustering algorithm for large databases. SIGMOD Rec. 27, 73–84 (1998)
4. Hettich, S., Bay, S.D.: The UCI KDD Archive, University of California, Department of Information and Computer Science, Irvine, CA, USA (1999), http://kdd.ics.uci.edu
5. Wan, L., Ng, W.K., Dang, X.H., Yu, P.S., Zhang, K.: Density-based clustering of data streams at multiple resolutions. ACM Trans. Knowl. Discov. Data 3, 14:1–14:28 (2009)
6. Chen, Y., Tu, L.: Density-based clustering for real-time stream data. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2007, pp. 133–142. ACM, New York (2007)
7. Udommanetanakit, K., Rakthanmanon, T., Waiyamai, K.: E-Stream: Evolution-Based Technique for Stream Clustering. In: Alhajj, R., Gao, H., Li, X., Li, J., Zaïane, O.R. (eds.) ADMA 2007. LNCS (LNAI), vol. 4632, pp. 605–615. Springer, Heidelberg (2007)
8. Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for clustering evolving data streams. In: Proceedings of the 29th International Conference on Very Large Databases, VLDB 2003, vol. 29, 81–92. VLDB Endowment (2003)
9. Cao, F., Ester, M., Qian, W., Zhou, A.: Density-based clustering over an evolving data stream with noise. In: 2006 SIAM Conference on Data Mining, pp. 328–339 (2006)
10. Tasoulis, D.K., Ross, G., Adams, N.M.: Visualising the Cluster Structure of Data Streams. In: Berthold, M., Shawe-Taylor, J., Lavrač, N. (eds.) IDA 2007. LNCS, vol. 4723, pp. 81–92. Springer, Heidelberg (2007)
11. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: Ordering points to identify the clustering structure. ACM SIGMOD Record 28(2), 49–60 (1999)
12. Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for projected clustering of high dimensional data streams. In: Proceedings of the Thirtieth International Conference on Very Large Databases, VLDB 2004, vol. 30, pp. 852–863. VLDB Endowment (2004)
13. Tasoulis, D.K., Adams, N.M., Hand, D.J.: Unsupervised clustering in streaming data. In: Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops, pp. 638–642. IEEE Computer Society, Washington, DC (2006)
14. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: an efficient data clustering method for very large databases. In: Proc. of the ACM SIGMOD Intl. Conference on Management of Data (SIGMOD), pp. 103–114 (1996)
15. Pei, Y., Zaïane, O.: A synthetic data generator for clustering and outlier analysis. Technical report, Computing Science Department, University of Alberta, Edmonton, Canada T6G 2E8 (2006)

Clustering Data Stream by a Sub-window Approach Using DCA

Minh Thuy Ta, Hoai An Le Thi, and Lydia Boudjeloud-Assala

Laboratory of Theoretical and Applied Computer Science (LITA)
UFR MIM, University of Lorraine,
Ile du Saulcy, 57045 Metz, France
minhthuy.ta@umail.univ-metz.fr,
{hoai-an.le-thi,lydia.boudjeloud-assala}@univ-lorraine.fr

Abstract. Data stream is one emerging topic of data mining, it concerns many applications involving large and temporal data sets such as telephone records data, banking data, multimedia data,... For mining of such data, one crucial strategy is analysis of packet data. In this paper, we are interested in an exploratory analysis of strategies for clustering data stream based on a sub-window approach and an efficient clustering algorithm called DCA (Difference of Convex functions Algorithm). Our approach consists of separating the data on different sub-windows and then apply a DCA clustering algorithm on each sub-window. Two clustering strategies are investigated: global clustering (on the whole data set) and independent local clustering (i.e. clustering independently on each sub-window). Our aims are study: (1) the efficiency of the independent local clustering, and (2) the adequation of local clustering and global clustering based on the same DCA clustering algorithm. Comparative experiments with clustering data stream using K-Means, a standard clustering method, on different data sets are presented.

Keywords: Clustering, Data stream, DCA, Sub-windows approach.

1 Introduction

A data stream is an ordered sequence of points $(x_1, x_2, ..., x_n)$ that must be accessed in order and that can be read only once or a small number of times. Each reading of the sequence is called a *linear scan* or a *pass* [18]. The volume of such data is so large that it may be impossible to store the data on disk, or even when the data can be stored, it is impossible to mining on the whole data set. The stream model is motivated by emerging applications involving massive data sets (telephone records, customer click streams, multimedia data, financial transactions,...). The data patterns may evolve continuously and depend on time or depend on the events; therefore, the data stream poses some special challenges for data mining algorithms. Not only by the huge volume of data streams, but also by the fact that the data in the streams are temporally correlated. Such

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 279–292, 2012.

temporal correlations can help detect the important data evolution characteristics, as a result of which, it is necessary to design the efficient stream mining algorithms.

For mining on stream data, one crucial and efficient strategy is analysis of packet data. The data set is divided into more significant sub-periods, called sub-windows, with the aim of detect data patterns evolution or emergence, which would not have been revealed by a global analysis in whole time period. A mining method is then applied on each sub-window. This strategy allows reduce costs (physical memory, CPU time, etc.) as we focus on the analysis of a portion of the data.

The sub-window approaches are studied in [1] where some strategies of clustering data stream are studied: global clustering, independent local clustering, dependent local clustering. There, the data set is divided into sub-windows on each of which the K-Means based algorithm is then applied, and web usage data sets have been tested.

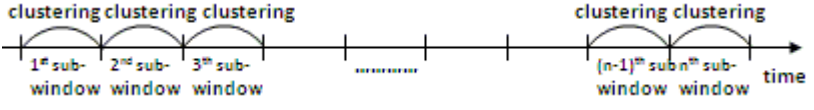


Fig. 1. Analysis on significant sub-windows (ref. [1])

In this paper, we are interested in an exploratory analysis of the independent local clustering strategy for clustering data stream based on a sub-window approach. Obviously, such an analysis depends on the performance of clustering algorithm to be used. We investigate in this work an efficient clustering algorithm in the nonconvex programming framework called DCA. Our approach consists of separating the data on different sub-windows and then apply DCA clustering algorithm on each sub-window. Our aims are study:

- the efficiency of the independent local clustering strategy, say the adequation of this local clustering and real clusters,
- the adequation between the independent local clustering and global clustering strategies based on the same DCA clustering algorithm.

Comparative experiments with clustering data stream using K-Means, a standard clustering method, on different data sets are reported.

The article is organized as follows: in Section 2 the two strategies of clustering data stream are presented. Section 3 introduces DC programming and DCA, as well as a DCA clustering algorithm. Numerical experiments on real data sets are reported in Section 4. Finally, Section 5 concludes the paper.

2 Clustering Data Stream Based on Sub-windows

Two sub-window approaches are proposed for data stream in the literature: logical window and sliding window. In this paper we adopt the logical window

approach in which data set is subdivided into separated sub-windows having the same number of clusters, and the number of elements in each window is given.

In this paper, we are concerned with an analysis of the two clustering strategies - global clustering and independent local clustering.

1. **Global clustering**: the clustering is performed on the whole data set.
2. **Independent local clustering**: the data set is subdivided into separated sub-windows, and the clustering is performed independently on each sub-window from a random starting point.

According to these strategies, two approaches are considered in our experiments in order to analyze the efficiency of the independent local clustering approach.

2.1 The First Analysis: Comparison between Independent Local Clustering and Real Clustering

For this study, we divide data set into n sub-windows by ratio of the elements in each group, and then perform clustering on each sub-window from a random starting point. By comparison with the real clustering, we get the percent of the bad placed objects (PBPO).

The schema given in Figure 2 describes this approach.

2.2 The Second Analysis: The Adequation between Independent Local Clustering and Global Clustering with the Same Clustering Algorithm

In the first step, we apply a clustering algorithm on the whole data set (global clustering). We get the results of global clustering with k groups and the ratio number of elements in each group. After that, we divide data set into n sub-windows by this ratio number of elements. Then, we perform a clustering algorithm on each sub-window (independent local clustering) and compare the results with the global clustering (see the schema given in Figure 3).

3 A DCA Clustering Algorithm

The efficiency of clustering data stream depends on clustering algorithm performed in each window. In this paper we investigate a clustering algorithm in the nonconvex programming framework based on DC (Difference of Convex functions) programming and DCA (DC Algorithm). DC programming and DCA were introduced by Pham Dinh Tao in their preliminary form in 1985. They have been extensively developed since 1994 by Le Thi Hoai An and Pham Dinh Tao and become now classic and more and more popular (see, e.g. the references given in [6]). Constituting the backbone of non-convex programming and global optimization which have very active developments in the two last decades, DCA has been successfully applied to many large-scale (smooth or non-smooth) nonconvex programs in various domains of applied sciences, in particular in data

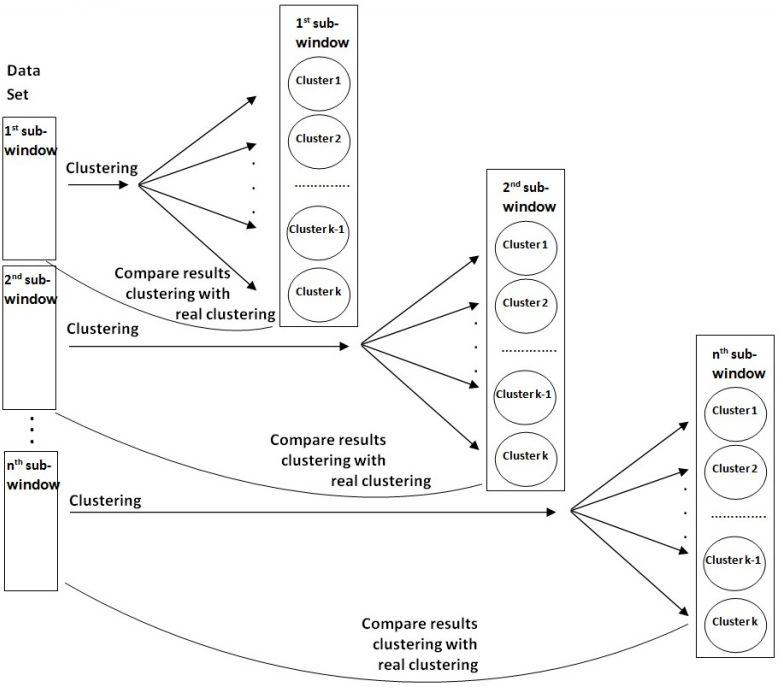


Fig. 2. Analysis of the efficiency of independent local clustering (ref. [1])

analysis and data mining, for which it provided quite often a global solution and proved to be more robust and efficient than standard methods (see [6] - [13], [15] - [17] and references therein). It is worth noting that the reference majorization algorithm developed by de Leeuw [4] for solving the Euclidean MDS problem, and the well known Convex Concave Procedure (CCCP) [22] as well as the Successive Linearization Algorithm (SLA) [3] in Machine Learning are special case of DCA.

3.1 General DC Programs

Consider the general DC program

$$\alpha = \inf\{f(x) := g(x) - h(x) \ : \ x \in \mathbb{R}^n\} \quad (P_{dc})$$

with g, h being proper lower semi-continuous convex functions on $\mathbb{R}^n$. Such a function f is called DC function, and $g - h$, DC decomposition of f while the convex functions g and h are DC components of f. It should be noted that a constrained DC program whose feasible set C is convex can always be transformed into an unconstrained DC program by adding the indicator function χ_C of C ($\chi_C\ (x) = 0$ if $x \in C, +\infty$ otherwise) to the first DC component g.

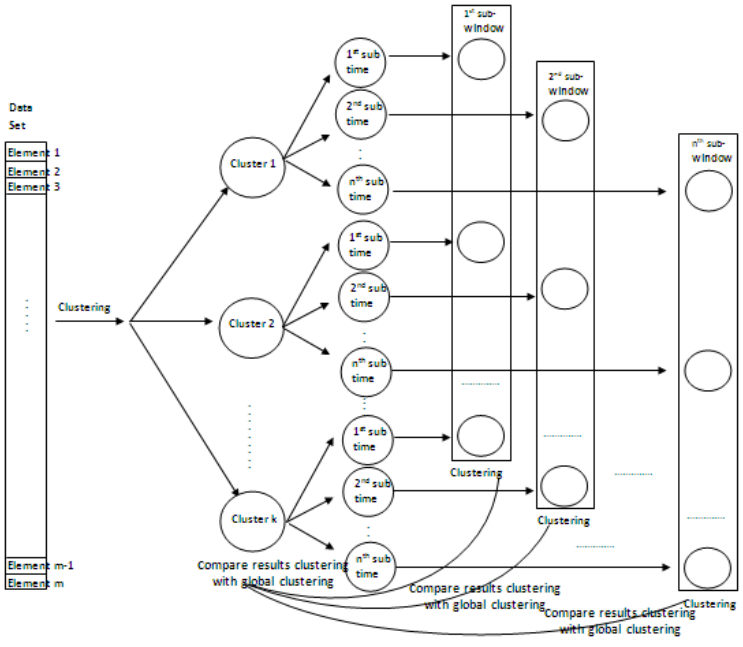


Fig. 3. Analysis of the adequation between independent local clustering and global clustering (ref. [1])

Recall that, for a convex function θ and a point $x_0 \in dom\ \theta := \{x \in \mathbb{R}^n : \theta(x_0) < +\infty\}$, $\partial\theta(x_0)$ denotes the subdifferential of θ at x_0,i.e.,

$$\partial\theta(x_0) := \{y \in \mathbb{R}^n : \theta(x) \geq \theta(x_0) + \langle x - x_0, y\rangle, \forall x \in \mathbb{R}^n\} \tag{1}$$

The necessary local optimality condition for (P_{dc}) is

$$\partial h(x^*) \subset \partial g(x^*). \tag{2}$$

A point that x^* verifies the generalized Kuhn-Tucker condition

$$\partial h(x^*) \cap \partial g(x^*) \neq \emptyset \tag{3}$$

is called a critical point of $g - h$.

3.2 Difference of Convex Functions Algorithms (DCA)

The main idea behind DCA is to replace, at the current point x^k of iteration k, the concave part $-h(x)$ with its affine majorization defined by

$$h_k(x) := h(x^k) + \left\langle x - x^k, \gamma^k \right\rangle, \quad \gamma^k \in \partial h(x^k)$$

to obtain the convex program of the form

$$\inf\{g(x) - h_k(x) : \ x \in \mathbb{R}^n\} \Longleftrightarrow \inf\{g(x) - \left\langle x, \gamma^k \right\rangle \ : x \in \mathbb{R}^n\}. \quad (P_k)$$

In fact, DCA is an iterative primal-dual subgradient method, but for simplicity we omit here the DC duality and the dual part of DCA. The generic DCA scheme is shown below. Note that the DCA is constructed from DC components g and h and their conjugates but not from the DC function f itself. Furthermore, DCA is a descent method without line-search which has linear convergence for general DC programs.

Algorithm
Let $x^0 \in \mathbb{R}^n$ be an initial guess. Set $k := 0$
REPEAT
$\gamma^k \in \partial H(x^k)$.
$x^{k+1} \in \text{argmin}\{G(x) - \langle x, \gamma^k \rangle : x \in IR^n\}$.
$k = k + 1$
UNTIL convergence

DCA schemes have the following properties ([10,16]):

i) the sequence $\{g(x^k) - h(x^k)\}$ is decreasing,
ii) if the optimal value α of problem (P_{dc}) is finite and the infinite sequences $\{x^k\}$ is bounded, then every limit point $\widetilde{x}$ of the sequence $\{x^k\}$ is a critical point of $g - h$.

Observe that a DC function has infinitely many DC decompositions and there are as many DCA as there are DC decompositions which have crucial impacts on the qualities (speed of convergence, robustness, efficiency, and globality of computed solutions) of DCA. Hence, the solution of a nonconvex program by DCA must be composed of two stages: the search of an appropriate DC decomposition and that of a good initial point.

3.3 DCA for Solving the Clustering Problem via MSSC (Minimum Sum of Squares Clustering) Formulation

An instance of the partitional clustering problem consists of a data set $\mathcal{A} := \{a^1, ...a^m\}$ of m points in $\mathbb{R}^n$, a measured distance, and an integer k; we are to choose k members x^ℓ $(l = 1, ...k)$ in $\mathbb{R}^n$) as "centroid" and assign each member of $\mathcal{A}$ to its closest centroid. The assignment distance of a point $a \in \mathcal{A}$ is the distance from a to the centroid to which it is assigned, and the objective function, which is to be minimized, is the sum of assignment distances. If the squared Euclidean distance is used, then the corresponding optimization formulation is expressed as ($\|.\|$ denotes the Euclidean norm) a so called MSSC problem

$$\min \quad \left\{ \sum_{i=1}^{m} \min_{\ell=1,\ldots,k} \left\| x^\ell - a^i \right\|^2 : x^\ell \in \mathbb{R}^n, \ell = 1, \ldots, k \right\}. \quad (MSSC)$$

The DCA applied to (MSSC) has been developed in [12]. For the reader's convenience we will give below a brief description of this method.

To simplify related computations in DCA for solving problem (MSSC) we will work on the vector space $\mathbb{R}^{k \times n}$ of $(k \times n)$ real matrices. The variables are then X

$\in \mathbb{R}^{k\times n}$ whose i^{th} row X_i is equal to x^i for $i = 1, ..., k$. The Euclidean structure of $\mathbb{R}^{k\times n}$ is defined with the help of the usual scalar product

$$\mathbb{R}^{k\times n} \ni X \longleftrightarrow (X_1, X_2, \ldots, X_k) \in (\mathbb{R}^n)^k,\ X_i \in \mathbb{R}^n, (i = 1, .., k),$$
$$\langle X, Y\rangle \ : = Tr(X^T Y) = \sum_{i=1}^{k} \langle X_i, Y_i\rangle$$

and its Euclidean norm $\|X\|^2 := \sum_{i=1}^{k} \langle X_i, X_i\rangle = \sum_{i=1}^{k} \|X_i\|^2$ (Tr denotes the trace of a square matrix). We will reformulate the MSSC problem as a DC program in the matrix space $\mathbb{R}^{k\times n}$ and then describe DCA for solving it.

a) Formulation of (MSSC) as a DC program

According to the property

$$\min_{\ell=1,\ldots,k} \left\|x^\ell - a^i\right\|^2 = \sum_{\ell=1}^{k} \left\|x^\ell - a^i\right\|^2 - \max_{r=1,\ldots,k} \sum_{\ell=1,\ell\neq r} \left\|x^\ell - a^i\right\|^2$$

and the convexity of the functions

$$\sum_{\ell=1}^{k} \left\|x^\ell - a^i\right\|^2 \quad , \max_{r=1,\ldots,k} \sum_{\ell=1,\ell\neq r} \left\|x^\ell - a^i\right\|^2,$$

we can say that (MSSC) is a DC program with the following DC formulation:

$$(MSSC) \Leftrightarrow \min \quad \{F(X) := G(X) - H(X) :\ X \in \mathbb{R}^{k\times n}\}, \tag{4}$$

where the DC components G and H are given by

$$G(X) = \sum_{i=1}^{m}\sum_{\ell=1}^{k} G_{i\ell}(X), G_{i\ell}(X) = \frac{1}{2}\left\|X_\ell - a^i\right\|^2 \text{ for } i = 1, \ldots, m, \ell = 1, \ldots, k \tag{5}$$

and

$$H(X) \ = \sum_{i=1}^{m} H_i(X),\ H_i(X) = \max_{j=1,\ldots,k} H_{ij}(X); \tag{6}$$

$$H_{ij}(X) := \sum_{\ell=1,\ell\neq j}^{k} \frac{1}{2}\left\|X_\ell - a^i\right\|^2 \text{ for } i = 1, \ldots, m. \tag{7}$$

It is interesting to note that the function G is a strictly convex quadratic form. More precisely we have, after simple calculations:

$$G(X) = \frac{m}{2}\|X\|^2 - \langle B, X\rangle + \frac{k}{2}\|A\|^2 \tag{8}$$

where $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{k\times n}$ are given by

$$\begin{aligned} A_i &:= a^i \quad \text{for} \quad i = 1,\ldots,m \\ B_\ell &:= a = \sum_{i=1}^{m} a^i \text{ for } \ell = 1,\ldots,k. \end{aligned} \tag{9}$$

In the matrix space $\mathbb{R}^{k\times n}$, the DC program (4) then is minimizing the difference of the simplest convex quadratic function (8) and the nonsmooth convex one (6). This nice feature is very convenient for applying DCA, which consists in solving a sequence of approximate convex quadratic programs whose solutions are explicit.

b) DCA for solving (4)

According to the description of DCA, determining the DCA scheme applied to (4) amounts to computing the two sequences $\{X^{(p)}\}$ and $\{Y^{(p)}\}$ in $\mathbb{R}^{k\times n}$ such that

$$Y^{(p)} \in \partial H(X^{(p)}), X^{(p+1)} \in \partial G^*(Y^{(p)}).$$

We shall present below the computation of $\partial H(X)$ and $\partial G^*(Y)$.

We first express the convex function H_{ij} by

$$\begin{aligned} H_{ij}(X) &= \sum_{\ell=1}^{k} \frac{1}{2}\left\|X_\ell - a^i\right\|^2 - \frac{1}{2}\left\|X_j - a^i\right\|^2 \\ &= \frac{1}{2}\left\|X - A^{[i]}\right\|^2 - \frac{1}{2}\left\|X_j - a^i\right\|^2 \end{aligned} \tag{10}$$

where $A^{[i]} \in \mathbb{R}^{k\times n}$ is the matrix whose rows are all equal to a^i.

That gives

$$\nabla H_{ij}(X) = X - A^{[i]} - e_j^{[k]}(X_j - a^i) \tag{11}$$

with $\{e_j^{[k]} : j = 1,...,k\}$ being the canonical basis of $\mathbb{R}^k$.

Hence, according to (6) we get the following simpler matrix formula for computing ∂H

$$Y \in \partial H(X) \Leftrightarrow Y = \sum_{i=1}^{m} Y^{[i]} \text{ with } Y^{[i]} \in \partial H_i(X) \text{ for } i = 1,...,k, \tag{12}$$

where $Y^{[i]}$ is a convex combination of $\{\nabla H_{ij}(X) : j \in K_i(X)\}$, i.e.,

$$Y^{[i]} = \sum_{j\in K_i(X)} \lambda_j^{[i]} \nabla H_{ij}(X) \text{ with } \lambda_j^{[i]} \geq 0 \text{ for } j \in K_i(X) \text{ and } \sum_{j\in K_i(X)} \lambda_j^{[i]} = 1, \tag{13}$$

with $K_i(X) := \{j = 1,\ldots,k : H_{ij}(X) = H_i(X)\}$.

In particular we can take for $i = 1,...,m$

$$Y^{[i]} = X - A^{[i]} - e_{j(i)}^{[k]}(X_{j(i)} - a^i) \text{ for some } j(i) \in K_i(X), \tag{14}$$

and the corresponding $Y \in \partial H(X)$ defined by

$$Y = mX - B - \sum_{i=1}^{m} e_{j(i)}^{[k]}(X_{j(i)} - a^i). \tag{15}$$

Since the function G is strictly convex quadratic, its conjugate G^* is differentiable and we have from (8)

$$X = \nabla G^*(Y) \Longleftrightarrow Y = \nabla G(X) = mX - B,$$

or again

$$X = \frac{1}{m}(B + Y). \tag{16}$$

Remark 1. According to the general DCA scheme and the explicit calculations (15), (16) of the subdifferentials ∂H and ∂G^* in the DC program (4), the sequences $\{X^{(p)}\}$ and $\{Y^{(p)}\}$ generated by DCA are explicitly computed.

We are now in a position to describe the DCA for solving problem (MSSC) via the DC decomposition (4).

c) Description of DCA to solve the MSSC problem (4)

Initialization: Let $\epsilon > 0$ be given, $X^{(0)}$ be an initial point in $\mathbb{R}^{k\times n}$, set $p := 0$;

Repeat

Calculate $Y^{(p)} \in \partial H(X^{(p)})$ by using (15)

$$Y^{(p)} = mX^{(p)} - B - \sum_{i=1}^{m} e_{j(i)}^{[k]}(X_{j(i)}^{(p)} - a^i)$$

and calculate $X^{(p+1)}$ according to (16)

$$X^{(p+1)} := \frac{1}{m}(B + Y^p). \tag{17}$$

Set $p + 1 \leftarrow p$

until $\| X^{(p+1)} - X^{(p)} \| \leq \epsilon(\| X^{(p)} \| + 1)$ or $\left|F(X^{(p+1)}) - F(X^{(p)})\right| \leq \epsilon(\left|F(X^{(p)})\right| + 1)$.

Remark 2. The DC decomposition (4) gives birth to a very simple DCA. It requires only elementary operations on matrices (the sum and the scalar multiplication of matrices) and can so handle large-scale clustering problems.

Table 1. Data Sets

Dataset	Points	Dimension	No. clusters	No. windows
Iris	150	4	3	2
Ionosphere	351	34	2	3
Pima	768	8	2	5
Wisconsin Breast	683	34	2	5
Waveform	5000	21	3	5
Magic	19020	10	2	7

4 Numerical Experiments

We consider 6 real-world data sets: *Iris, Ionosphere, Pima, Wisconsin Breast, Waveform, Magic* from [19].The global clustering and local clustering strategies based on DCA and K-Means algorithms have been implemented in the Visual C++ 2008, and run on a PC Intel i5 CPU650, 3.2 GHz of 4GB RAM. The information about data sets is summarized in table 1.

The First Experiment
First, we divide data set into n sub windows by ratio of the elements in each group. For example, the *Ionosphere* data set has 351 elements partitioned in 2 groups: the first group has 125 elements; the second group has 226 elements. If we want to obtain 3 windows, we chose randomly 41 (125/3) elements in the first group and chose randomly 75 (226/3) elements in the second group to compose the window 1. We construct windows 2 and 3 using the similar process.

After, we perform the clustering algorithm on each sub-window. We get the percent of the bad placed objects (PBPO) in comparing with real clusters.

The table 2 presents the comparative results of independent local clustering strategy based on DCA and K-Means algorithm. Here "Min PBPO" (resp. "Max PBPO") denotes the minimum (resp. the maximum) of PBPO over 10 executions of DCA and/or K-Means algorithm on each sub-window. For studying the efficiency of clustering algorithm we also report the results of global clustering (clustering on the whole data set) in comparing with real clusters. The starting points of DCA and K-Means are the same and they are randomly chosen from given objects.

From the results reported in Table 2 we observe that

i) The independent local clustering strategy is as efficient as the global clustering strategy, in comparing with the real clusters. In other words, clustering by logical sub-windows approach can be efficiently used for clustering data stream and/or clustering on mass of data.
ii) Clustering of stream data based on DCA is in general better than the one based on K-Means.

The Second Experiment
As in the first experiment, we perform clustering with two algorithms: DCA and K-Means. We are interested in the adequation between local and global

Table 2. Comparative results of global and independent local clustering strategies using DCA and K-Means

Iris	K-Means Min PBPO	DCA Min PBPO	K-Means Max PBPO	DCA Max PBPO	K-Means Avg PBPO	DCA Avg PBPO	K-Means Avg Time	DCA Avg Time
Window1	10.6667%	**10.0000%**	47.000%	**46.5200%**	18.4000%	**18.2000%**	0.0020	0.0016
Window2	12.0000%	**4.0000%**	52.0000%	**45.3333%**	15.0000%	15.0000%	0.0026	0.0027
Dataset	10.6667%	**8.0000%**	54.6670%	**49.3333%**	18.4000%	**16.2000%**	0.0025	0.0038
Wisconsin Breast	K-Means Min PBPO	DCA Min PBPO	K-Means Max PBPO	DCA Max PBPO	K-Means Avg PBPO	DCA Avg PBPO	K-Means Avg Time	DCA Avg Time
Window1	2.2059%	2.2059%	2.2059%	2.2059%	2.2059%	2.2059%	0.0066	0.0070
Window2	1.4716%	1.4716%	1.4716%	1.4716%	1.4716%	1.4716%	0.0005	0.0058
Window3	5.1471%	5.1471%	5.8824%	5.8824%	5.5147%	**5.3677%**	0.0052	0.0062
Window4	5.1471%	**3.6765%**	6.6177%	6.6177%	5.8088%	**5.0735%**	0.0050	0.0062
Window5	4.3166%	4.3166%	5.0360%	**4.3166%**	4.4604%	**4.3166%**	0.0054	0.0062
Dataset	3,95315%	3.95315%	4,09956%	4.09956%	4.07028%	**4.0410%**	0.0091	0.0137
Ionosphere	K-Means Min PBPO	DCA Min PBPO	K-Means Max PBPO	DCA Max PBPO	K-Means Avg PBPO	DCA Avg PBPO	K-Means Avg Time	DCA Avg Time
Window1	25.8621%	25.8621%	34.4828%	**32.7586%**	28.7931%	**27.8448%**	0.0100	0.0130
Window2	32.7586%	32.7586%	34.4828%	**33.6207%**	33.6207%	**33.2759%**	0.0110	0.0120
Window3	29.4118%	29.4118%	35.2941%	35.2941%	**33.1092%**	33.4454%	0.0110	0.0120
Dataset	29.0598%	29.0598%	35.3276%	**29.3447%**	29.8006%	**29.1738%**	0.0140	0.0188
Pima	K-Means Min PBPO	DCA Min PBPO	K-Means Max PBPO	DCA Max PBPO	K-Means Avg PBPO	DCA Avg PBPO	K-Means Avg Time	DCA Avg Time
Window1	32.6797%	32.6797%	33.3333%	**32.6797%**	32.8976%	**32.6797%**	0.0103	0.0150
Window2	32.6797%	32.6797%	32.6797%	32.6797%	32.6797%	32.6797%	0.0107	0.1533
Window3	35.9477%	**34.6405%**	35.9477%	**35.5120%**	35.9477%	**35.5120%**	0.0053	0.0100
Window4	32.0261%	32.0261%	34.6405%	**33.3333%**	33.3333%	**32.8976%**	0.0053	0.0100
Window5	32.6923%	32.6923%	32.6923%	32.6923%	32.6923%	32.6923%	0.0053	0.0153
Dataset	33.9844%	**33.7240%**	34.0144%	**33.9844%**	34.0175%	**33.9583%**	0,0116	0.0170
Waveform	K-Means Min PBPO	DCA Min PBPO	K-Means Max PBPO	DCA Max PBPO	K-Means Avg PBPO	DCA Avg PBPO	K-Means Avg Time	DCA Avg Time
Window1	50.2503%	**46.3463%**	50.6507%	**48.8488%**	50.4171%	**47.2472%**	0.0830	0.1093
Window2	48.4484%	**47.1471%**	**48.9489%**	59.0591%	**48.6820%**	51.1512%	0.0833	0.1143
Window3	48.8488%	**22.8228%**	51.3514%	**50.6507%**	50.3837%	**39.9399%**	0.0880	0.1197
Window4	48.9489%	**48.3483%**	60.4605%	**50.5910%**	53.4535%	**39.9399%**	0.0987	0.1147
Window5	**46.9124%**	47.1116%	52.7888%	**50.8964%**	53.4535%	**49.8165%**	0.0830	0.1247
Dataset	49.8200%	**47.5400%**	60.7000%	**52.5800%**	51.0240%	**47.7160%**	0.1574	0.2535
Magic	K-Means Min PBPO	DCA Min PBPO	K-Means Max PBPO	DCA Max PBPO	K-Means Avg PBPO	DCA Avg PBPO	K-Means Avg Time	DCA Avg Time
Window1	**34.7570%**	35.0147%	35.2356%	**35.0147%**	35.0368%	**35.0147%**	0.2184	0.3368
Window2	35.1988%	**35.1252%**	35.6406%	**35.6143%**	35.4934%	**35.4579%**	0.2090	0.2994
Window3	**35.6038%**	35.7511%	35.8247%	**35.7511%**	35.7511%	35.7511%	0.2154	0.3714
Window4	34.3888%	**34.3520%**	34.4624%	**34.4256%**	34.4379%	**34.3765%**	0.2150	0.3216
Window5	36.0457%	36.0457%	36.0457%	36.0457%	36.0457%	36.0457%	0.2030	0.3278
Window6	34.6834%	34.6834%	34.6834%	34.6834%	34.6834%	**34.6588%**	0.1998	0.3146
Window7	35.6828%	**35.3891%**	35.6828%	**35.3891%**	35.6828%	**35.3891%**	0.2150	0.3964
Dataset	35,0894%	**35.0263%**	**35.0894%**	35.1052%	35.0894%	**35.0820%**	0.4714	0.6118

clustering strategies. We first perform the global clustering algorithm on the whole data set. After that we divide the data set on sub-windows with the same procedure used in the first experiment. In each sub-window, we run the clustering algorithm and then compare the obtained results with clusters furnished by the global clustering strategy.

Table 3 presents the comparative results of independent local clustering strategy based on DCA and K-Means algorithm. Here "Min Error" (resp. "Max

Error") denotes the minimum (resp. the maximum) of "errors" of the local clustering in comparing with clusters furnished by the global clustering over 10 executions of DCA and/or K-Means algorithm on each sub-window.

Table 3. the adequation of global clustering and independent local clustering based on DCA and K-Means

Iris	K-Means Min Error	DCA Min Error	K-Means Max Error	DCA Max Error	K-Means Avg Error	DCA Avg Error	K-Means Avg Time	DCA Avg Time
Window1	1.3333%	**0.0000%**	46.3333%	**17.3333%**	14.9333%	**7.5644%**	0.0015	0.0024
Window2	**0.0000%**	1.3333%	45.0000%	**42.6667%**	10.0000%	10.0000%	0.0008	0.0047
Wisconsin Breast	K-Means Min Error	DCA Min Error	K-Means Max Error	DCA Max Error	K-Means Avg Error	DCA Avg Error	K-Means Avg Time	DCA Avg Time
Window1	0.0000%	0.0000%	0.7353%	**0.0000%**	0.4412%	**0.0000%**	0.0094	0.0062
Window2	0.0000%	0.0000%	**0.0000%**	0.7353%	**0.0000%**	0.3676%	0.0062	0.0124
Window3	0.0000%	0.0000%	**0.0000%**	0.7353%	**0.0000%**	0.2206%	0.0062	0.0154
Window4	0.0000%	0.0000%	2.9412%	**0.0000%**	2.0588%	**0.0000%**	0.0062	0.0216
Window5	2.1583%	**0.0000%**	**2.1583%**	2.8777%	2.1583%	**0.8633%**	0.0064	0.0278
Ionosphere	K-Means Min Error	DCA Min Error	K-Means Max Error	DCA Max Error	K-Means Avg Error	DCA Avg Error	K-Means Avg Time	DCA Avg Time
Window1	2.5862%	**0.8621%**	47.4138%	**45.6900%**	16.3793%	**10.0860%**	0.0101	0.0118
Window2	0.0000%	0.0000%	48.2760%	**45.6900%**	9.2241%	**7.5862%**	0.0101	0.0237
Window3	1.6807%	**0.8403%**	**44.5380%**	47.0590%	**10.1681%**	12.1850%	0.0107	0.0353
Pima	K-Means Min Error	DCA Min Error	K-Means Max Error	DCA Max Error	K-Means Avg Error	DCA Avg Error	K-Means Avg Time	DCA Avg Time
Window1	1.9068%	**1.3072%**	12.4180%	**2.6144%**	5.4902%	**1.6994%**	0.0081	0.0082
Window2	0.6536%	0.6536%	3.9216%	**3.2680%**	1.6340%	**1.1765%**	0.0063	0.0152
Window3	5.2288%	**0.0000%**	12.4180%	**7.1896%**	10.4575%	**5.8170%**	0.0062	0.0222
Window4	0.0000%	0.0000%	18.654%	**0.0000%**	2.0915%	**0.0000%**	0.0058	0.0289
Window5	3.2051%	**2.5641%**	12.1800%	**3.8462%**	4.1027%	**3.0128%**	0.0067	0.0366
Waveform	K-Means Min Error	DCA Min Error	K-Means Max Error	DCA Max Error	K-Means Avg Error	DCA Avg Error	K-Means Avg Time	DCA Avg Time
Window1	1.4014%	**0.8008%**	**2.8028%**	3.0032%	1.9119%	**1.8622%**	0.0919	0.1404
Window2	1.0010%	**0.7007%**	**1.5015%**	4.0040%	**1.2412%**	2.0420%	0.0928	0.2797
Window3	0.6006%	**1.0010%**	**0.6006%**	2.2022%	**0.6006%**	1.6216%	0.0899	0.4180
Window4	**0.1001%**	0.3003%	**0.8008%**	5.1051%	**0.5205%**	2.7424%	0.0887	0.5520
Window5	1.5936%	**0.9961%**	**42.5300%**	48.3070%	**6.0259%**	6.4243%	0.0976	0.6879
Magic	K-Means Min Error	DCA Min Error	K-Means Max Error	DCA Max Error	K-Means Avg Error	DCA Avg Error	K-Means Avg Time	DCA Avg Time
Window1	8.1370%	**1.9514%**	10.2730%	**6.3697%**	9.5435%	**5.9278%**	0.2579	0.2963
Window2	**2.2176%**	2.4762%	7.4512%	**4.1243%**	3.4822%	**3.1842%**	0.2447	0.5807
Window3	1.9514%	**0.4050%**	2.1723%	**0.6259%**	2.1060%	**0.5817%**	0.2298	0.9000
Window4	0.9208%	**0.9205%**	**1.1419%**	1.5832%	**1.0309%**	1.4507%	0.2351	1.2713
Window5	**0.6627%**	0.7732%	1.0309%	**0.7732%**	**0.6996%**	0.7732%	0.2173	1.5880
Window6	1.8041%	**1.1046%**	1.8041%	**1.1782%**	1.8041%	**1.1267%**	0.2191	1.5880
Window7	3.4875%	**1.3216%**	3.5609%	**1.3216%**	3.5095%	**1.3216%**	0.2033	2.1687

From the results reported in Table 3 we see that

i) With an appropriate starting point, the independent local clustering strategy using DCA is very efficient in comparing with the global clustering. The Min Error varies from 0% to 2.5% over the all windows.

ii) Clustering of stream data based on DCA is in general better than the one based on K-Means. The Min Error of local clustering using K-Means varies from 0% to 8.1% over the all windows.

5 Conclusion

We have studied the clustering data stream by an logical sub-window approach using an efficient DCA based clustering algorithm. Preliminary numerical experiments show the adequation between independent local clustering and global clustering strategies. They also prove that the independent local clustering strategy using DCA can be effectively investigated for clustering data stream and clustering on mass of data. The performance of DCA suggests us to investigating it in other sub-window approaches for clustering data stream such as dependent local clustering or clustering on sliding windows. Works in these directions are in progress.

References

1. Da Silva, A.G.: Analyse des données évolutives: application aux données d'usage du Web. Thèse de Doctoral, Paris IX Dauphine, pp. 62–72 (2006)
2. Aggarwal, C.C.: Data Streams: Models and Algorithms, Advances in Database Systems, vol. 31, pp. 1–5. Springer (2007) ISBN 978-0-387-28759-1
3. Bradley, B.S., Mangasarian, O.L.: Feature selection via concave minimization and support vector machines. In: Shavlik, J. (ed.) Machine Learning Proceedings of the Fifteenth International Conferences (ICML 1998), pp. 82–90. MorganKaufmann, San Francisco (1998)
4. De Leeuw, J.: Applications of convex analysis to multidimensional scaling, Recent developments. In: Barra, J.R., et al. (eds.) Statistics, pp. 133–145. North-Holland Publishing company, Amsterdam (1997)
5. Hartigan, J.A.: Clustering algorithms. John Wiley and Sons (1975)
6. An, L.T.H.: DC programming and DCA, http://lita.sciences.univ-metz.fr/~lethi/DCA.html
7. An, L.T.H., Tao, P.D.: Solving a class of linearly constrained indefinite quadratic problems by DC algorithms. Journal of Global Optimization 11(3), 253–285 (1997)
8. An, L.T.H., Tao, P.D.: DC Programming Approach for Solving the Multidimensional Scaling Problem. In: Nonconvex Optimizations and Its Applications: Special Issue, From Local to Global Optimization, pp. 231–276. Kluwer Academic Publishers (2001)
9. An, L.T.H., Tao, P.D.: Large Scale Molecular Optimization from distances matrices by a DC optimization approach. SIAM Journal of Optimization 14(1), 77–116 (2003)
10. An, L.T.H., Tao, P.D.: The DC (difference of convex functions) Programming and DCA revisited with DC models of real world nonconvex optimization problems. Annals of Operations Research 133, 23–46 (2005)
11. An, L.T.H., Minh, L.H., Tao, P.D.: Optimization based DC programming and DCA for Hierarchical Clustering. European Journal of Operational Research 183, 1067–1085 (2007)
12. An, L.T.H., Tayeb Belghiti, M., Tao, P.D.: A new efficient algorithm based on DC programming and DCA for clustering. Journal of Global Optimization 37, 609–630 (2007)
13. An, L.T.H., Minh, L.H., Vinh, N.V., Tao, P.D.: A DC Programming approach for Feature Selection in Support Vector Machines learning. Journal of Advances in Data Analysis and Classification 2(3), 259–278 (2008)

14. MacQueen, J.B.: Some Methods for classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–288. University of California Press, Berkeley (1967)
15. Neumann, J., Schnörr, C., Steidl, G.: SVM-Based Feature Selection by Direct Objective Minimisation. In: Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A. (eds.) DAGM 2004. LNCS, vol. 3175, pp. 212–219. Springer, Heidelberg (2004)
16. Tao, P.D., An, L.T.H.: Convex analysis approach to d.c. programming: Theory, Algorithms and Applications. Acta Mathematica Vietnamica, Dedicated to Professor Hoang Tuy on the Occasion of his 70th Birthday 22(1), 289–355 (1997)
17. Tao, P.D., An, L.T.H.: DC optimization algorithms for solving the trust region subproblem. SIAM J. Optimization 8, 476–505 (1998)
18. Guha, S., Meyerson, A., Mishra, N., Motwani, R., O'Callaghan, L.: Clustering Data Streams: Theory and Practice. IEEE TKDE 15, 515–516 (2003)
19. http://archive.ics.uci.edu/ml/
20. http://faculty.washington.edu/kayee/cluster/
21. http://genomics.stanford.edu/
22. Yuille, A.L., Rangarajan, A.: The Convex Concave Procedure (CCCP). In: Advances in Neural Information Processing System 14, MIT Press, Cambrige (2002)

Improvement of K-means Clustering Using Patents Metadata

Mihai Vlase, Dan Munteanu, and Adrian Istrate

"Dunarea de Jos" University of Galati, Department of Computer Science and Engineering, Stiintei 2, Galati, Romania
{Mihai.Vlase,Dan.Munteanu,Adrian.Istrate}@ugal.ro

Abstract. Over time, many clustering methods were proposed, but there are many specific areas where adaptations, customizations and modifications of classical clustering algorithms are needed in order to achieve better results. The present article proposes a technique which uses a custom patent model, aiming to improve the quality of clustering by emphasizing the importance of various patent metadata. This can be achieved by computing different weights for different patent metadata attributes, which are considered to be valuable information.

Keywords: clustering, patents, metadata, customized weights, data model

1 Introduction

One of the most important tasks in the process of patenting is searching for similarities between patents and several types of searches are employed during this process. One of them is the preliminary search, when the invention is just an idea and inventors want to find out if a similar invention already exists. Another type of search is the "prior art" search that is performed during the writing of the patent and also afterwards. This search is conducted both by the inventors, before patent is applied for acceptance, and consequently by the patent agents that verify existing applications for patents and validate (or invalidate) them. This second type of search is used to find all patents similar to the invention that is intended to be patented and to ensure that the invention's claims have not been previously used into another patent. If this search is not done correctly and not given due importance, there could be a risky situation where another company could initiate a patent infringement lawsuit and could allege that one or more of his patent claims have been violated. When such infringement lawsuit is filed, one step in the process is to utilize another type of search, where lawyers aim to find "prior art" patents that have been omitted and were not cited previously in the patent subject matter. [1]

Because of the importance of these types of searches, users need as many sources of information and as many searching tools as possible. Because of the great number of patents existing today, most of the times a simple keyword search into a database

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 293–305, 2012.

of patents may not be satisfactory. Thus, displaying the search results sorted by certain criteria, relevant to the users, is frequently needed.

In a regular patent search engine, keyword search is done by using the data provided by users. The search results are given as a set of patents ordered by various criteria (e.g. year of publication). Usually, the number of patents resulting from such a search is very high. The use of some additional filters is frequently required in order to refine the results in order to obtain an improved relevancy.

Clustering patents allow such filtering of search results. Instead of hundreds of results that should be examined one by one, the results can be divided into patents groups with similar content. When clusters of similar documents based on extracted keywords are displayed the visible relationships among them become clearer. [2]

Over time, many special clustering algorithms have been designed to be applied in patent databases. An example of such patent analysis platform is Patent iNSIGHT Pro [3] or PatentCluster search engine [4]. The clustering used in these platforms is based on the text content of patents, especially abstracts and claims, but ignores the metadata information contained in patents.

A patent is composed of several major sections such as title, bibliographic or metadata information, abstract, detailed description of the patent, claims or references to "prior art" patents. Metadata contains important information as publication, inventor, applicant or classification as described in the next section. These metadata fields are considered valuable information that could be used to improve clustering on patents.

Often, in a search result, users are interested in identifying reference companies from a specific industry. By grouping patents from a search result in clusters, users may find relevant the identification of companies and their patents in a given cluster. This means that a special importance should be given to metadata fields in clustering.

This paper proposes a method to achieve a clustering in which the information from metadata is to be considered. This can be achieved by using a modified variant of k-means clustering algorithm and its adaptation to the particular case of patents.

The clusters thus generated, can be further used in a search engine by listing the search results from each cluster ordered by a rank [5].

This research proposes a patent model which will be used in a k-mean clustering algorithm. The patent model is constructed from a set of vectors of attributes, one vector consisting only of attributes extracted from the content of the patent and the rest of the vectors consisting of attributes extracted from various patent metadata fields. Weights for each vector of the set are calculated differently, the weights of the metadata attribute vectors having a calculated higher value. In this way the patent metadata will hold a greater influence in clustering.

2 Patent Analysis

A patent is a document which describes an invention which can be manufactured, used, and sold with the authorization of the owner of the patent. An invention is a solution to a specific technical problem. A patent document normally contains at least

one claim, the full text of the description of the invention, and bibliographic information such as the applicant's name. [6]

Patent documents have a fixed rigorous structure, containing standardized fields like patent number, applicant, inventors, assignee, technology field classification, description, claims, etc [7]. Most of this information can be found in the front page of a patent document and they are called *patent metadata* [8]. All these special and specific features of patent documents make them a valuable source of knowledge [9].

For the patent analysis, researchers are often using classifications or clustering. Supervised classification is used to group patents by a preexisting classification. Clustering, on the other hand, is the unsupervised classification of patents into groups based on similarities of the internal features or attributes.

One of the most widely used clustering algorithms, especially in text clustering, is the k-means algorithm [10-12]. Due to the variety of areas where clustering is used and because of specific conditions to each application, there are many variations of the basic algorithm proposed in the academic literature.

3 Prior Work

The applications of clustering usually deal with large data sets with many attributes. In real-life, applications do not always deal with homogeneous data. Most of the time, heterogeneous data are involved. On one hand patents may be viewed as a homogeneous data, if only the abstract and description of the patent are considered. But on the other hand patents may be looked at as heterogeneous data if, beside the abstract and the description, the metadata fields are also considered.

Regarding the clustering of heterogeneous data sets, an interesting approach was proposed by Modha & Spangler [13] where for obtaining relevant data clustering which integrate multiple, heterogeneous attribute subset in k-means clustering algorithm, they adaptively compute relative weights assigned to various attribute subset that simultaneously minimizes average intra-cluster dispersion and maximizes average inter-cluster dispersion along all attribute subsets[14]. However, in this case, the control of influence given by a subset of attributes on other subsets is lost. In the particular case of patents, a subset of attributes from a metadata field contains far fewer attributes then the subset derived from the patent description. As a result, metadata subset influence should be controlled and separately calculated.

4 K-means Algorithm

The basic idea of k-means clustering is that items are grouped into k clusters in such way that all items in same cluster are as similar to each other as possible and items not in same cluster are as different as possible. Several distance measures to calculate similarity and dissimilarity are used. One of the important concepts in k-means is the centroid: each cluster has a centroid, which can be considered as the most representative item of the cluster.

Initial k cluster centroids are chosen arbitrarily. After that, objects from the database are then distributed among the chosen clusters based on minimum distances. After all objects are distributed, the cluster centers are then updated to reflect the means of all objects into the respective cluster domains. This process is iterated while the cluster centers continue to move or objects keep switching clusters. Performance of this algorithm is influenced by the number and location of the initial cluster centers.

5 Data Model for Patents

The most common model used for information retrieval from text documents is the vector space model [15]. In this model, documents are represented as vectors of term.

In vector space model the matrix *M* is defined in equation (1), where the lines are represented by attributes and columns are represented by objects. For text documents, attributes are represented by the words dictionary from all documents, and the objects are the documents.

$$M = \begin{pmatrix} w_{11} & \cdots & w_{1j} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots & & \vdots \\ w_{i1} & \cdots & w_{ij} & \cdots & w_{in} \\ \vdots & & \vdots & \ddots & \vdots \\ w_{w1} & \cdots & w_{wj} & \cdots & w_{wn} \end{pmatrix} \quad (1)$$

The values w_{ij} of the matrix M are weights computed with term frequency – inverse document frequency (TF-IDF). TF-IDF is a weight model in which the score of a term in the document is the ratio of the number of terms in that document divided by the frequency of the number of documents in which that term occurs.

The present article proposes a particular model of patents, where text terms from description and text terms from metadata are extracted in separate vectors. The metadata vectors are treated separately, higher weights being computing for them.

For the patent documents a 3-uple is defined:

$$P = \langle P_c, P_m, M_n \rangle$$

where $P_c = \{wn_1, wn_2, \ldots wn_{nc}\}$ is the set of words from the description or abstract of a patent and nc the number of words from the description or abstract of the patent; $P_m = [mv_1, mv_2, \ldots mv_{nm}]$ is the vector of metadata values from a patent and nm the number of metadata fields from the patent; $M_n = [mn_1, mn_2, \ldots mn_{nm}]$ is the vector of metadata names from the patent.

PDB is defined as the set of all patents from database as following:

$$PDB = \{P_1, P_2, \ldots P_n\}$$

where *n* is the number of documents from database.

The words dictionary of the content of the entire set of patents is defined as

$$DC = \{wn_i \in P_{c_1} \cup P_{c_2} \ldots \cup P_{c_n}\}$$

Let $nd = |DC|$ be the number of words from the dictionary *DC*.

The words dictionary for each metadata from the patents is defined in the same way.

$$DM_{mn_t} = \{mv_{t_i} \in P_{m_1}[t] \cup P_{m_2}[t] \ldots \cup P_{m_n}[t]\}$$

where $t = 1 \ldots nm$ and $P_{m_k}[t]$ is the *t* element from vector P_m from each patent P_k.

Let $nmn_t = |DM_{mn_t}|$ be the number of elements from DM_{mn_t}.

In the vector space matrix defined for text documents (1), all weights are computed with the same function TF-IDF. For the particular case of patent documents, a vector space matrix compound from a set of subsets of attributes is proposed. Each subset contains weights calculated with a different TF-IDF function. Equation (2) defines this matrix.

$$MP = \begin{pmatrix} MC \\ MM_{mn_1} \\ MM_{mn_2} \\ \ldots \\ MM_{mn_{nm}} \end{pmatrix} \tag{2}$$

where

$$MC = \begin{pmatrix} wwn_{11} & \cdots & wwn_{1j} & \cdots & wwn_{1n} \\ \vdots & \ddots & \vdots & & \vdots \\ wwn_{i1} & \cdots & wwn_{ij} & \cdots & wwn_{in} \\ \vdots & & \vdots & \ddots & \vdots \\ wwn_{nd\ 1} & \cdots & wwn_{nd\ j} & \cdots & wwn_{nd\ n} \end{pmatrix} \tag{3}$$

and

$$MM_{mn_t} = \begin{pmatrix} wwv_{t_{11}} & \cdots & wwv_{t_{1j}} & \cdots & wwv_{t_{1n}} \\ \vdots & \ddots & \vdots & & \vdots \\ wwv_{t_{i1}} & \cdots & wwv_{t_{ij}} & \cdots & wwv_{t_{in}} \\ \vdots & & \vdots & \ddots & \vdots \\ wwv_{t_{nmn_t\ 1}} & \cdots & wwv_{t_{nmn_t\ j}} & \cdots & wwv_{t_{nmn_t\ n}} \end{pmatrix} \tag{4}$$

The merged matrix *MP* from (2) has $nd + \sum_{t=1..nm} nmn_t$ lines and *n* columns. The lines of the merged matrix *MP* represent the sum of words dictionary from all patents contents and words dictionary from all patents metadata. The columns of the merged matrix represent the patents.

For a better understanding of the proposed model, the following example is given.

Let suppose that exist 3 patents P_1, P_2, P_3. For P_1 has P_c = {"human", "antibodies", "mouse", "isotypes"} the set of words contained into the patent, P_m = {"Inventor 1", "Corporation 1"} the set of metadata values from patent and M_n= {"Inventor", "Applicant"} the metadata names. Similar P_2 is defined with P_c = {"image", "camera", "mouse", "resolution"}, P_m = {"Inventor 2", "Corporation 2"} and M_n= {"Inventor", "Applicant"}, and P_3 with P_c = {"wireless", "computer", "mouse"}, P_m = {"Inventor 3", "Corporation 2"} and M_n= {"Inventor", "Applicant"}.

The words dictionary of the content of the entire set of patents, DC = {"human", "antibodies", "mouse", "isotypes", "image", "camera", "resolution", "wireless", "computer"}.

There are 2 metadata names, so the words dictionary for "Inventor" metadata $DM_{Inventor}$ = {"Inventor 1", "Inventor 2", "Inventor 3"} and the words dictionary for "Applicant" metadata $DM_{Applicant}$ = {"Corporation 1", "Corporation 2"}.

In this particular case, the matrix MP is defined

$$MP = \begin{pmatrix} MC \\ MM_{Inventor} \\ MM_{Applicant} \end{pmatrix} =$$

		P_1	P_2	P_3
MC	human	wwn_{11}	wwn_{12}	wwn_{13}
	antibodies	wwn_{21}	wwn_{22}	wwn_{23}
	mouse	…	…	…
	isotypes	…	…	…
	image	…	…	…
	camera	…	…	…
	resolution	…	…	…
	wireless	…	…	…
	computer	wwn_{91}	wwn_{92}	wwn_{93}
$MM_{Inventor}$	Inventor 1	$wwv_{1_{11}}$	$wwv_{1_{12}}$	$wwv_{1_{13}}$
	Inventor 2	$wwv_{1_{21}}$	$wwv_{1_{22}}$	$wwv_{1_{23}}$
	Inventor 3	$wwv_{1_{31}}$	$wwv_{1_{32}}$	$wwv_{1_{33}}$
$MM_{Applicant}$	Corporation 1	$wwv_{2_{11}}$	$wwv_{2_{12}}$	$wwv_{2_{13}}$
	Corporation 2	$wwv_{2_{21}}$	$wwv_{2_{22}}$	$wwv_{2_{23}}$

6 Weighting Functions

The idea behind the data model in this paper is that the patent metadata can provide additional information for patents clustering. For instance if "Applicant" metadata have been used, by clustering using this model, the aim is to group similar patents by text and to add in each group relevant patents that belong to the applicants from that group.

Usually the applicants are specialized in one technical area, so it is very likely that the patents belonging to one applicant are from the same area of activity. Analyzing the patent database we noticed that applicants have all or the vast majority of patents applied in a single class. Therefore, the patents with the same applicant should be in the same or neighboring clusters.

At first sight, grouping by applicant should be enough and no clustering is needed. However, not all patents from an applicant are relevant to a search query, so only a specific set of patents from an applicant is needed.

Applicant's names may be considered common words in the words dictionary generated from the entire set of patents and these names may be added in the *MC* matrix. But in this case, the applicant metadata's importance is much diminished and the control over the applicant metadata's influence could be lost.

Furthermore, the metadata term frequency in the document is "1" because metadata contains a single term. In order to have an enhancement of applicant metadata in the similarity of patents, the applicant's influence should be more significant than regular words from a patent description. To accomplish this, different TFIDF functions for each subset of data are used.

The weights wwn_{ij} from the equation (3) and weights $wwv_{t_{ij}}$ from equation (6) are computed as the product of TF and IDF, but as it will be shown further, TF function is different for each set of weights:

$$wwn_{ij} = tf_{ij} * idf_i$$

$$wwv_{t_{ij}} = tf_{ij}{}' * idf_i$$

There are many well studied weighting schemes used to compute TFIDF weights [16]. One of the most common and efficient term weighting scheme used for text documents clustering is defined below:

$$tf_{ij} = c_1 + (1 - c_1) * \frac{n_{ij}}{max_i\, n_{ij}} \tag{5}$$

$$idf_i = c_2 + log\, \frac{n - n_{word(i)}}{n_{word(i)}} \tag{6}$$

where n_{ij} is the number of occurrences of word i in the text document j, and $max_i\, n_{ij}$ is the highest number of occurrences of a word in the text document j. $n_{word(i)}$ is the number of documents in which the word i occurs and $c1$ and $c2$ are constants [17].

This classical weighting scheme is used for weights wwn_{ij}.

In the classical vector space model (1), using TFIDF weighting schemes (5) and (6), the importance of a term in a document is even higher as the value of that term frequency is higher, in a more limited number of documents. Two documents are more similar if they have more terms in common and the frequencies of these terms have a closer value.

The influence of "Applicant" metadata terms is modified if the TF weight function is modified as follows:

n_{ij} term from equation (5), which represents the number of occurrences of word *i* in the text document *j*, will be replaced with $n_{ij}{}'$:

$$n_{ij}{}' = c_3 * \frac{\ln(n_{word(ij)})}{\max(\ln(n_{word(ij)}))} \tag{7}$$

where $n_{word(ij)}$ is the number of documents in which the term *i* occurs, if term *i* occurs in the text document *j*. $\max(\ln(n_{word(ij)}))$ represents the maximum of *ln* of number of documents in which a term appears, in other words represents *ln* from the number of patents issued by the applicant with the largest number of patents. c_3 is a constant.

Thereby tf_{ij} become:

$$tf_{ij}{}' = c_1 + (1 - c_1) * \frac{n_{ij}{}'}{max_i\, n_{ij}{}'} \tag{8}$$

By using the equation (7) the meaning of frequency from equation (5) has been changed. Therefore the frequency of term in documents has been changed from 1 to the *ln* of the number of documents in which a metadata term appears. *ln* is used because a very big influence to the applicants with a large number of patents should be avoided (there are applicants with hundreds of patents).

c_3 constant is used in order to weight the importance of the applicant. By increasing the value of c_3 the importance of the applicant is increased.

A value of c_3 can be calculated by considering the frequencies used in matrix *MC*. An approach could be by calculating c_3 as the maximum value for frequency of terms in all documents. Another similar approach could be by taking c_3 as the average of maximum frequency of terms in each document.

In the present model the constant c_3 value was selected as the maximum value for frequency of terms in all documents, pondered as needed with constant c_4.

$$c_3 = c_4 * \max(n_{ij})$$

By modifying the value of constant c_4 the values of a specific metadata field is controlled, and so the importance of the metadata field in the model is controlled.

Further, this customized TF-IDF model for patents was used in k-means clustering algorithm.

In this research the similarity function used in k-means algorithm is one of the common similarity functions applied for text documents, namely the cosine between the document vectors.

The results of clustering are analyzed in the next section.

7 Implementation and Evaluation

The practical implementation of the proposed model and the k-means clustering algorithm was made in Java language and for data storage and manipulation a MySQL

database was used. For the constants used in algorithms were chosen the following values:

$$c_1 = 0.8;\ c_2 = 0.2;\ c_4 = 0.25$$

For tests, samples extracted from the EPO database (European Patent Organization) were used. The patent fields available in the EPO database are: patent number, international class, title, applicant name, inventor name, representative, application year and claims From these fields have been used for tests only: patent number, international patent class, title, applicant name and claims.

International Patent Classification (IPC) provides a hierarchical system of language independent symbols for the classification of patents and utility models according to the different areas of technology to which they pertain [18]. This classification is made by human experts and is annually updated with new branches as needed.

All patents have one main class and none or more secondary classes. Often, a patent may as well belong to any class selected for the patent, but the final selection for the main class is made by the human experts, which usually are the inventors. Most of the time the class selected by the experts as main class represents the industry where the invention will be applied, therefore not all classes selected as main classes are representative for the patents.

In this research only main classes are considered.

A patent class example is A 01 B 1 / 00, where starting from left to right, each group represents a class and then a subclass. In our case A is a main class, A 01 is a subclass of A, and A 01 B is a subclass of A 01 and so on. Each patent have a main class selected that belongs to such a 5-level hierarchy.

Several tests have been run and only the representative tests are presented as described below.

For each test sample two clustering were run. First, a k-means clustering with the classic model was run (without taking into account the applicant metadata) and then a k-means clustering using the model proposed in the article was run (where the applicant metadata was taken into account).

For each clustering type 10 runs were performed and the best clustering result was stored for later analysis. As the measure for quantifying the quality of each clustering result the weighted similarity of the internal cluster similarity which is the square of the length of the centroid vectors or, in other words, the average pairwise similarity was used in each clustering algorithm.

To compare the quality of the tests results computed by the two clustering techniques, F-measure was selected. The reference used to compute F-measure is represented by classes derived from the existing patents classification.

The first test was run on a sample when only patents that belong to classes “A 61 K 39/xx” şi “G 06 F 3/xx” were selected from the entire database, where xx can take any value. Only two subclasses of level 4 and all their subclasses (level 5) have been chosen. “A 61 K 39” corresponds to the class “Medicinal preparations containing antigens or antibodies” and “G 06 F 3” corresponds to the class “Input / Output arrangements for transferring data to be processed into a form capable of being handled by the computer”. In order to test how good the performances of the clustering are,

two classes from different technical fields with clearly defined patent content ware chosen. Patents in the same cluster have many words in common, and patents from different clusters have very few words in common.

There are 1887 patents into the selected sample. After stop words were removed in a preprocessing step, an 8292 words dictionary resulted.

After clustering using the two algorithms the F-measure values from Table 1 were obtained.

Table 1. F-measure values for 10 runs of k-means algorithms with and without taking in account applicant, with k=2

k-means without applicant	k-means with applicant
0.9989403081913734	1
0.9984106045353137	0.9989399223930397
0.9994701063563782	1
0.9994701063563782	1
0.9989403081913734	1
0.9989403081913734	0.9994700099070553
0.9989403081913734	1
1	1
0.9989403081913734	1
0.9994701063563782	0.9989399223930397

Both algorithms had a high success rate as shown in Table 1. Even more, in clusters in which the applicant was taken into account, average result for F-measure is closer to 1, meaning that the distribution of patents in clusters is obtained almost always the same as in reference classes.

However, as was previously stated, the aim of this clustering is not to obtain clusters identical to the classification that already exists in patents, but to get groups of similar patents and content, which often do not respect the existing classification. F-measure is used in this research to compare the quality of the results of the two clustering algorithms with each other, using the same reference: the existing classification of patents.

The second test was run on a sample where four clusters have been involved. This new sample was selected from the entire database of patents for the patents belonging to classes “B 06 B x/xx”, “D 07 B x/xx”, “C 07 B x/xx” and “G 02 C x/xx”, where x/xx can take any value. “B 06 B” represents class “Generating or transmitting mechanical vibrations in general”, “D 07 B” - “Ropes or cables in general”, “C 07 B” - “General methods of organic chemistry; apparatus therefore” and “G 02 C” - “Spectacles; sunglasses or goggles insofar as they have the same features as spectacles; contact lenses”

The total number of patents in this second sample is 1036. After preprocessing step, a dictionary of 7143 words resulted.

Table 2. F-measure values for 10 runs of k-means algorithms with and without taking in account applicant, with k=4

k-means without applicant	k-means with applicant
0.5461674365294689	0.9124094468917905
0.7861729251761738	0.7673523367167561
0.7191038742088458	0.8297373641870704
0.7001758444743358	0.6540451822291918
0.7537968805130522	0.7868430198460827
0.9429385115120684	0.7212660694873748
0.7497156183217322	0.7817012018612501
0.8479571103115027	0.6057751522785366
0.7560356185839272	0.7230972262367343
0.8099655276505202	0.7465955631048976

F-measure values obtained after clustering are displayed in Table 2.

By comparing the two columns of F-measure values from Table 2, can be noticed that the highest value for the classical clustering algorithm is 0.942, and the highest value for the clustering that takes into account the applicant is 0.912. Also, should be noticed that if the averages of the F-measure values from each column are compared this values are close: 0.7829 and 0.7753 respectively. This means that the two clustering algorithms generate results almost as good.

Further, the clustering results are analyzed in order to observe the distribution of applicants and their patents in each cluster.

In all 1036 patents from the sample considered there are 603 applicants identified, some of them being found with 16 or even 20 or 21 patents. The distribution of these applicants in the clusters is considered to be relevant for the clustering results. This distribution is presented in Table 3.

Table 3. Number of applicants with patents distributed on each cluster

	Reference clusters	Clusters from k-means without applicant	Clusters from k-means with applicant
Number of applicants with patents in one cluster	593	582	588
Number of applicants with patents in 2 clusters	9	20	15
Number of applicants with patents in 3 clusters	1	1	0

Using the second clustering (clustering with applicants), the number of applicants that have patents in two clusters is less than the number of applicants from the first clustering (clustering without the applicant), as shown in Table 3: 15 versus 20. Also in the first clustering, an applicant with patents in three clusters can be seen and in the second clustering the same applicant now has patents in two clusters. This means that patents owned by the same applicant are more grouped in the cluster in which the applicant has the majority of patents.

By using the clustering algorithms, there should be noticed that the number of applicants with patents in two clusters is higher than in the reference classes, because by clustering, patents were redistributed according to their similarity: 20 and 15 versus 9. That means clustering algorithms have found different groups of patents then the agents choice for main patent classes.

From the above two observations we can conclude that by using clustering a benefit can be obtained from the advantage of grouping patents by content, which is relevant to a search result, and more, that the result of clustering where the applicants were considered had led to grouping the applicants and their relevant patents into the same cluster.

8 Conclusions

Using the k-means algorithm proposed in this article, the advantages of clustering that generates documents with close content and the advantages of grouping relevant patent of an applicant into a cluster are combined, thus obtaining clusters that can return relevant results to the users who perform searches.

By controlling the influence of applicants, clusters that contain only the relevant patent applicants, and not all their patents, can be obtained. The higher the influence of applicants in clustering is, the more patents of the same applicant appear in the same cluster, decreasing the content similarity of patent obtained exclusively by classical clustering.

From the above described model, there should be noticed that the larger the number of patents with the same applicant is, the closer these patents are. But the applicant influence should not be overly high (the number of patents with the same applicant may be hundreds) because only clusters of patents with the same applicant will be retrieved. Therefore the influence of metadata had to be limited and calculated according to the values and influence of the other words from the set of patents. (equation (7))

The model is also applicable to the other metadata fields, where the title or the inventor can be mentioned. For example, a greater importance to the words from the title could be given, so patents containing similar words in the title will be closer, even if they have a rather different patent content.

The present research proposes to use metadata to help to determine weights to improve k-means clustering. The application area is for patent databases, but the idea could be generalized and also investigated in other areas.

References

1. Pressman, D.: Patent It Yourself, 11th edn., Nolo (2005)
2. Delphion – Text Clustering, http://www.delphion.com/products/research/products-cluster
3. Manish, S.: Text Clustering on Patents. White Paper. Gridlogics Tech. Pvt. Ltd. (2009)
4. PatentCluster, http://www.patentcluster.com/

5. Vlase, M., Munteanu, D.: Patent relevancy on patent databases. In: Networking in Education and Research, Proceedings of the 8th RoEduNet International Conference (2009)
6. WIPO – Glossary, http://www.wipo.int/pctdb/en/glossary.jsp#p
7. WIPO Guide to Using PATENT INFORMATION. WIPO Publication No. L434/3(E) (2010) ISBN 978-92-805-2012-5
8. Giereth, M., Brügmann, S., Stäbler, A., Rotard, M., Ertl, T.: Application of semantic technologies for representing patent metadata. In: Proceedings of the First International Workshop on Applications of Semantic Technologies (2006)
9. Chau, M., Huang, Z., Qin, J., Zhou, Y., Chen, H.: Building a scientific knowledge web portal: The NanoPort experience. Decision Support Systems 42(2), 1216–1238 (2006)
10. MacQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297 (1967)
11. Hartigan, J.: Clustering Algorithms. John Wiley & Sons, New York (1975)
12. Hartigan, J., Wong, M.: Algorithm AS136: A k-means clustering algorithm. Applied Statistics 28, 100–108 (1979)
13. Modha, D., Spangler, S.W.: Feature weighting in k-means clustering. Machine Learning 52(3) (2003)
14. Berkhin, P.: A Survey of Clustering Data Mining Techniques. In: Grouping Multidimensional Data, pp. 25–71 (2006)
15. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM 18(11), 613–620 (1975)
16. Manning, C.D., Raghavan, P., Schütze, H.: An Introduction to Information Retrieval. Cambridge University Press, Cambridge (2009)
17. Salton, G., Buckley, C.: Term-weighting approaches in automatic retrieval. Information Processing & Management 24(5), 513–523 (1988)
18. WIPO - International Patent Classification, http://www.wipo.int/classifications/ipc/en/

Content Independent Metadata Production as a Machine Learning Problem

Sahar Changuel and Nicolas Labroche

Université Pierre et Marie Curie - Paris 6,
CNRS, UMR7606, LIP6, France
{Sahar.Changuel,Nicolas.Labroche}@lip6.fr

Abstract. Metadata provide a high-level description of digital library resources and represent the key to enable the discovery and selection of suitable resources. However the growth in size and diversity of digital collections makes manual metadata extraction an expensive task. This paper proposes a new content independent method to automatically generate metadata in order to characterize resources in a given learning objects repository. The key idea is to rely on few existing metadata to learn predictive models of metadata values. The proposed method is content independent and handles resources in different formats: text, image, video, Java applet, etc.

Two classical machine learning approaches are studied in this paper: in the first approach a supervised machine learning technique classify each value of a metadata field to be predicted according to the other a-priori filled metadata fields. The second approach used the FP-Growth algorithm to discover relationships between the different metadata fields as association rules. Experiments on two well-known educational data repositories show that both approaches can enhance metadata extraction and can even fill subjective metadata fields that are difficult to extract from the content of a resource, such as the difficulty of a resource.

Keywords: Metadata extraction, machine learning, association rules.

1 Introduction

The number of digital library repositories is growing rapidly underway worldwide, and as a consequence, the whole field of learning objects is rapidly maturing as a research area in its own right. And with the growth in the number, size and diversity of digital collections, the use of metadata has been fairly widely accepted as the solution for making electronic resources accessible to users [1]. In this sense, metadata functions in a manner similar to a card or record in a library catalogue, providing controlled and structured descriptions of resources through searchable "access points" such as title, author, date, description and subject.

Metadata of materials within a repository are the key that unlocks their potential for reuse. At its best (i.e. "accurate, consistent, sufficient, and thus reliable"), metadata is a powerful tool that enables the user to discover and retrieve

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 306–320, 2012.

relevant materials quickly and easily and to assess whether they may be suitable for reuse [2]. However, most sophisticated and semantically rich applications of metadata to documents are still handcrafted. But with large deployments where a considerable number of learning objects are to be managed, manual metadata creation is too time-consuming and costly. This can represent a barrier in an area where one of the benefits is supposedly saving time, effort and cost. Therefore, automatic procedures for the extraction of metadata from electronic resources are of great interest.

Addressing this challenge is a growing domain of research on automatic metadata generation which can be categorized into two subcategories [3]: metadata harvesting and metadata extraction.

Metadata harvesting occurs when metadata is automatically collected from previously defined metadata fields. The harvesting process relies on the metadata produced by humans or semi-automatic processes supported by software [4]. For example, web editing software generally automatically produces metadata at the time a resource is created or updated for 'format', 'date of creation', and 'revision date', without human intervention. Furthermore, in HTML pages, metadata can be specified manually in the corresponding <meta> tags. These tags can indicate the page title, the author name, the description, the keywords or any other metadata. A current limitation of the harvesting method is that the metadata elements are not always populated by resource creators or softwares.

On the other hand, metadata extraction, occurs when an algorithm automatically extracts metadata from the content of a resource. Among many proposed methods, regular expression [5], rule-based parser [6], and machine learning [7,8] are the most popular of these. In general, machine learning methods are robust and adaptative and, theoretically, can be used on any documents set. However, document parsing and generating the labeled training data are very time-consuming and costly.

This paper focus on the problem of automatic metadata generation to characterize learning resources in a given repository without accessing to the content of the resources. The problem of metadata production is considered as a machine learning task which predicts the values of the different metadata fields based on previously filled metadata fields in the repository. The proposed method is able to handle resources in different formats: text, image, video, etc., which, to the best of our knowledge, has not been addressed in the literature.

More precisely we propose two methods to study the relationships between the metadata fields. The first method is based on a supervised machine learning approach which aims to classify each metadata field using the other metadata fields as instances to characterize the resource. The second method is particularly adapted for resources with few a-priori filled metadata fields, and hence apply the FP-Growth [9] algorithm to discover relationships between the different metadata fields in the form of association rules.

The paper is organized as follows: in the next section, we describe the learning object repositories used in this paper: Ariadne an iLumina. Section 3 describes the supervised machine learning approach used to predict the values of the different

metadata fields. In Section 4, we present the generation of the rules associating the different metadata fields. Lastly, in Section 5, we draw the conclusions and describe future works.

2 Data Acquisition

The metadata files used in our experiments are collected from two on line learning object repositories (LOR): Ariadne and ilumina.

– Ariadne [10] is a LOR which represents a tool for cataloging and archiving educational resources based on metadata in LOM[1] format. It holds various collections of documents, newspaper articles, databases of readers and authors, places and events. For some collections, namely external publications, Ariadne only keeps metadata and the access to the content of the resources needs authentication.
To get the metadata files, we use the web application Ariadne Harvester[2] using its corresponding OAI-PMH target[3]. 4773 metadata files are obtained and stored locally in XML format. The files are then parsed in order to extract the different metadata fields (using XPath queries).
Figure 1 illustrates statistical properties of some metadata fields distributions in the obtained data set.

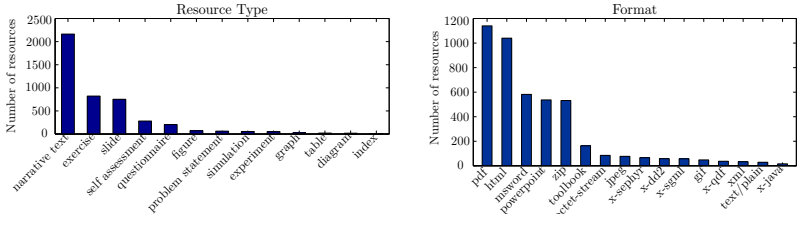


Fig. 1. Metadata distribution of the metadata fields: *Resource type* and *Format* in Aridane data set

– ILumina[4] is a digital library of sharable undergraduate teaching materials for chemistry, biology, physics, mathematics, and computer science. Resources in iLumina are catalogued in the MARC[5] and NSDL[6] metadata formats, which capture both technical and education-specific information about each

[1] Learning Object Metadata: http://ltsc.ieee.org/wg12/index.html
[2] http://ariadne.cs.kuleuven.be/lomi/index.php/Harvesting_LOM#Metadata_Validation
[3] http://ariadne.cs.kuleuven.be/ariadne-ws/services/oai
[4] http://www.ilumina-dlib.org/
[5] Machine Readable Cataloging: http://www.loc.gov/standards/
[6] http://nsdl.org/collection/metadata-guide.php

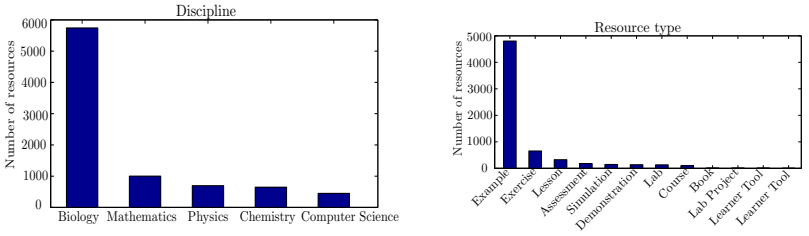


Fig. 2. Metadata distribution of the metadata fields: *Discipline* and *Interactivity level* in ilumina data set

resource. The metadata files have been downloaded from the iLumina web site. The files are in HTML format and share the same layout structure which help us to extract the values of the different metadata fields. 8563 metadata files are obtained. Figure 2 illustrates the distribution of the values of some metadata fields in the obtained data set. It can be observed that the metadata distribution is highly unbalanced: the fields "Discipline" and "Resource type", for example, are biased by the values: *Biology* and *Example*.

Both repositories are interesting because they contain different types of resources such as images, videos, Java applets, etc., and not only textual resources. The automatic extraction of metadata from the content of such resources is difficult and time consuming [11], we want to verify if generating relationships between metatada can help to better characterize these resources.

3 Classifiers Predictions for Metadata Production

In this section, we propose to formalize the metadata production problem as a traditional supervised machine learning problem. More precisely, each prediction task for a value of a metadata field is seen as a classification task in which the values of the other metadata fields are used as attributes to describe the learning instances.

With metadata fields that have more than two possible values, we perform a multi-class classification. We adopt the "one against all" strategy, in which a C-class problem is transformed into C two-class problems. In this case, when considering the ith two-class problem, class i is learned separately from all the remaining classes.

3.1 Experimental Protocols

Datasets. Tables 2 and 3 illustrate the different metadata handled in our work as well as the number of possible values for each metadata field in Ariadne and iLumina successively. From these tables, it can be observed that the majority of the metadata fields have more than two possible values (except the *Interactivity type* in Ariadne and the *Interactivity level* and the *End User* in iLumina).

Table 1. Number of possible values for each metadata field in Ariadne and iLumina

Table 2. Ariadne

Metadata fields	Nbr. values
Difficulty	5
Format	52
Resource type	25
Interactivity type	2
Interactivity level	5
Author	810
Discipline	15

Table 3. iLumina

Metadata fields	Nbr. values
Difficulty	5
Format	19
Resource type	16
Interactivity level	2
Author	49
Discipline	5
End user	2
Mediatype	13

Algorithms. Experiments are conducted to predict all the metadata fields presented in tables 2 and 3 except the field *Format*. Indeed, the format can be easily obtained from the resource and do not require a machine learning algorithm. Experiments are conducted with 10-folds cross validation using different machine learning algorithms [12]: Naive Bayes (NB), C4.5, Random Forest with 10 trees and Support Vector Machines (SVM) with a polynomial kernel.

The NB model contains each class probability and conditional probability of each attribute value given a class. Classification uses the model to find a class with maximum probability given an instance [13].

C4.5 [14] produces decision tree by top-down induction derived from the divide-and-conquer algorithm. Each node in the tree is the best attribute selected based on information gain criterion.

Random forest (RF) is a state-of-the-art ensemble decision tree learner developed by Breiman [15]. Decision trees choose their splitting attributes from a random subset of k attributes at each internal node. The best split is taken among these randomly chosen attributes and the trees are built without pruning, as opposed to C4.5.

SVM [16] is a learning algorithm that constructs a hyper plane with maximal margin between classes. It finds some support vectors, which are the training data that constrain the margin width. We particularly use the Sequential minimal optimization algorithm (SMO) algorithm which resolves quadratic programming optimization problem that arises when determining the maximum margin hyperplane of the support vector machines classifier [17]. Because SMO is a binary classification algorithm, for multiclass classification purposes required in this work it is adapted such that it performs $n \times (n-1)/2$ binary classifications.

3.2 Results and Discussions

To assess the classification performance of each metadata field, we measure the average of the F-measure score obtained from the classification of all the classes of

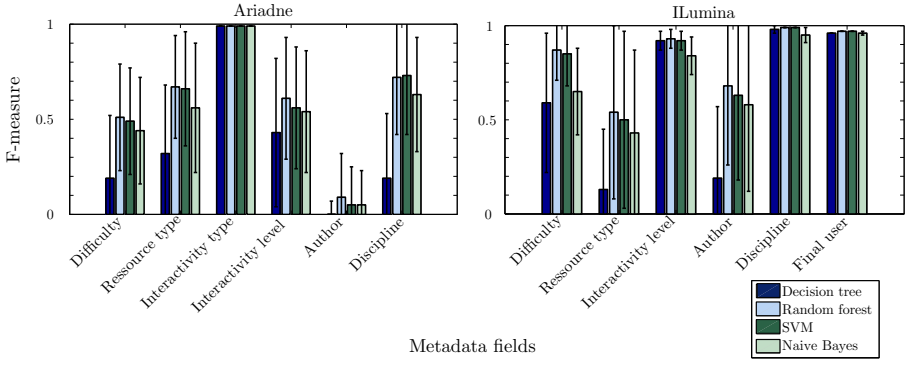


Fig. 3. Results of metadata classification in both repositories Ariadne and iLumina

the given metadata field. The F-measure metric which is the weighted harmonic mean of precision and recall:

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{1}$$

Where the precision measures the number of correctly identified items as a percentage of the number of items identified, and the recall measures the number of correctly identified items as a percentage of the total number of correct items.

Concerning the metadata fields having two possible values, we adopt a binary classification using the same machine learning algorithms. The obtained results are illustrated in Fig 3.

Fig 3 shows that better results are obtained with both the Random Forest and the SVM algorithm, with RF models performing slightly better. Random forest improves the average results of each metadata field compared to the decision tree algorithm. In fact, with our unbalanced data distributions, the risk of overfitting is important. The Random Forest algorithm avoid overfitting thanks to the double "randomization" compared to a single decision tree (randomization in the choice of the data samples used for growing the tree and in the choice of the attributes that split the nodes of the tree). On the other hand, the SVM algorithm, based on the structural risk minimization, outperforms the Naive Bayes algorithm.

From Fig 3, it can also be underlined that with metadata fields having two possible values, the binary classification gives good results (*Interactivity type* in Ariadne, *Interactiviy level* and *End User* in iLumina). The algorithms are indeed able to properly classify these fields based on th values of other metadata fields. This result can be explained by the fact that with these metadata fields, the classes imbalance is less important than with the other fields characterized by several classes. In addition, binary classification is known to be more efficient than multi-classification. The importance of such results lies in the fact that

the values of these metadata fields are difficult to extract from the content of a resource.

Fig 3 also shows that it is difficult to classify the author field using other metadata fields as attributes. We have shown in previous studies [18] that best results can be obtained to induce the author's name using the content of the resource.

Besides, it appears that the classification of the fields *Difficulty*, *Interactivity level* and *Discipline* gives better results with the iLumina data set than with the Ariadne data set. Indeed, with the Ariadne data set, 73% of the resource have the field *Difficulty* filled by the value: *medium*. This imbalance makes it difficult to classify the other classes of the *Difficulty* field as we can see through the confusion matrix given in Table 4. Nevertheless, we can notice through the confusion matrices of the field *Difficulty* in Ariadne and iLumina illustrated in tables 4 and 5, that the diagonal values correspond to the maximum value of each column, except for the class *very difficult* in Ariadne whose major examples are predicted as *difficult* and which cannot be considered as a problem from an application point of view. Predicting the difficulty of a resource is a challenging task, and through the given results it appears that our approach is able to automatically induce the value of this subjective metadata field.

Concerning the *discipline* field, there are 15 possible discipline values with Ariadne and 5 with iLumina. This makes the classification of the iLumina instances easier than that of Ariadne. Nevertheless, with the Random Forest algorithm, we obtain an F-measure of 72% with the Ariadne examples. And despite the high value of standard deviation (0.31), the algorithm gives good classification results for the majority of the examples of each class. Indeed, 11 of the 15 categories have an F-measure above 74%. The high value of the standard deviation

Table 4. Confusion matrix of the difficulty field with the Ariadne data set. The correct classification rate is 82.45%

Predicted as ⇒	very easy	easy	medium	difficult	very difficult
very easy	**35**	8	29	2	0
easy	14	**121**	101	6	0
medium	10	31	**1514**	50	0
difficult	3	7	111	**130**	2
very difficult	1	0	3	5	**0**

Table 5. Confusion matrix of the difficulty field with the iLumina data set. The correct classification rate is 93.25%

Predicted as ⇒	very easy	easy	medium	difficult	very difficult
very easy	**2778**	26	0	0	0
easy	69	**3550**	279	9	0
medium	0	107	**1195**	14	0
difficult	0	17	27	**45**	0
very difficult	0	0	0	0	**6**

is mainly due to the classification of the examples of two classes: *Arts* and *Literature* (F-measure = 0) which are not very representative in the data base (7 and 11 instances respectively).

Consequently, through the previous analysis, we find that good results are obtained for predicting many metadata fields. The proposed method is interesting insofar as there is no need to access to the content of the resource to extract the desired information. Moreover it can fill subjective metadata fields that are difficult to extract from the content of a resource, such as the *End user*, the *difficulty* and the *educational category* (exercice, example, lesson, etc.) of a resource.

4 Association Rules Generation for Metadata Production

The method proposed in the previous section gives interesting results to predict the metadata values. However, a new resource may have an insufficient number of a-priori filled metadata. In this case, the classifier may not predict the value of a given metadata field correctly.

In order to overcome this problem, we propose to analyze the relationships between the different metadata fields by generating association rules.

4.1 Experimental Protocol

We are interested in rules of the form $A \Rightarrow B$ where A denotes conjunctions of presence of different metadata fields values and B corresponds to the value of one metadata field.

To identify these rules, discrete attributes are transformed to binary attributes which represent the *items*. Each resource represents a transaction and is characterized by a list of items. To generate association rules between metadata, we apply the FP-Growth [9] algorithm since it is known to be more effective than Apriori [9, 19], and because of its FP-Tree structure which allows a considerable reduction of the processing time. Indeed, Apriori needs $n + 1$ scans, where n is the length of the longest pattern, whereas, FP-growth use only two scans of the database to find the frequent itemsets. First, an FP-tree is created which is a condensed representation of the dataset. Then a mining algorithm generate all possible frequent patterns from the FP-tree by recursively traversing the conditional trees generated from the FP-tree.

In our experiment the support threshold is fixed to 0.01 and the confidence threshold to 0.8. The low value of the support comes from the fact that the numbers of resources with some metadata values are very low.

4.2 Initial Results

90 rules are obtained with Ariadne and 206 with iLumina. Since the iLumina data set is highly unbalanced, the majority of the rules are generated with the most frequent items, which are “Mediatype = image”, “Discipline = biology”, “Interactivity level = low” and “Resource type = example”. Examples of rules obtained with iLumina are the followings:

- Mediatype = image ⇒ Interactivity level = low
- Discipline = biology ⇒ Interactivity level = low
- Mediatype = image, Discipline = biology, Difficulty = easy ⇒ Resource type = example

Indeed, 70% of the ressource in the iLumina data set are images, and it appears that there is a high co-occurence between images, the biology discipline and the low level of interactivity. This means that, users in iLumina generally insert ressources in image format that are used as examples in the domain of biology, and we can easily guess that images have a low level of interectivity. This result is indeed compatible with common representations.

As a consequence, the obtained rules can characterize image format resources which are in the biology domain. In order to characterize resources in other formats, we remove the resources in image format from the data set to obtain 2383 transactions.

4.3 Rules Pruning

After filtering the image type iLumina transactions, we obtain 90 rules with Ariadne and 754 rules with iLumina using the FP-Growth algorithm. However, we notice that many rules are redundant. An example of redundant rules are r and R as follows:

- R: Format = powerpoint ⇒ Resource type = slide (supp = 0.09, conf = 0.87)
- r: Discipline = computer science, Format = powerpoint ⇒ Resource type = slide (supp = 0.05, conf = 0.91)

Both rules have the same conclusion and a common item in the condition. However, intuitively, R seems to have a more predictive power than r, the resources in powerpoint format are intuitively slides. If we have R, then r is insignificant because it gives little extra information. Its slightly higher confidence is more likely due to chance than to true correlation.

Thus, such spurious rules should be removed. This is similar to pruning of overfitting rules in classification [14]. Rules that are very specific (with many conditions) tend to overfit the data and have little predictive power [20]. Therefore, we decide to use a pruning technique in order to keep only the most interesting rules.

To prune redundant rules, we use the method proposed in [20] which measures the significance of a rule using the χ^2 test. Computing the χ^2 test for the pair of variables (A,B) requires constructing two contingency tables. The observed contingency table for (A,B) has four cells, corresponding to the four possible boolean combinations of A, B. The value in each cell is the number of observations (samples) that match the boolean combination for that cell. These values may be expressed in terms of the total number of samples n and of the observed relative frequencies (probabilities) corresponding to the four boolean combinations as shown in Table 6.

Chi-square analysis dictates that the observed contingency table should be compared with that which would be obtained asymptotically as $n \to \infty$ if the variables A and B were statistically independent. The latter table is shown in Table 7.

Table 6. Observed contingency table for (A, B)

	B	$\overline{B}$
A	$nP(A \cap B)$	$nP(A \cap \overline{B})$
$\overline{A}$	$nP(\overline{A} \cap B)$	$nP(\overline{A} \cap \overline{B})$

Table 7. Expected contingency table for (A, B)

	B	$\overline{B}$
A	$nP(A)P(B)$	$nP(A)(1 - P(B)$
$\overline{A}$	$n(1 - P(A))P(B)$	$n(1 - P(A))(1 - P(B)$

The χ^2 test is defined in terms of the entries of the observed contingency table (Table 6) and the expected contingency table (Table 7) as follows.

$$\chi^2 = \sum_{0 \leq i,j \leq 1} \frac{(f_{o_{i,j}} - f_{i,j})^2}{f_{i,j}} \tag{2}$$

Where f_o is an observed frequency, and f is an expected frequency. The closer the observed frequencies are to the expected frequencies, the greater is the weight of evidence in favor of independence.

Concerning the rules r and R, r can be pruned with respect to R because within the subset of data cases covered by R, r is not significant (a rule covers a set of data cases, if the data cases satisfy the conditions of the rule). To prune non significant rules we use the approach proposed in [20]. Authors propose to perform a χ^2 test on r with respect to each ancestor rule R (which has the same consequent as r but fewer conditions). If the test shows a positive correlation, it is kept. Otherwise, r is pruned.

In our case, pruning is done in a post-processing phase once the rules are generated by the FP-Growth algorithm. Each generated rule has a set of evaluation measures such as support and confidence. To obtain f_o, f and $\chi^2(r, R)$, we should have the observed and theoretical contingency tables of both rules r and R. Getting these contingency tables from the data base can be expensive in terms of computation since it requires parsing all the transactions. In order to avoid this computation time, we propose to express the χ^2 correlation test using the obtained evaluation measures of the generated rules.

4.4 Improving FP-Growth with Efficient χ^2 Based Rules Pruning

It was demonstrated in [21] that it is possible to express the χ^2 test, the observed and the expected frequencies f and f_o of an association rule R based on the support, the confidence and the lift of the rule. The obtained equations are the following:

- $\chi^2(R) = n(lift - 1)^2 \dfrac{supp.conf}{(conf - supp)(lift - conf)}$

- $f = n.supp$
- $f_o = n.\frac{supp}{lift}$, n being the total number of samples.

In this paper, we propose to express $\chi^2(r, R)$ using the evaluation measures of both rules r and R in order to avoid parsing all the transactions once the association rules are obtained. The proposed method is detailed hereafter.

Let the following rules R and r, where A, B and C are items and R is ancestor of r :

- R: A $\longrightarrow$ C
- r: A, B $\longrightarrow$ C

Let f_o be the observed frequency of r and f its theoretical frequency compared to R, we can demonstrate that:

$$\begin{aligned} f_o &= supp(r).n \\ f &= \frac{n.supp(r).conf(R)}{conf(r)} \end{aligned} \tag{3}$$

Proof:

f_o is the number of samples covered by r, then $f_o = |A \cap B \cap C|$.

Moreover, the theoretical frequency f corresponds to the number of samples covered by r among the samples that are already covered by R. Then $f = \frac{|A \cap B|.|A \cap C|}{|A|}$.

We also have:

$$\begin{aligned} &- \quad supp(r) = \frac{|A \cap B \cap C|}{n} \qquad - \quad supp(R) = \frac{|A \cap C|}{n} \\ &- \quad conf(r) = \frac{|A \cap B \cap C|}{|A \cap B|} \qquad - \quad conf(R) = \frac{|A \cap C|}{|A|} \end{aligned} \tag{4}$$

Which implies:

- $\mathbf{f_o = supp(r).n}$
- $f = \frac{|A \cap B|.|A \cap C|}{|A|} = \frac{\mathbf{n.supp(r).conf(R)}}{\mathbf{conf(r)}}$

Using the obtained equations, we can express $\chi^2(r, R)$ in terms of support and confidence as follow:

$$\chi^2(r, R) = \frac{(f_o - f)^2}{f} = n.supp(r)\frac{[conf(R) - supp(r).conf(R)]^2}{conf(r).conf(R)} \tag{5}$$

In a post-processing stage, after generating the rules, it is easy to obtain the different components of Equation 5. Using the obtained formula, we apply the pruning algorithm presented in [20] in order to remove the non significant rules.

4.5 Results and Discussion

After applying the pruning algorithm on the obtained rules, we obtain 25 rules from the Ariadne data set (out of 90 rules initially generated) and 55 rules from the iLumina data set (out of 754 initial rules). The number of rules has decreased to keep only the most important ones which enables us to better interpret them and to use them in a metadata generation system.

- **Ariadne rules**

Among the obtained rules, some are given in Table 8. Note that for anonymity reasons, the names of the authors have been replaced by letters.

When analyzing the generated rules, we can notice that many rules are interesting for the induction of the metadata values. Indeed, it appears that a resource with a very low level of interactivity is *expositive* (rule n^o1). Moreover, with an author who generally creates difficult resources or resources with a high level of interactivity, these properties can be generalized to all the resources the same author creates (rules n^o3 and n^o5). In addition to that, since authors usually creates resources in the same disciplines, the discipline of a resource can then be induced from the name of the author (rule n^o4). This can avoid parsing the content of the resources and applying expensive classification approaches to extract the discipline of a resource. On the other hand, it can be observed that a resource with an *active* interactivity type can be characterized by a high level of interactivity (rule n^o6).

Accordingly, it is easy to notice that all these characterizations correspond to a commonly accepted and intuitive representation.

Lastly, among the obtained rules, some rules can be considered as generalizable since they can be independent from the data set. Examples of such rules are the followings:

- Resource type = narrative text $\longrightarrow$ Interactivity type = expositive.
- Resource type = slide $\longrightarrow$ Interactivity type = expositive.
- Interactivity type = active $\longrightarrow$ Interactivity level = high.

Table 8. Examples of rules generated from Ariadne

N	**Condition**	**Conclusion**	**Support**	**Confidence**
1	Interactivity level = very low	Interactivity type = expositive	0.11	0.97
2	Interactivity level = low	Interactivity type = expositive	0.30	0.88
3	Author = X	Interactivity level = high	0.10	0.88
4	Author = X	Discipline = mecanique	0.11	1.0
5	Author = X	Difficulty = difficult	0.1	0.88
6	Interactivity type = active	Interactivity level = high	0.30	0.86
7	Resource type = exercise	Interactivity level = high	0.17	0.94
8	Resource type = exercise	Interactivity type = active	0.18	1.0
9	Interactivity type = expositive, Difficulty = easy	Interactivity level = low	0.23	0.92
10	Discipline = computer science, Interactivity level = high	Resource type = exercise	0.10	0.84

- **ILumina rules**

Some examples of the rules generated from the iLumina repository are illustrated in Table 9. It can be noticed that, again, there is a causal relationship between the author's name and the discipline of the resource. It can also be observed that *Lesson* resources have a low level of interactivity which can be seen as an obvious result. On the other hand, it appears that the generated rules are able to characterize resources in different formats, such as video (*rule* n^o1), Java Applet (rule n^o5) and Maple application (rule n^o7). It is interesting to characterize these resources since it is difficult to extract adequate information from the content of such kind of resources.

Lastly, we again can distinguish rules which may be generalized to other repositories, such as the following rules:

- Mediatype = video $\longrightarrow$ Interactivity level = low.
- Format = application/maple $\longrightarrow$ Discipline = mathematics.
- Format = text/html $\longrightarrow$ Mediatype = web Page.

Accordingly, we notice that the pruning approach allows us to keep only the most interesting rules on the one hand, and to be able to interpret them on the other hand. Our goal in generating the association rules is to find causal relationships between the different metadata fields and to check whether the obtained rules are semantically correct. The obtained results confirm our assumption since it is possible to find interesting relationships between metadata fields when considering all the metadata of resources in a repository. The obtained relationships can be used to automatically annotate or to help indexers to annotate the metadata of resources in a repository.

Table 9. Examples of rules generated from iLumina

N	**Condition**	**Conclusion**	**Support**	**Confidence**
1	Mediatype = video	Interactivity level = low	0.23	0.97
2	Resource type = lesson	Interactivity level = low	0.2	0.96
4	Auteur = A	Discipline = mathematics	0.18	1.0
5	Mediatype = java Applet	Discipline = physics	0.27	0.94
6	Auteur = B	Discipline = physics	0.28	1.0
7	Format = application/maple	Discipline = mathematics	0.18	1
8	Auteur = A	Resource type = lesson	0.18	1.0

5 Conclusion

The use of metadata has been fairly widely accepted as the solution for making electronic resources accessible to users. And with the rise of digital library repositories, automatic metadata annotation is becoming a real need.

This paper proposes a new content independent method to automatically generate metadata in order to characterize resources in a given learning objects

repository. Thus it considers the problem of metadata production as a machine learning task which aims at learning predicting relationships between the different metadata fields based on some previously filled metadata fields in the repository. Two machine learning methods are experimented in this paper: the first method classifies each value of a metadata field to be predicted according to the other a-priori filled metadata fields. Experiments are conducted comparing different machine learning algorithms: Naive Bayes, C4.5, Random Forest and SVM, obtaining better results with the two latter algorithms. Interesting results are obtained for predicting subjective metadata fields that are difficult to extract from the content of the resource like the difficulty of a resource or its interactivity level.

The second method applies the FP-Growth algorithm to discover relationships between the different metadata fields in the form of association rules. A χ^2 based method is proposed to prune the rules and to keep only the most interesting ones. The obtained rules show interesting causal relations between different metadata fields. These relations are concordant with common representations and can indeed be used to help the indexers while filling the metadata fields in a repository.

The proposed methods are interesting insofar as there is no need to access to the content of the resource to extract information that characterize it. Our approach can be a solution to the tedious manual annotation of educational resources. Further research can be conducted to compare our approach in terms of performance and computation time with existing content-based methods.

References

1. Liddy, E., Chen, J., Finneran, C., Diekema, A., Harwell, S., Yilmazel, O.: Generating and Evaluating Automatic Metadata for Educational Resources. In: Rauber, A., Christodoulakis, S., Tjoa, A.M. (eds.) ECDL 2005. LNCS, vol. 3652, pp. 513–514. Springer, Heidelberg (2005)
2. Greenberg, J., Robertson, W.D.: Semantic web construction: an inquiry of authors' views on collaborative metadata generation. In: Proc. of the 2002 Int. Conf. on Dublin Core and Metadata Applications: Metadata for e-communities: Supporting Diversity and Convergence, Dublin Core Metadata Initiative, pp. 45–52 (2002)
3. Greenberg, J., Spurgin, K., Crystal, A.: Functionalities for automatic metadata generation applications: a survey of metadata experts' opinions. International Journal of Metadata, Semantics and Ontologies 1, 3–20 (2006)
4. Greenberg, J.: Metadata extraction and harvesting: A comparison of two automatic metadata generation applications. Journal of Internet Cataloging 6, 59–82 (2004)
5. Tang, X., Zeng, Q., Cui, T., Wu, Z.: Regular expression-based reference metadata extraction from the web. In: IEEE 2nd Symposium on Web Society (SWS), pp. 346–350 (2010)
6. Shek, E.C., Yang, J.: Knowledge-based metadata extraction from postscript files. In: Proc. of the 5th ACM Conference on Digital Libraries, pp. 77–84. ACM Press (2000)
7. Changuel, S., Labroche, N., Bouchon-Meunier, B.: A General Learning Method for Automatic Title Extraction from HTML Pages. In: Perner, P. (ed.) MLDM 2009. LNCS, vol. 5632, pp. 704–718. Springer, Heidelberg (2009)

8. Han, H., Giles, C.L., Manavoglu, E., Zha, H., Zhang, Z., Fox, E.A.: Automatic document metadata extraction using support vector machines. In: Proc. of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL 2003, pp. 37–48. IEEE Computer Society (2003)
9. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. SIGMOD Record 29, 1–12 (2000)
10. Duval, E., Forte, E., Cardinaels, K., Verhoeven, B., Van Durm, R., Hendrikx, K., Forte, M.W., Ebel, N., Macowicz, M., Warkentyne, K., Haenni, F.: The Ariadne knowledge pool system. Communications of the ACM 44, 72–78 (2001)
11. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: Proc. of the 26th Annual International ACM SIGIR Conf. on Research and Development in Informaion Retrieval, SIGIR 2003, pp. 119–126. ACM, New York (2003)
12. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations 11 (2009)
13. John, G., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: Proc. of the Eleventh Conf. on Uncertainty in Artificial Intelligence, pp. 338–345. Morgan Kaufmann (1995)
14. Quinlan, R.: C4.5: Programs for Machine Learning, 1st edn. Morgan Kaufmann (1993)
15. Breiman, L.: Bagging predictors. Machine Learning 24, 123–140 (1996)
16. Vapnik, V.N.: The Nature of Statistical Learning Theory, 2nd edn. Springer (1999)
17. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization, pp. 185–208. MIT Press, Cambridge (1999)
18. Changuel, S., Labroche, N., Bouchon-Meunier, B.: Automatic Web Pages Author Extraction. In: Andreasen, T., Yager, R.R., Bulskov, H., Christiansen, H., Larsen, H.L. (eds.) FQAS 2009. LNCS, vol. 5822, pp. 300–311. Springer, Heidelberg (2009)
19. Wang, R., Xu, L., Marsland, S., Rayudu, R.: An efficient algorithm for mining frequent closed itemsets in dynamic transaction databases. International Journal of Intelligent Systems Technologies and Applications 4, 313–326 (2008)
20. Liu, B., Hsu, W., Ma, Y.: Pruning and summarizing the discovered associations. In: Proc. of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 125–134. ACM Press (1999)
21. Alvarez, S.A.: Chi-squared computation for association rules: preliminary results. Technical report, Computer Science Department, Boston College (2003)

Discovering K Web User Groups with Specific Aspect Interests

Jianfeng Si[1], Qing Li[1], Tieyun Qian[2], and Xiaotie Deng[1]

[1] Department of Computer Science,
City University of Hong Kong, Hong Kong, China
jianfsi2@student.cityu.edu.hk, {itqli,csdeng}@cityu.edu.hk
[2] State Key Laboratory of Software Engineering,
Wuhan University, Wuhan, China
qty@whu.edu.cn

Abstract. Online review analysis becomes a hot research topic recently. Most of the existing works focus on the problems of review summarization, aspect identification or opinion mining from an item's point of view such as the quality and popularity of products. Considering the fact that authors of these review texts may pay different attentions to different domain-based product aspects with respect to their own interests, in this paper, we aim to learn K user groups with specific aspect interests indicated by their review writings. Such K user groups' identification can facilitate better understanding of customers' interests which are crucial for application like product improvement on customer-oriented design or diverse marketing strategies. Instead of using a traditional text clustering approach, we treat the clusterId as a hidden variable and use a permutation-based structural topic model called *KMM*. Through this model, we infer K groups' distribution by discovering not only the frequency of reviewers' product aspects, but also the occurrence priority of respective aspects. Our experiment on several real-world review datasets demonstrates a competitive solution.

1 Introduction

With the accessibility of large volume of online review data(which is a kind of feedback of products), more and more researchers become interested in such valuable information. These reviews are usually written by experienced customers or domain experts, and can well reflect user experiences or preferences on products with commercial value. But it is hard for human users to do any summarization on such large datasets. There are a lot of existing works related to review analysis, yet mostly focus on the task of products recommendation to new users. In our research, we look into the review text writing styles in order to identify certain number of user groups, each of which shares similar interests in domain products. Such user groups can help manufacturers better understand customers for further customer-oriented product design, and also help guide the user-targeting marketing strategy. Take the notebook marketing as an example, basically there would be five different product categories for customers to choose, namely: *Basic, Portable, Performance, High-Performance, Multi-Media*, which reveal five distinct customer concerns. For instance, customers targeting at *High-Performance* would

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 321–335, 2012.

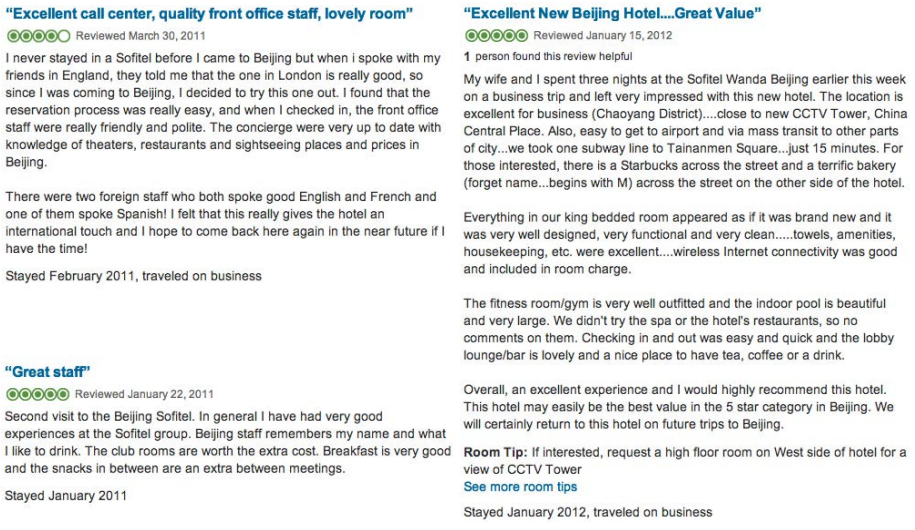


Fig. 1. Reviews of *Sofitel Wanda Beijing Hotel* from *www.tripadvisor.com*

typically comment using words like "CPU, speed, performance, RAM, etc.", while those targeting at *Portable* would comment using words like "size, weight, battery, etc.". The former group could be "software engineer" while the latter would more likely to be "business man". As revealed by this example, aspect interests reflected by text writings actually can identify user groups with separative taste of interests.

In this paper, we extend a permutation-based structural topic model for the task of unsupervised learning of user groups. Reviewers or customers are likely to express their usage experiences or personal preferences on domain products. In particular, different people may focus on different aspects to different extent, and also the order of aspects in which they express matters with their concern priorities. With the above consideration, we focus on the task of identifying certain number of user groups, with each of the groups taking a similar taste on aspect interests. Note that we term these as user groups instead of text clusterings to emphasize the actual human interests and concerns under text.

Compared to the regular articles, the online reviews are written in a much more free style, Fig.1 contains three hotel reviews, the left two reviews talk about staff service mostly while the right one pays more attention to the hotel location and room condition, in a word, they stress differently with own concerns. So it is hard to derive a unique global topic structure among review corpus. Instead, we identify K topic structures each of which shares similar aspect interests. The choice of K depends on what granularity the groups need to be, e.g., we may set $K = 5$ in the notebook example.

Chen et.al. [6] proposed a structural topic model with the assumption of "one centroid ordering constraint" for learning discourse-level document structure. We make use of their work, and further extend their latent topic model to solve the user grouping

problem by introducing a new hidden variable k, to indicate the group to which documents belong. We output K topic structures w.r.t. topic centroid orderings and topic frequencies.

The rest of the paper is organized as follows. In Section 2 we introduce the related works. In Section 3, we define our problem formulation and propose the model:*KMM*. We address the parameter inference problem in Section 4. Our experiment on several real world review datasets is presented in Section 5, and we conclude current and further work in Section 6.

2 Related Works

Online review analysis has attracted much attention recently, including opinion summarization [16,19,21], sentiment analysis [1,2,8], opinion spam detection [12,13,18]. Topic modeling has been explored for the task of aspect identification [17,21] where aspects are treated as topics. The interested posterior distributions are estimated using approximate inference techniques such as Gibbs sampling [3] or variational inference [14].

Compared to the bag-of-words and bag-of-topics assumption [5], newly developing topic models are more likely to integrate topic models with structural models [4,15] under the consideration of inner-connected relationship between topics. Many works break the bag-of-topics assumption and introduce extra-sentential constraints on topic assignment with structural considerations [11,20,21]. In particular, the relationships between topics assigned to adjacent textual units(sentences, paragraphs or sections instead of words) bias the topic distributions of adjacent textual units to be similar [20], forming a Markovian transition process. So the topic assignments are locally infected. For example, the Hidden Topic Markov Model(*HTMM*) [11] defined a generative process for the documents' topics, in which sentence i gets the same topic assignment as i-1 with a relatively high probability.

As the Markovian process only makes local decisions regarding the topic transitions, Chen et.al. [6] proposed a structural topic model which learned a global document structure under the assumption that there existed a global topic structure in a domain-based document collection. For example, when an article introduces a city, it mostly introduces its history first, then geography, politics, economy, etc., that is, the order as "history, geography, politics, economy, etc." defines the centroid topic ordering when we introduce one city. Each document follows the centroid ordering with some possible dispersion to get its own topic ordering(for example, to introduce the economy before politics sometimes). Their work makes use of the *Generalized Mallows Model(GMM)* [7] over permutations to express the centroid topic ordering, but it can only find one global structure which is not adapted to the discourse-level text corpus like reviews, thus we extend their model with an aim to solve the user grouping problem.

3 Model

We define in this section the problem of research and propose a new structural topic model called *K Mallows Model(KMM)*.

3.1 Problem Formulation

Given a product review corpus$\{d_1, \cdots, d_D\}$ in a certain domain, we consider the grouping problem on reviewers/users $\{u_i = Author(d_i)\}$ who generate these reviews. We assume there exist K such user groups $\{C_1, \cdots, C_K\}$ with separative common product interests and tastes, as indicated by their review writings. Note that although here we do clustering on reviews, we term these as user groups to emphasize the user interests on the product aspects. Take the hotel reviews as an example, we may discover one user group who care mostly on the room condition, then staff service, and lastly mention the food a little bit, while there would be another user group who care about the food mostly and firstly, then the hotel location and surrounding environment. Those two groups draw different topic concern frequency and concern priority.

Each document contains N_d sentences$\{s_{d,1}, \cdots, s_{d_{N_d}}\}$ and we regard a product aspect as a topic, and each sentence gets a topic assignment $z_{d,s} \in \{1, \cdots, T\}$. Our work is to find out K clusters on the review text corpus accompanied by a meaningful explanation. Our structural topic model jointly learns (1)K different aspect frequency distributions(e.g., the *Frequency* column in Table 2), (2)K centroid aspect orderings(e.g., the *Ordering* column in Table 2), (3)the user grouping based on (1,2)(e.g., the $\{C_0, C_1, C_2\}$ in Table 2), accompanied by the shared language model of each aspect described by the word distribution(e.g., Table 3).

3.2 Model Overview

There are two constraints considered in the *GMM* topic model of [6]: the first posits that each document exhibits coherent, nonrecurring topics, the second states that documents from the same domain tend to present a similar topic structure. We extend their model by breaking the second constraint, and assume that K similar topic structures exist in the review corpus, so as to help us identify K different user groups. We term the extended model as *K Mallows Model(KMM)* topic model.

Two points are considered when making the extension:

1. Lack of a global uniform topic structure
   Online review writings by various writers/reviewers are of totally free styles, since these writers may be freshmen, experienced customers, domain experts or sometimes spam makers. Beyond that, people always focus on different aspects w.r.t. their own interests. So, it is hard to derive a global uniform topic structure.
2. Power of the discriminative grouping
   Product manufacturers succeed from adopting a customer-oriented strategy. As the consuming market grows, all-in-one product design no longer applies. Customers should be discriminatively treated, so it is necessary to identify the discriminative grouping of various kinds of customers.

Similar to the *GMM* topic model, our model firstly finds out how frequently each topic is expressed in the document and how the topics are ordered. These ordered topic sequences then determine the selection of words for each sentence(we treat "sentence" as the basic text unit of topic assignment). The graphical model of the original *GMM* and our *KMM* are shown in Fig. 2[1] and Fig.3, respectively.

[1] Different from our model, K in *GMM* as shown by Fig.2 represents the number of topics.

As seen from Fig.3, there are K topic frequency distributions:$\{\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_K\}$ and also K centroid topic permutations:$\{\boldsymbol{\pi}_1^{'},\cdots,\boldsymbol{\pi}_K^{'}\}$. The combination of these two reflects K user groups with respective similar product interests. So, during the generative process of each document, we firstly select the groupId/clusterId: k_d, then draw the topic frequency and topic ordering w.r.t. the cluster it belongs to.

Here is the specification of all parameters and variables occurred in Fig.3:

1. User setting parameters
   - T - number of topics
   - K - number of clusters/groups
2. Document characteristics
   - D - number of documents in the corpus
   - N_d - number of sentences in document $d \in \{1,\cdots,D\}$
   - N_s - number of words in sentence $s \in \{1,\cdots,N_d\}$
3. Symmetric Dirichlet priors
   - α_0 - prior of the cluster size distribution
   - β_0 - prior of the topic frequency distribution
   - θ_0 - prior of the language model
4. Dirichlet distributions
   - $\boldsymbol{\theta}$ - parameters of the distribution over topic frequency: $\boldsymbol{\theta} \sim Dirichlet(\theta_0)$
   - $\boldsymbol{\beta}$ - parameters of the language model: $\boldsymbol{\beta} \sim Dirichlet(\beta_0)$
   - $\boldsymbol{\alpha}$ - parameters of the distribution of clusters' member size: $\boldsymbol{\alpha} \sim Dirichlet(\alpha_0)$
5. Standard *Mallows Model*
   - ρ - dispersion parameter of standard *Mallows Model*
   - $\boldsymbol{\pi_0}$ - natural ordering: $\boldsymbol{\pi_0} = \{1,\cdots,T\}$
   - $\boldsymbol{v}^{'}$ - inversion count vector of each cluster's centroid ordering w.r.t. π_0
   - $\boldsymbol{\pi}^{'}$ - centroid ordering of each cluster
   - $\boldsymbol{v}$ - inversion count vector of each document w.r.t. the $\boldsymbol{\pi}^{'}$ of the belonging cluster
   - $\boldsymbol{\pi}$ - topic ordering of each document
6. Other hidden variables
   - $\boldsymbol{k}$ - groupId/clusterId of each document
   - $\boldsymbol{t}$ - topic frequency vector of each document
   - $\boldsymbol{z}$ - topic assignment of a sentence
7. Observed variable
   - $\boldsymbol{w}$ - words in a document

As mentioned earlier, for each document d with N_d sentences, we firstly draw a clusterId:k_d, then obtain a bag of topics $\boldsymbol{t}_d$ and topic ordering $\boldsymbol{\pi}_d$. Here, the bag of topics $\boldsymbol{t}_d$ is drawn in the traditional *LDA* [5] way under the multinomial distribution with parameter vector $\boldsymbol{\theta}_k \in \{\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_K\}$, the latter is shared among all members of cluster:k, indicating the similar interests in frequency in that group, with K different $\boldsymbol{\theta}$ parameters representing the different frequency distributions separately. Similarly, the topic ordering variable $\boldsymbol{\pi}_d$ is a permutation over topics 1 to T, indicating the topic

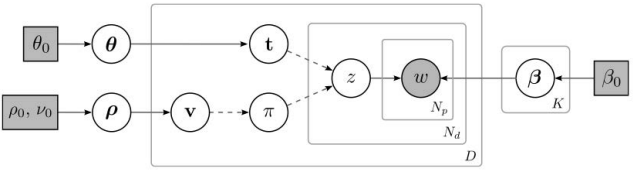


Fig. 2. *GMM* Generative Bayesian Graphical Model [6]

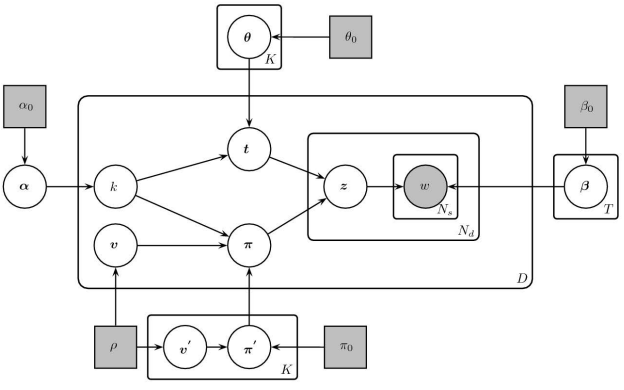


Fig. 3. *KMM* Generative Bayesian Graphical Model

occurrence order in the document, which is drawn from the standard *Mallows Model*. Combining $\{\boldsymbol{\theta}_k\}$ and $\{\boldsymbol{\pi}_k\}$, there are K different groups/clusters taking their individual aspect interests. Unlike these cluster related parameters, the T language models $\{\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_T\}$ are shared among the whole document corpus.

3.3 *Generalized Mallows Model*(*GMM*) over Permutations

To be self-contained, we give some necessary introduction on the *GMM* [6].

GMM exhibits two important properties: Firstly, *GMM* concentrates probability mass on one centroid ordering, which represents the in-domain documents' structural similarity; orderings which are close to the centroid will get high probability mass while those whose many elements have been moved will get less probability mass. Secondly, its parameter set scales linearly with the number of elements being ordered, making it sufficiently constrained and tractable for inference [6].

In *GMM*, the order of a permutation is represented as an inversion count vector $(v_1, \cdots, v_T)$, where v_T should always be 0.

The sum of all the components of the inversion count vector is the Kendall τ distance between the new ordering and the centroid ordering, which reflects the minimum number of adjacent elements' swaps needed to transform the ordering into the centroid.

The probability mass function of *GMM* is defined as follows:

$$GMM(\boldsymbol{v};\boldsymbol{\rho}) = \frac{e^{-\sum_j \rho_j v_j}}{\psi(\boldsymbol{\rho})} = \prod_{j=1}^{T-1} \frac{e^{-\rho_j v_j}}{\psi_j(\rho_j)} \tag{1}$$

where $\psi(\boldsymbol{\rho}) = \prod_j \psi_j(\rho_j)$ is the normalization factor, with:

$$\psi_j(\rho_j) = \frac{1 - e^{-(T-j+1)\rho_j}}{1 - e^{-\rho_j}} \tag{2}$$

For parameter $\rho_j > 0$, the *GMM* assigns the highest probability mass to each $v_j = 0$, and the probability mass drops exponentially as the inversion counts become bigger.

In our *KMM* model, we set each ρ_j to be a scalar number:ρ for simplicity, which reduces the *Generalized Mallows Model* to be the standard *Mallows Model*.

3.4 Generative Process

The generative process defines how the documents are produced by introducing hidden variables. In section 4, we shall present details of the learning process of these variables and parameters. The specific steps of the generative process are given below:

1. For each topic $t \in \{1, \cdots, T\}$, draw a language model $\boldsymbol{\beta}_t \sim Dirichlet(\beta_0)$, which specifies the word distribution over topic t.
2. For each cluster $k \in \{1, \cdots, K\}$, draw its parameters separately to reveal the different topic structures, as follows:
   - draw a topic distribution $\boldsymbol{\theta}_k \sim Dirichlet(\theta_0)$, which expresses how likely each topic would occur in each cluster;
   - draw a centroid ordering $\boldsymbol{\pi}_k^{'}$ by firstly drawing its corresponding inversion count vector:$\boldsymbol{v}_k^{'}$ according to Equation (1), and then convert it to the ordering:$\boldsymbol{\pi}_k^{'}$ which expresses the topic occurrence priority for each cluster.
3. Draw a cluster distribution $\boldsymbol{\alpha} \sim Dirichlet(\alpha_0)$, which indicates how likely each cluster is assigned to each document.
4. For each document d with N_d sentences:
   - draw a sample $k_d \sim Multinomial(\boldsymbol{\alpha})$ which indicates the clusterId of d;
   - draw a bag of topics $\boldsymbol{t}_d$ by sampling N_d times from $Multinomial(\boldsymbol{\theta}_k)$;
   - draw a topic ordering $\boldsymbol{\pi}_d$ by sampling an inversion count vector $\boldsymbol{v}_d \sim GMM(\rho, \boldsymbol{\pi}_k)$;
   - compute the topic assignment vector $\boldsymbol{z}_d$ for document d's N_d sentences by sorting $\boldsymbol{t}_d$ according to $\boldsymbol{\pi}_d$;
   - for each sentence s in document d:
     - sample each word w in s according to the language model of topic t=$z_{d,s}$: $w \sim Multinomial(\beta_{z_{d,s}})$.

3.5 Analysis of Parameters Setting

Parameter T represents the number of topics in data corpus, so the choice of T decides the granularity of the presented topics. Setting T=1 results in a big topic containing every word in vocabulary:V, while setting $T = |V|$ results in $|V|$ small topics, each of which contains only one word. But both of the extreme cases are meaningless. According to the common sense, different products take different number of aspects but mainly fall into the range of [2,10]. Also, because our user groups are modeled under the random space of aspect frequencies and orderings, a too small setting of T will restrict the ability to find enough discriminative groups because of the random space limitation. On the other hand, a too big setting of T will make our algorithm drop greatly in time efficiency during the learning process because of the random space explosion.

Parameter K decides the number of distinct groups. If we set K=1, we just get one global topic structure which reflects the human users' common consideration on product aspects as [6], so no discriminative user groups are available. On the other hand, if we only consider the different aspect priority orderings, there would be maximum $T!$ different groups. But in fact, we are only interested in the main discriminative groups, and also from the manufactories' point of view, the K should be decided according to their improvement ability. Indeed, manufactories are often not possible to identify and accommodate every specific customer need within a certain period of time.

4 Inference

We use Gibbs sampling [3] which is a stochastic inference method to infer the parameters. It is a kind of Markov Chain Monte Carlo which can construct a Markov chain over the hidden variable space, in which its stationary distribution converges to the target joint distribution.

In our Gibbs sampling process, there are in total four hidden variables to be resampled: $\{\boldsymbol{k}, \boldsymbol{t}, \boldsymbol{\pi}, \boldsymbol{\pi}^{'}\}$, $\boldsymbol{k}$ is the clusterId of a document, $\boldsymbol{t}$ indicates the topic frequency in a document, $\boldsymbol{\pi}$ determines the topic ordering in a document, and $\boldsymbol{\pi}^{'}$ determines the centroid topic ordering in each cluster. The resampling process of $\boldsymbol{t}$ and $\boldsymbol{\pi}$ is almost the same as that in [6], except for the following:

1. The topic occurrence statistics are computed at the level of each cluster collection instead of the whole corpus;
2. Instead of a global one, each document gets its own topic centroid ordering indicated by its clusterId.

All resampling equations are obtained by the following four steps:

1. Resample the topic count $\boldsymbol{t}_d$ for each document by resampling every sentence i in d(denoted as $s_{d,i}$):

$$\begin{aligned} p(t_{d,i} = t|\cdots) &\propto p(t_{d,i} = t|\boldsymbol{t}_{-(d,i)}, k_d, \theta_0) * p(\boldsymbol{w}_d|\boldsymbol{t}_d, \boldsymbol{\pi}_d, \boldsymbol{w}_{-d}, \boldsymbol{z}_{-d}, \beta_0) \\ &= \frac{N_{k_d}(\boldsymbol{t}_{-(d,i)}, t) + \theta_0}{N_{k_d}(\boldsymbol{t}_{-(d,i)}) + T\theta_0} * p(\boldsymbol{w}_d|\boldsymbol{t}_d, \boldsymbol{\pi}_d, \boldsymbol{w}_{-d}, \boldsymbol{z}_{-d}, \beta_0) \quad (3) \end{aligned}$$

Here, $N_{k_d}(\boldsymbol{t}_{-(d,i)}, t)$ is the total number of sentences assigned to topic t in cluster k_d without counting on $t_{d,i}$. And $N_{k_d}(\boldsymbol{t}_{-(d,i)})$ is the total number of sentences in cluster k_d except for $s_{d,i}$.

2. Resample the topic ordering $\boldsymbol{\pi}_d$ for each document by resampling every component of corresponding inversion count vector $\boldsymbol{v}_d$:

$$\begin{aligned} p(v_{d,j} = v|\cdots) &\propto p(v_{d,j} = v|\rho) * p(\boldsymbol{w}_d|\boldsymbol{t}_d, \boldsymbol{\pi}_d, \boldsymbol{w}_{-d}, \boldsymbol{z}_{-d}, \beta_0) \\ &= GMM(v;\rho) * p(\boldsymbol{w}_d|\boldsymbol{t}_d, \boldsymbol{\pi}_d, \boldsymbol{w}_{-d}, \boldsymbol{z}_{-d}, \beta_0) \end{aligned} \quad (4)$$

3. Resample the clusterId:k_d for each document:

$$\begin{aligned} p(k_d = k|\cdots) &\propto p(k_d = k|\boldsymbol{k}_{-d}, \alpha_0) * \prod_{d \in C_k} p(\boldsymbol{v}_d|\boldsymbol{\pi}_d, \boldsymbol{\pi}_k^{'}, \rho) * p(\boldsymbol{t}_d|k, \theta_0) \\ &= \frac{N(\boldsymbol{k}_{-d}, k) + \alpha_0}{N(\boldsymbol{k}_{-d}) + K\alpha_0} * \prod_{d \in C_k} [GMM(\boldsymbol{v}_d; \boldsymbol{\pi}_d, \boldsymbol{\pi}_k^{'}, \rho) * \prod_{s \in d} \frac{N_k(\boldsymbol{t}_{-d}, t_s)}{N_k(\boldsymbol{t}_{-d})}] \end{aligned} \quad (5)$$

Here, $N(\boldsymbol{k}_{-d}, k)$ is the total number of documents assigned to clusterId:k without counting on d. $N(\boldsymbol{k}_{-d})$ is the total number of documents except for d. $N_k(\boldsymbol{t}_{-d}, t_s)$ is the number of sentences assigned to t_s in cluster:k without counting on d. And $N_k(\boldsymbol{t}_{-d})$ is the number of sentences except for those from d.

4. Resample the centroid topic ordering $\pi^{'}$ of each cluster by resampling corresponding inversion count vector $v^{'}$:

$$p(v_{k,j}^{'} = v|\cdots) \propto \prod_{d \in C_k} p(\boldsymbol{v}_d|\boldsymbol{\pi}_d, \boldsymbol{\pi}_k^{'}, \rho) = \prod_{d \in C_k} GMM(\boldsymbol{v}_d; \boldsymbol{\pi}_d, \boldsymbol{\pi}_k^{'}, \rho) \quad (6)$$

```
1: init centroid topic ordering for clusters: {π'_1, ··· , π'_K}
2: init clusterId for all documents: {k_1, ··· , k_D}, k_d ∈ {1, ··· , K}
3: init topic counts for all documents: {t_1, ··· , t_D}
4: init topic ordering for all documents: {π_1, ··· , π_D} by initializing its v_d
   combining with π'_{k_d}
5: init topic assignments for all documents: {z_1, ··· , z_D} by combining t_d and π_d
6: for it = 1 to MaxIteration step 1 do
7:    for d = 1 to D step 1 do
8:       remove statistic on d
9:       resample t_d according to Equation (3)
10:      resample π_d according to Equation (4)
11:      resample k_d according to Equation (5)
12:      add back statistic on d
13:   end for
14:   for k = 1 to K step 1 do
15:      resample π'_k according to Equation (6)
16:   end for
17: end for
```

Algorithm 1: Resampling Algorithm by Gibbs Sampling

In Equations (3) and (4), the document probability is computed in the same way as [6] except that statistics are taken for each cluster separately:

$$p(\boldsymbol{w}_d|\boldsymbol{t}_d, \boldsymbol{\pi}_d, \boldsymbol{w}_{-d}, \boldsymbol{z}_{-d}, \beta_0) = p(\boldsymbol{w}_d|\boldsymbol{z}, \boldsymbol{w}_{-d}, \beta_0)$$
$$= \prod_{t=1}^{T} \int_{\beta_t} p(\boldsymbol{w}_d|\boldsymbol{z}_d, \boldsymbol{\beta}_t) p(\boldsymbol{\beta}_t|\boldsymbol{z}, \boldsymbol{w}_{-d}, \beta_0) d\boldsymbol{\beta}_t \tag{7}$$

The overall resampling algorithm can thus be described by Algorithm 1.

During resampling, we try every possible topic assignment(T in total) to every sentence(N_d in total) in a document(D in total). So the time complexity of one iteration would be $O(D * N_d * T)$.

5 Experiment

In this section, we apply our algorithm on several real-world online review datasets, demonstrate the competitive grouping performance and show how different groups with specific aspect interests are discovered from data collections.

5.1 Datasets

1. Amazon(*AZ*)[2]
   The Amazon review dataset from *www.amazon.com* crawled by [12] in 2006 contains reviews of manufactured products. We choose several product categories and select a subset under several memberIds(reviewers) for each category. With the availability of memberId attribute which indicates the author of the review text, we treat it as the true class label, through which we evaluate the user grouping performance in the form of review clustering, by checking whether or not reviews contributed by the same author will be clustered into the same group. When preprocessing the data, review spams are found to exist in the form of duplicate reviews with different productIds under the same memberId. So we remove those spams by pair-wise checking through *TF-IDF* based the cosine similarity.
2. OpinionRank(*OR*)[3]
   The OpinionRank review dataset [9] contains full reviews for cars and hotels from Edmunds and Tripadvisor. We choose the hotel reviews under *city:beijing* to demonstrate the resulting groups reflecting users' discriminative topic interests, and each topic is presented by its top 20 words in the corresponding language model.

The general statistics over those datasets are shown as Table 1.

[2] http://131.193.40.52/data/
[3] http://kavita-ganesan.com/entity-ranking-data/

Table 1. Statistics on Online Review Datasets

Dataset	*AZ:Camera*	*AZ:Computer*	*AZ:*$Apparel_1$	*AZ:*$Apparel_2$	*OR:hotel:beijing*
#Reviews	192	67	110	92	5256
avgTxtSize	1303	1482	462	860	983

5.2 Evaluation Methodology

By considering the memberId as the true class label, we evaluate the clustering performance via two popular criterion functions: Purity [22], Normalized Mutual Information(*NMI*) [23], as summarized below:

– Purity

$$Purity = \frac{1}{n}\sum_{i=1}^{K} Max(n_{i*}) \tag{8}$$

– *NMI*

$$NMI(X,Y) = \frac{I(X,Y)}{\sqrt{H(X)H(Y)}} = \frac{\sum_{i=1}^{K}\sum_{j=1}^{C} n_{ij} log(\frac{n*n_{ij}}{n_i*n_j})}{\sqrt{(\sum_{i=1}^{K} log(\frac{n_i}{n}))(\sum_{j=1}^{C} n_j log(\frac{n_j}{n}))}} \tag{9}$$

In the above equations, n is the total number of documents, n_{ij} is the member size of class j in cluster i, n_i is the member size of cluster i and n_j is the member size of class j. X is the cluster label variable for cluster assignments while Y is the actual class label variable. K represents the total cluster number while C represents the class number.

5.3 Parameter Setting

According to [10], we set the Dirichlet hyper-parameters as follows: $\alpha_0 = 50.0/K$, $\beta_0 = 0.1$, $\theta_0 = 50.0/T$. For parameter $\boldsymbol{\pi}_0$, it is regarded as the natural permutation over T topics:$\{1, \cdots, T\}$ without loss of generality. To simplify the learning process, we set the parameter ρ to be a scalar, instead of being a T-1 dimensional vector in *GMM*, which reduces the *Generalized Mallows Model* to a standard *Mallows Model*. By experience, setting $\rho = 1$ results in a good balance between the real-world ordering randomness and punishment of ordering dispersion.

For dataset *AZ*, we set $T = 10$ by experience, and K is set to be the distinct count of memberId (which is 3,3,3,5 for *AZ:Camera, AZ:Computer,* $AZ{:}Apparel_1$,$AZ{:}Apparel_2$ respectively), whereas for *OR*, we don't have their author Ids, so we set $T = 6$ and $K = 3$ to reflect the word distribution of six topics and specific aspect interests of three groups.

5.4 Comparison of Grouping Performance with K-means Baseline

To demonstrate the competitive performance of user grouping, we do experiment on *AZ:Camera, AZ:Computer,* $AZ{:}Appeal_1$ and $AZ{:}Appeal_2$, each of which contains reviews selected from 3, 3, 3 and 5 different reviewers. Accordingly, the K is set to 3, 3, 3 and 5 respectively.

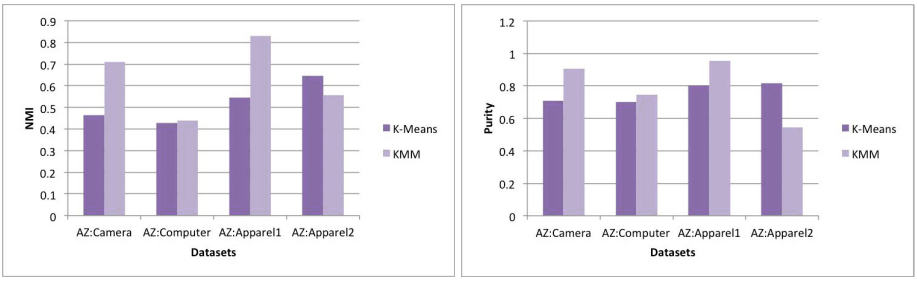


Fig. 4. Clustering performance on the four datasets

We use one of the most popular clustering algorithm K-Means as baseline, and the pair-wise similarity is computed using the standard cosine similarity on the *TF-IDF* score based *Vector Space Model*.

The performance comparison is shown in Fig.4. As observed, our algorithm beats the K-Means baseline for datasets:*AZ:Camera, AZ:Computer* and *AZ:*$Apparel_1$, with respective improvement being around 54.3%, 2.3% and 53.7% in terms of *NMI*.

When we look into the *AZ:*$Apparel_2$ which contains reviews selected from 5 distinct reviewers, we find that the performance is not as good as K-Means. The reason is due to that our clustering result is based on the identification of the common taste in product aspect interests, while the K-Means is based on the *Vector Space Model* in individual review text. With the limited number of reviews for each reviewer, K-Means tends to be easier to discriminate individual reviewer, whereas in our model different reviewers could be clustered into one group when they share common aspect interests. Also, it indicates that our algorithm is not adapted for the task of individual person identification which requires much more refined individual characteristics.

5.5 Illustration of User Groups with Specific Aspect Interests

To illustrate the validity of user grouping, we further do experiment on *OR:hotel:beijing* by setting $K = 3, T = 6$.

The parameter $\theta_{k,t}$ can be estimated by :

$$\hat{\theta}_{k,t} = \frac{N_\theta(k,t) + \theta_0}{N_\theta(k) + T\theta_0} \tag{10}$$

where $N_\theta(k,t)$ is the total number of sentences assigned to topic t in cluster k, and $N_\theta(k)$ is the total number of sentences in cluster k.

We identify six aspects: $\{T_1 : General_1$, $T_2 : Room$, $T_3 : Food$, $T_4 : L\&S$, $T_5 : General_2, T_6 : Service\}$[4]. The aspects ordering: $\boldsymbol{\pi}'$ and frequency :$\boldsymbol{\theta}_k$ (computed based on Equation (10)) for all three clusters are listed in Table 2. We can see that users in C_0 pay more attentions to *Room* and don't care *Service* very much. On the other hand, users in C_1 care about the *Service* mostly, while those in C_2 are mostly interested

[4] See Appendix A for their definitions.

in *Food* and less interested in *L&S*. Besides the topic frequencies, we also see that while their topic orderings are quite different, they all start from $General_1$, which reflects the fact that people often give a general description first, followed by different aspects w.r.t. their own interests.

Table 2. Three user groups with specific topic ordering and frequency

C_0		C_1		C_2	
Ordering	Freq(%)	Ordering	Freq(%)	Ordering	Freq(%)
$General_1$	12.7	$General_1$	12.4	$General_1$	9.1
Room	**25.8**	$Room$	13.5	$General_2$	13.5
$Service$	10.7	$L\&S$	10.3	$Room$	17.2
$Food$	17.5	$Food$	11.6	**Food**	**37.7**
$L\&S$	16.9	**Service**	**35.5**	$Service$	13.7
$General_2$	16.4	$General_2$	16.7	$L\&S$	8.8

This kind of grouping information can help hotel managers make improvement in respective aspects with consideration on customers' specific interests, and also help travel advisors develop different suggestions to accommodate various customers.

6 Conclusion

The online data such as review texts can reveal much useful business information. The *KMM* topic model proposed in this paper aims at efficiently identifying hidden user groups accompanied by a detailed explanation in the forms of aspect frequency and ordering priority. The result can be used widely such as in product improvement, marketing strategy development or customer-targeting online advertisement plans.

As shown by the work reported in this paper, the incorporation of structural restrictions into a traditional bag-of-topics topic model such as *LDA* can greatly improve the model's expressive power, which is important to automatic text understanding. We plan to develop an even more complicated and adaptive structural model in our subsequent research.

Acknowledgment. The work described in this paper has been supported by the NSFC Overseas, HongKong & Macao Scholars Collaborated Researching Fund (61028003) and the Specialized Research Fund for the Doctoral Program of Higher Education, China (20090141120050).

References

1. Abdul-Mageed, M., Diab, M.T., Korayem, M.: Subjectivity and sentiment analysis of modern standard arabic. In: ACL (Short Papers) 2011, pp. 587–591 (2011)
2. Beineke, P., Hastie, T., Manning, C., Vaithyanathan, S.: An Exploration of Sentiment Summarization. In: Proceeding of AAAI, pp. 12–15 (2003)

3. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
4. Blei, D.M., Griffiths, T.L., Jordan, M.I.: The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. J. ACM 57, 7:1–7:30 (2010)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003)
6. Chen, H., Branavan, S.R.K., Barzilay, R., Karger, D.R.: Content modeling using latent permutations. J. Artif. Intell. Res. (JAIR) 36, 129–163 (2009)
7. Fligner, M.A., Verducci, J.S.: Distance based ranking models. Journal of the Royal Statistical Society. Series B (Methodological) 48(3), 359–369 (1986)
8. Gamon, M., Aue, A., Corston-Oliver, S., Ringger, E.: Pulse: Mining Customer Opinions from Free Text. In: Famili, A.F., Kok, J.N., Peña, J.M., Siebes, A., Feelders, A. (eds.) IDA 2005. LNCS, vol. 3646, pp. 121–132. Springer, Heidelberg (2005)
9. Ganesan, K., Zhai, C.: Opinion-based entity ranking. Information Retrieval (2011)
10. Griffiths, T.L., Steyvers, M.: Finding scientific topics. PNAS 101(suppl. 1), 5228–5235 (2004)
11. Gruber, A., Rosen-Zvi, M., Weiss, Y.: Hidden Topic Markov Models. In: Artificial Intelligence and Statistics (AISTATS), San Juan, Puerto Rico (2007)
12. Jindal, N., Liu, B.: Opinion spam and analysis. In: Proceedings of the International Conference on Web Search and Web Data Mining, WSDM 2008, pp. 219–230 (2008)
13. Jindal, N., Liu, B., Lim, E.-P.: Finding unusual review patterns using unexpected rules. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM 2010, pp. 1549–1552 (2010)
14. Jordan, M. (ed.): Learning in Graphical Models. MIT Press (1999)
15. Li, W., McCallum, A.: Pachinko allocation: Dag-structured mixture models of topic correlations. In: ICML (2006)
16. Liu, B.: Opinion observer: Analyzing and comparing opinions on the web. In: Proceedings of the 14th International Conference on World Wide Web, WWW 2005, pp. 342–351 (2005)
17. Mei, Q., Liu, C., Su, H., Zhai, C.: A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In: Proceedings of the 15th International Conference on World Wide Web, WWW 2006, pp. 533–542 (2006)
18. Mukherjee, A., Liu, B., Wang, J., Glance, N., Jindal, N.: Detecting group review spam. In: Proceedings of the 20th International Conference Companion on World Wide Web, WWW 2011, pp. 93–94 (2011)
19. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT 2005, pp. 339–346 (2005)
20. Purver, M., Griffiths, T.L., Körding, K.P., Tenenbaum, J.B.: Unsupervised topic modelling for multi-party spoken discourse. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, ACL-44, pp. 17–24 (2006)
21. Titov, I., McDonald, R.: Modeling online reviews with multi-grain topic models. In: Proceeding of the 17th International Conference on World Wide Web, WWW 2008, pp. 111–120 (2008)
22. Zhao, Y., Karypis, G.: Criterion functions for document clustering: Experiments and analysis. Tech. rep., University of Minnesota (2002)
23. Zhou, X., Zhang, X., Hu, X.: Semantic smoothing of document models for agglomerative clustering. In: Proceeding 20th International Joint Conf. Artificial Intelligence, IJCAI 2007, pp. 2928–2933 (2007)

A Aspect Definition in the Form of Word Distribution

All six aspects' definitions are shown in Table 3, (only top 20 words are presented) where $L\&S$ means the $Location$ & $Surrounding$. The probability is estimated by:

$$\hat{\beta}_{t,w} = \frac{N_\beta(t,w) + \beta_0}{N_\beta(t) + V\beta_0} \tag{11}$$

where $N_\beta(t,w)$ is the total number of times word w is assigned to topic t, $N_\beta(t)$ is the total number of words assigned to topic t, and V is the vocabulary size.

As observed from Table 3, the aspects are clearly identified by their top words except for the $General_1$ and $General_2$ which contain some general words instead of specific aspect related words. This is reasonable as people may express some general idea about products, so we treat this kind of general topics as "General Aspects".

Table 3. Top 20 words in aspects' word distributions $\{T_1, T_2, T_3, T_4, T_5, \}$

$General_1$	%	$Room$	%	$Food$	%	$L\&S$	%	$General_2$	%	$Service$	%
hotel	6.62	**room**	4.73	**breakfast**	3.21	hotel	3.87	hotel	4.69	hotel	2.85
stayed	3.64	**rooms**	2.29	hotel	1.98	**location**	2.18	stay	2.10	**staff**	2.34
beijing	3.00	hotel	1.57	good	1.69	city	2.09	beijing	1.80	**english**	1.34
nights	1.79	**clean**	1.40	**food**	1.64	**walk**	2.06	staff	1.18	**helpful**	1.09
stay	1.36	**bathroom**	1.32	**buffet**	1.38	**shopping**	1.50	good	1.18	taxi	0.97
china	1.03	**comfortable**	1.05	room	1.31	**forbidden**	1.47	service	1.00	great	0.93
trip	0.84	**bed**	1.04	**restaurant**	1.26	**subway**	1.34	great	0.97	room	0.81
hotels	0.81	nice	1.02	great	0.91	**street**	1.31	recommend	0.94	**desk**	0.77
great	0.78	**shower**	0.83	**chinese**	0.89	beijing	1.24	room	0.72	quot	0.74
room	0.75	**large**	0.77	service	0.87	**taxi**	1.01	hotels	0.69	**service**	0.73
location	0.75	good	0.74	pool	0.86	**walking**	1.00	location	0.68	**friendly**	0.72
booked	0.72	floor	0.72	nice	0.70	**square**	0.97	place	0.60	chinese	0.69
days	0.69	**beds**	0.63	staff	0.66	great	0.91	time	0.58	**concierge**	0.69
good	0.65	**tv**	0.62	**western**	0.62	good	0.89	definitely	0.57	beijing	0.63
reviews	0.60	**spacious**	0.59	**free**	0.62	**station**	0.88	go	0.54	day	0.62
night	0.59	great	0.57	excellent	0.62	**minutes**	0.87	quot	0.52	good	0.60
business	0.58	**water**	0.52	**restaurants**	0.59	**close**	0.82	rooms	0.52	front	0.60
plaza	0.53	quot	0.49	quot	0.58	**located**	0.81	back	0.49	driver	0.59
star	0.50	staff	0.48	day	0.57	**distance**	0.79	china	0.49	wall	0.58
holiday	0.49	modern	0.47	internet	0.51	**area**	0.78	star	0.49	time	0.54

An Algorithm for the Automatic Estimation of Image Orientation

Mariusz Borawski and Dariusz Frejlichowski

West Pomeranian University of Technology, Szczecin,
Faculty of Computer Science and Information Technology,
Zolnierska 52, 71-210, Szczecin, Poland
{mborawski,dfrejlichowski}@wi.zut.edu.pl

Abstract. The paper presents a new method for the determination of an image orientation — the distinction between portrait- and landscape-oriented images. The algorithm can be applied to photographs taken outdoors. The approach is based on the use of a subpart of an image containing the sky. For determining the orientation the run of the standard deviation increment is analysed. It is obtained for the processed image and matched by means of correlation with the same characteristic of the sub-block of an image containing the sky.

Keywords: Image Orientation Detection, Standard Deviation Increment, Polar Transform.

1 Introduction

The increasing popularity of digital cameras and inexpensive scanners resulted in a significant increase of multimedia data (mostly in the form of digital images) stored on our Personal Computers in everyday life. It is very common to keep thousands and thousands of images on a hard drive. In order to make this task easier the algorithms developed for Content Based Image Retrieval could be applied, resulting in the creation of specialised image management systems, assisting the user in storing, indexing, browsing, and retrieving images from his database ([1]). Moreover, one of the crucial abilities expected for the mentioned group of systems is the automatic detection of image orientation ([1]). The user expects that an image to be displayed, regardless of its source — a digital camera or scanner — will be provided in a correct orientation. This process is expected to be performed automatically.

The automatic detection of image orientation is not an easy task. Therefore, it is obvious that several attempts to solve this problem have already been made so far, however with varying results. Chronologically, first algorithms developed for the problem were connected with a very specific application — the detection of a scanned document orientation (e.g. [2,3]). However, the constrained nature of this problem (text cues) and the impossibility to apply the proposed algorithms to natural images is often called (e.g. [4]).

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 336–344, 2012.

The first algorithms developed particularly for natural images were based on the low-level features (e.g. [5,6,7]). However, it was stressed that this approach can be insufficient, since humans usually use more sophisticated clues for the detection of an image orientation, for example facilitated by the semantic scene analysis ([4]). An example, based on the detection of some characteristic elements in an image (e.g. face, sky, white ceiling, wall, grass), has been proposed in [8]. This process constituted an addition to the use of low-level features. Similarly, in [9] some high-level cues — objects with distinguishable orientation and objects with a usually fixed position — along with low-level features have been used.

The image orientation detection was in [1] performed by means of spatial colour moments for feature representation and Bayesian learning framework for classification. The class-conditional probability density functions were estimated by means of the Learning Vector Quantization. In [10] the colour moments were also used. Additionally, the Edge Direction Histogram was applied during the stage of feature representation, and AdaBoost at the classification. An unusual approach for the discussed problem was proposed in [11]. The textures within an image were analysed. The method was based on the assumption that more textured areas are located in the lower part of an image. At the stage of classification the AdaBoost was again applied.

In the paper a new approach is proposed. It is based on the analysis of the sky visible within an image. It means that this approach can be applied only for outdoor images. This method consists in the observation of the fact that texture ('surface quality') of the sky visible within an image is different for landscape- and portrait- oriented images. In order to perform this distinction automatically the correlation of increments of standard deviations, calculated for the spectra of image subparts is determined.

The rest of the paper is organised in the following way. The second section describes the proposed algorithms in a detailed way. The third section provides some experimental results. Finally, the last section briefly concludes paper.

2 The Approach for Automatic Image Orientation Estimation

Roughly speaking, the approach proposed in this paper is composed of the following steps: image downscaling, localisation of the image subpart containing the sky, calculation of the logarithm of the absolute spectrum for it, transformation of the obtained representation into polar co-ordinates, and finally, determination of the image orientation. The downscaling in the first step is necessary for speeding up the performance of the later stages. It can be applied, since for the proposed approach the image size does not have to be large. In the experiments described in the following section the size 320×200 was sufficient.

The localisation of the sky within an image is based on the colour. It is quite characteristic. However, there are many objects within an image that can be similar in terms of colour to it. Therefore, additionally the variability of the localised image subpart has to be analysed, since it is significantly less changeable.

The investigation of the colour and variability was performed for each pixel inside the localised area, including the information about its nearest neighbouring pixels. More precisely, in order to carry out this task, for each pixel a subpart of the image with side equal to 30 pixels was taken, with the processed pixel in the centre of it. For each element inside this area the following conditions were verified:

$$\begin{cases} b_{\mathrm{b}}(x,y) \geq 150 \ , \\ b_{\mathrm{b}}(x,y) \geq b_{\mathrm{g}}(x,y) + 10 \ , \\ b_{\mathrm{g}}(x,y) \geq b_{\mathrm{r}}(x,y) + 10 \ , \\ b_{\mathrm{g}}(x,y) - 100 \leq b_{\mathrm{r}}(x,y) \ \ , \end{cases} \tag{1}$$

where: $b_r(x, y)$ is the red component for the analysed pixel, $b_g(x, y)$ is the green one, and $b_b(x, y)$ the blue one.

If all the conditions from the above equation are fulfilled for all pixels within an analysed block, the block is initially marked as the one belonging to the sky. It is later converted to grey-scale:

$$b_{\mathrm{grey}}(x,y) = 0.299 b_{\mathrm{r}}(x,y) + 0.587 b_{\mathrm{g}}(x,y) + 0.114 b_{\mathrm{b}}(x,y) \ \ , \tag{2}$$

where b_{grey} is a pixel within the localised block after the conversion to grey-scale.

For the grey-scaled block the standard deviation is calculated and compared with the threshold, experimentally established as equal to 10. If the analysed value is lower or equal to the threshold, it is eventually marked as being part of the sky. For the pixels inside the 30×30 block, fulfilling the above condition, other sub-blocks (19×19 pixels size) are created. Again, the analysed pixel is in the centre of the sub-block. The sub-block has to entirely be placed within the 30×30 block. The ones crossing the borders of the block are rejected. For the sub-blocks the absolute spectrum of the two-dimensional Fourier transform is calculated. It is invariant to the cyclic shifting of the image vertically and horizontally. This property is useful, since the sky area in an image is undulated horizontally, and this undulation is shifted in various parts of an image. Thanks to the use of the absolute spectrum this shifting has no influence on the results.

The spectrum of an image has large values for low-frequency components and very small ones for high frequencies. It hinders the comparisons between the spectra or their subparts, with the low and high frequency components. In order to solve this problem the logarithm of the spectra can be used. It reduces the differences between very large and very small values. Hence, in the proposed approach, the natural logarithm was applied. However, it cannot be calculated for the zero value, that is the lowest value that can appear in the absolute spectrum. In order to omit this limitation, a very small value ($1 \cdot 10^{-5}$) was added to all the elements in the absolute spectrum.

Since the image of the clear sky is undulated from the top to the bottom, its absolute spectrum is very characteristic. In its centre a high peak representing the mean value is located. Through this peak a line connected to the undulation of the sky is crossing. Unfortunately, the values for this line are only slightly

higher than the noise. Hence, they can be observed only for sub-images containing a clear sky. Any objects in the sky, including clouds, make the detection of this line impossible. Moreover, in the spectrum one can detect a second line, perpendicular to the previous one and crossing through the centre of the spectrum as well (see Fig. 1). The second line is connected to the horizontal undulation of the sky image and it is always significantly smaller. This property enables to easily distinguish the two lines which in turn results in the determination of the image orientation.

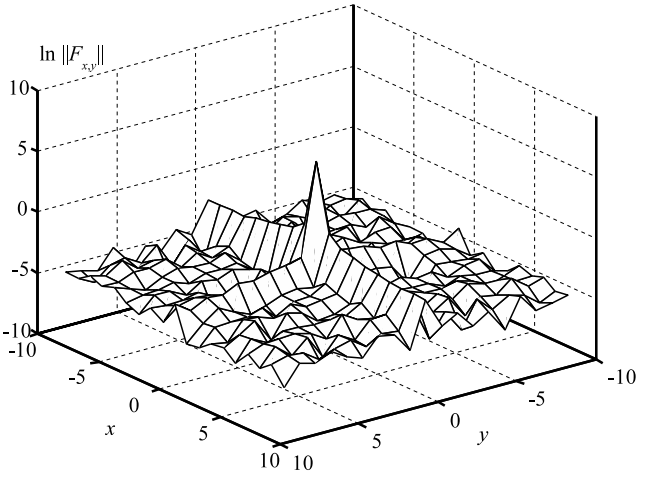


Fig. 1. The natural logarithm of the absolute spectrum calculated for the sub-block

Each sub-block is transformed from Cartesian to polar co-ordinates. This transform is very efficient and hence common in various problems connected with image and object representation (e.g. [12,13]), however the selection of the origin of this transformation is very crucial for further work ([14]). Here, the location of the constant component representing zero frequencies is used. Usually, the algorithms for the derivation of the 2D discrete Fourier transform locate it in the edge of the spectrum. However, for convenient calculations we can shift it into the centre of the spectrum. The spectrum is scanned beginning with the constant component towards the high frequency components along the lines under the various angles. The spectrum symmetry property for images enables the scanning to be performed in the range of angles from -90 to 90 degrees. The step equal to 10 degrees was assumed here. For the values scanned along the lines the standard deviation is calculated. In result, we obtain 18 standard deviation values, for various angles of scanning the spectrum.

The run of the standard deviation calculated by means of the abovementioned approach for the subpart of an image covering the sky is very characteristic. For the zero angle very low values are obtained, because the derivation of the standard deviations is realised along the lines corresponding to the undulation of the

sky (see Fig. 2). For the angles close but not equal to zero the standard deviation is calculated basing on the fragment of the line representing the undulation of the sky and low values outside this line. Hence, the standard deviation has large values. For the third case — the angles far from zero — in the calculation of the standard deviation only the area outside the line corresponding to the undulation of the sky is considered. Therefore, the standard deviation values have small values. As a result, in the diagram of the standard deviations close to the zero angle a very characteristic shape appears which is similar to the letter 'M'. A similar, but smaller shape appears close to the 90 degrees angle and it is connected with the horizontal undulation of the sky.

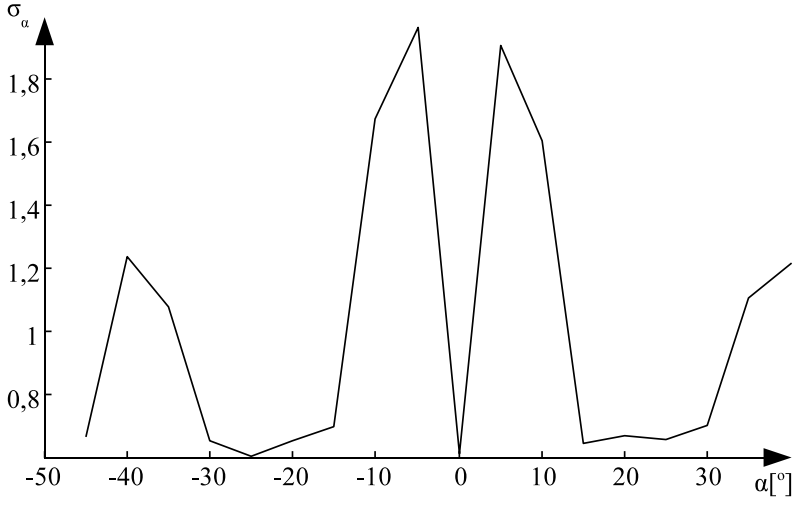


Fig. 2. The diagram of the standard deviations of the spectrum of the sub-block in the polar co-ordinates

The detection of the 'M' shape within the run of the standard deviation can be hampered by various distortions, e.g. small objects or local brightening or darkening in the sky. In order to avoid this problem, within a block we have to calculate the average values of all scanned pixels within particular sub-blocks, and for those points calculate the standard deviations. This enables to remove the deformations of the spectrum and the detection of the 'M' shape becomes facilitated. Moreover, the distinction between horizontal and vertical undulation of the sky is possible.

The determination of the image orientation can be based on the comparison between the obtained standard deviations and the template. The run similar to the one provided in Fig. 2. can be assumed as the template. Since the shift of the runs according to each other has to be taken into account, the correlation could be efficiently used at the matching stage. However, it cannot be applied to the matching of standard deviation runs, because of the character of the

arithmetic operations performed on the standard deviations. They are absolute and cannot be negative. On the other hand, the correlation works with the negative, zero, and positive values. Hence, in order to be able to match the standard deviations we have to convert the notions of the absolute values into the relative ones. This process can be performed by making an assumption that the origin of standard deviations is the zero value. However, since we allow the negative values, the notion 'standard deviation' can no longer be used. Instead, we are working with the increments of the standard deviation — the result of a subtraction of standard deviations, which can be negative.

The correlation can be interpreted as the derivation of the scalar product of the unit vectors. For the perpendicular vectors the resultant value equal to zero is obtained, while the $+1$ and -1 for the parallel ones. In order to calculate the scalar product firstly we have to define a vector containing the increments of the standard deviation ([15]):

$$A = (\Delta\sigma_{A1}, \Delta\sigma_{A2}, \ldots, \Delta\sigma_{An}) \ . \tag{3}$$

The scalar product for this kind of vector can be defined in the same way as the regular scalar product ([15]):

$$(A, B) = \sum_{i=1}^{N} \Delta\sigma_{Ai} \Delta\sigma_{Bi} \ , \tag{4}$$

where N denotes the dimension of a vector space.

It is important to note that (A, B) is not a standard deviation and it is also not its increment. However, in some situations it can represent it.

Basing on the definition of the scalar product we can obtain the formula for the correlation ([15]):

$$a_{\mathrm{rel}}(x, y) \circ b_{\mathrm{rel}}(x, y) = \frac{1}{\|A_{\mathrm{rel}}\| \, \|B_{\mathrm{rel}}\|} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} a_{\mathrm{rel}\Delta\sigma}(i, j) \, b_{\mathrm{rel}\Delta\sigma}(x + i, y + j) \ , \tag{5}$$

where the length of the vectors $\|A_{\mathrm{rel}}\|$ i $\|B_{\mathrm{rel}}\|$ can be calculated by means of the formula ([Bor07]):

$$\|A_{\mathrm{rel}}\| = \sqrt{(A_{\mathrm{rel}}, A_{\mathrm{rel}})} = \sqrt{\sum_{x=0}^{m-1} \sum_{y=0}^{n-1} a_{\mathrm{rel}\Delta\sigma}(x, y)^2} \ . \tag{6}$$

A_{rel} i B_{rel} represent the relative part of the vectors A and B. They are obtained through the subtraction of the average values of A and B from their co-ordinates.

As a result of the derivation of the correlation an array is obtained, in which the highest value indicates the degree of similarity between matched arrays of the increments of the standard deviations. The location of this value indicates the shifting of the values in the arrays according to one another. If the maximal value

is obtained at the zero index of the matching array, one can determine the landscape orientation of an image. The index value equal to 9 indicates the portrait orientation of an image. In practice, the values can be slightly extended: indices 0–4 and 14–17 indicate the landscape orientation, and 5–13 — the portrait one. Additionally, one can utilize the correlation coefficient for the estimation of the similarity between the analysed run of the standard deviation increment and the one for the sky.

3 Experimental Results

The method described in the previous section was tested by means of 100 digital images containing the sky. The photos were taken during various times of the day, various geographical locations and elevations. Some of the photos were obtained by means of a digital camera (49) and some by means of an analog one (51). In Fig. 3, the results of the proposed method for orientation detection are presented. In six cases the obtained result was wrong — two for the images obtained using the digital camera, and four for the analog one. Those cases were marked out using the ellipse. In 14 cases, crossed in Fig. 3, the method rejected the results as the ambiguous ones.

Fig. 3. The experimental result of the proposed approach for image orientation detection

Table 1 provides the calculated angles, maximal correlation values, and the percentage P of the correlation coefficients properly determining the orientation of an image. The second and third measures can be used as the estimators of the proposed approach.

Table 1. The obtained angles, correlation values, and the percentage P of the correlation coefficients properly determining an orientation

Row in Fig. 3	1	2	3	4	5	6	7	8	9	10	11	12	Col. in F. 3
Angle [°]	0	–	90	0	0	90	90	–	–	90	90	90	
Correlation val.	1	–	0.96	1	1	1	0.97	–	–	0.97	1	1	1
P [%]	100	–	100	100	97	100	100	–	–	100	100	100	
Angle [°]	90	0	0	90	90	0	0	90	90	0	0	90	
Correlation val.	1	1	1	1	1	0.98	1	0.98	0.9	0.97	1	1	2
P [%]	100	100	93	100	100	100	100	100	92	73	96	100	
Angle [°]	0	–	0	0	0	0	90	0	–	–	90	0	
Correlation val.	1	–	0.99	1	0.99	0.95	0.98	1	–	–	0.99	0.59	3
P [%]	100	–	82	100	100	100	100	100	–	–	100	100	
Angle [°]	0	0	0	0	90	0	0	0	0	0	90	90	
Correlation val.	1	0.6	0.99	1	1	1	1	0.99	0.96	0.99	1	1	4
P [%]	95	100	100	84	100	94	87	100	100	76	76	100	
Angle [°]	90	90	0	0	0	90	90	0	0	0	0	0	
Correlation val.	0.98	1	0.93	0.99	1	0.99	0.54	0.52	0.99	0.99	0.98	0.99	5
P [%]	96	100	100	98	100	100	74	100	100	100	100	100	
Angle [°]	–	0	90	90	90	90	0	0	–	90	90	–	
Correlation val.	–	0.88	0.95	0.94	1	1	1	0.47	–	0.96	0.84	–	6
P [%]	–	83	90	98	98	58	96	100	–	100	100	–	
Angle [°]	–	–	90	–	90	0	90	90	–	–	0	0	
Correlation val.	–	–	1	–	1	1	0.98	0.77	–	–	0.96	1	7
P [%]	–	–	56	–	100	100	100	100	–	–	97	100	
Angle [°]	0	0	0	0	90	0	0	0	90	0	90	90	
Correlation val.	1	0.66	1	1	0.99	0.78	0.99	0.96	0.99	0.65	0.96	0.82	8
P [%]	100	100	100	98	54	100	100	100	65	100	100	57	
Angle [°]	90	90	0	90									
Correlation val.	1	1	0.98	0.99									9
P [%]	100	100	100	99									

4 Conclusions

In the paper an approach for the image orientation determination was proposed and experimentally evaluated. The method consists of the following steps:

- image downscaling,
- localisation of the image subpart containing the sky,
- calculation of the logarithm of the absolute spectrum for it,
- transformation of the obtained representation into polar co-ordinates.
- determination of the image orientation.

The method makes use of the subparts of an image containing the sky. The method failed solely in 6 out of 100 experimental images, obtained by means of various technologies (digital, analog), in various geographical locations and times of the day. In four cases the wrongly judged images were aken using the analog camera. Similarly, most of the images rejected by the method were the analog ones. Hence, one can conclude that the approach works better with the digital images.

References

1. Vailaya, A., Zhang, H., Yang, C., Liu, F.-I., Jain, A.K.: Automatic Image Orientation Detection. IEEE Transactions on Image Processing 11(7), 746–755 (2002)
2. Hinds, S.C., Fisher, J.L., D'Amato, D.P.: A Document Skew Detection Method Using Run-Length Encoding and the Hough Transform. In: 10th International Conference on Pattern Recognition, vol. 1, pp. 464–468 (1990)
3. Akiyama, T., Hagita, N.: Automated Entry System for Printed Documents. Pattern Recognition 23(11), 1141–1154 (1990)
4. Luo, J., Crandall, D., Singhal, A., Boutell, M., Gray, R.T.: Psychophysical Study of Image Orientation Perception. Spatial Vision 16(5), 429–457 (2003)
5. Wang, Y., Zhang, H.: Content-Based Image Orientation Detection with Support Vector Machines. In: Proc. of the IEEE Workshop on Content-based Access of Image and Video Libraries (2001)
6. Wang, Y., Zhang, H.: Detecting image orientation based on low-level visual content. In: Computer Vision and Image Understanding, vol. 93(3), pp. 328–346 (2004)
7. Lyu, S.: Automatic Image Orientation Determination with Natural Image Statistics. In: Proc. of the 13th Annual ACM International Conference on Multimedia, MULTIMEDIA 2005, pp. 491–494 (2005)
8. Luo, J., Boutell, M.: A Probabilistic Approach to Image Orientation Detection via Confidence-Based Integration of Low-Level and Semantic Cues. IEEE Trans. on Pattern Analysis and Machine Intelligence 27(5), 715–726 (2005)
9. Wang, L., Liu, X., Xia, L., Xu, G., Bruckstein, A.: Image orientation detection with integrated human perception cues (or which way is up). In: Proc. of the International Conference on Image Processing, ICIP 2003, vol. 3, pp. 539–542 (2003)
10. Zhang, L., Li, M., Zhang, H.-J.: Boosting image orientation detection with indoor vs. outdoor classification. In: Proc. of the 6th IEEE Workshop on Applications of Computer Vision (WACV 2002), pp. 95–99 (2002)
11. Tolstaya, E.: Content-based image orientation recognition. In: Proc. of the International Conference on Computer Graphics and Vision, GraphiCon 2007, pp. 158–161 (2007)
12. Frejlichowski, D.: An Experimental Comparison of Seven Shape Descriptors in the General Shape Analysis Problem. In: Campilho, A., Kamel, M. (eds.) ICIAR 2010. LNCS, vol. 6111, pp. 294–305. Springer, Heidelberg (2010)
13. Frejlichowski, D.: Pre-processing, Extraction and Recognition of Binary Erythrocyte Shapes for Computer-Assisted Diagnosis Based on MGG Images. In: Bolc, L., Tadeusiewicz, R., Chmielewski, L.J., Wojciechowski, K. (eds.) ICCVG 2010, Part I. LNCS, vol. 6374, pp. 368–375. Springer, Heidelberg (2010)
14. Frejlichowski, D.: Shape representation using Point Distance Histogram. Polish Journal of Environmental Studies 16(4A), 90–93 (2007)
15. Borawski, M.: Vector Calculus in Image Processing. Szczecin University of Technology Press (2007)

Multi-label Image Annotation Based on Neighbor Pair Correlation Chain

Guang Jiang[1,2], Xi Liu[3], and Zhongzhi Shi[1,⋆]

[1] The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[2] Graduate University of Chinese Academy of Sciences, Beijing 100049, China
[3] Information Technology Laboratory Fujitsu Research & Development Center Co., LTD., Beijing 100025, China

Abstract. Image annotation plays an important role in content-based image understanding, various machine learning methods have been proposed to solve this problem. In this paper, label correlation is considered as an undirected bipartite graph, in which each label are correlated by some common hidden topics. As a result, given a label, random walk with restart on the graph supplies a most related label, repeating this precedure leads to a label chain, which keep each adjacent labels pair correlated as maximally as possible. We coordinate the labels chain with its respective classifier training on bottom feature, and guide a classifier chain to annotate an image. The experiment illustrates that our method outperform both the baseline and another popular method.

Keywords: Image annotation, Image understanding, ensemble learning, correlation analysis, Image classification.

1 Introduction

Image content understanding is the foundation of many related applications(object recognition, cross-media retrieval etc.). How to mine the semantic meaning of an image is critical. However, "semantic gap" exits between low features and high semantic understanding of images, and brings a big puzzle. Image annotation aims to solve this problem by recognizing objects appears in the image. The goal is to allow one to retrieve, index, organize and understand large collections of image.

As a ensemble learning, multi-label classification has many advantages for image attention (utilizing different classifiers, avoiding over-fitting etc.). Differs from traditional supervised learning that a single instance is assumed to be associated with a single label. Image annotation is a multi-label problem, in which an instance is often related to several labels. Image annotation is such the case, each object appears in the image supplies a label. This problem can be

⋆ Supported by the National Natural Science Foundation of China(No. 61035003, 60933004, 60903141, 61072085), National Basic Research Program of China (No.2007CB311004).

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 345–354, 2012.

considered as multi-class problem, therefore is usually transformed into several binary-classification problems.

In an early stage, Boutell etc.[3] construct independent classifier for each scene to give an image different scene labels, which is also called binary relevance method (BR), which transform multi-label problem into several binary problems for each label. However, this method assume all the labels are independent, this is obviously unreasonable. Another method[15] is proposed to introduce dependence among labels, which named label combination. The basic idea is to combine a subset of labels into a new label and train the classifier respectively. However, this method is puzzled by the "combination explosion". Godbole etc.[7] improve BR method by training support vector machines (SVM) twice, the second training would involve classification results of first one into the kernel function. Recently, Read ect.[14] proposed a ensemble framework for classifier chain(ECC) used in multi-label classification. In this method, a randomized label sequence is created, moreover, all the precedent ones before the label are considered as features when its classifier is being trained. Several chain classifiers are ensembled to make up for the deficiency of randomized chain. However, the randomized label sequence make use of relevance among different labels partly.

There are two approaches to model semantic relation among different labels: semantic dictionary(WordNet etc.)[10] and data-driven solution[1]. The former supply relation collected by labor, may get out of the application background. On the contrary, the later usually depend on correlation computation models(correlation analysis etc.) to supply good result which is more closer to the problem at the price of large computation.

In this paper, we use a generative graphic model to compute the correlation among labels, and represented as a graph. Finally, random walk with restart is used to create a label chain in which adjacent labels pairs keep semantic close enough. The rest of the paper is organized as follows. Section 2 describes how to compute label correlation using topic model. Section 3 show a modified classifier chain algorithm for image annotation. Experimental results are shown in Section 4. Finally, we conclude in Section 5.

2 Model Label Correlation under a Graph

Recently, researchers often draw support from the Web to define similarities or correlation for label pair as follows([16], [11]): retrieval each label on the Web, and compare respective image result set according to a given measurement. This method is very convenient, but also hampered by the noise images. However, we argue that label correlations are implying in the dataset itself, and could be utilized by an appropriate method. Our work is related to wang[16], which define a Markov process on annotations to re-rank image annotation results from existed method. In contrast, we consider an approach for image annotation and involve label correlation simultaneously.

Latent Dirichlet Allocation(LDA) is a generative probabilistic model which allows explain data by unobserved groups (named "topic") that why they are

similar. It is presented as a graphic model[2] to discover hidden thematic structure from large archives of documents. In this model, the document is modeled as a distribution of underlying topics, while each topic is modeled as a distribution of all the words.

LDA has been used to image annotation, scene understanding etc. Different meanings of topic are considered in different missions: "outdoor" or "indoor" for scene understanding, while "tiger" or "forest" for object category. The de Finetti's theorem[5] ensure the joint probabilities of a sequence of words and topics have the form:

$$p(\boldsymbol{w}, \boldsymbol{z}) = \int p(\theta)(\prod_{n=1}^{N} p(w_n|z))p(z|\theta)d\theta \tag{1}$$

In which, $p(z|\theta)$ is regarded as prior distribution, whereas $p(w_n|z)$ as likelihood, which describe the distribution of all words under topic z. Learning these various distribution is a problem of Bayesian Inference, and accurate inference is infeasible usually, hence approximate inference methods (Gibbs sampler, mean-field variational methods etc.) are used to train the model under EM framework.

We define label correlation from their co-occurance in different images, and consider "tiger" is related to "forest" because they often appear in a same image. However, LDA convert each image into a vector in a semantic space: "topic space" instead of bag of words(BOW); and would supply us finer-grained correlation of words(labels accordingly in this paper) by topics: $p(w|z_k)$.

On the other hand, it can be found that when two labels relate more close, only they co-occur in some topics with higher probabilities. For example, "bear" and "ice" often appears in topic "outdoor", also topic "cold", but rarely "indoor". Therefore, label co-occurance could illustrated in the elaborately defined topic, not in raw images.

Inspired by this fact, we regard an image as a vector of its labels, then use any topic model, we train the model and get a distribution $p(w|z_k)$ of label p for each topic z_k, then we construct a undirected bipartite graph to represent label correlation as in Fig. 1. In this graph, there are two kinds of vertexes: V_w, which are set of labels with size $|W| = n$, and V_z, set of topics with size $|Z| = K$ respectively. The weight of each edge $e(w_i, z_k)$ for $1 \leq i \leq n, 1 \leq k \leq K$ is defined as:

$$e(w_i, z_k) = p(w_i|z_k) \tag{2}$$

In which $p(w_i|z_k)$ is the conditional probability appeared in equation (1). Based on Eqation (2), adjacent matrix can be defined. The graph illustrates correlation relationships of all labels.

It is thus evident that we can formalize our problem into how to measure the closeness of arbitrary label pair (w_i, w_j) in V_w, owing to the undirected and only connectivity between V_z and V_w, this procedure can be considered as accumulating all messages from w_i through all z and w to w_j. However, few

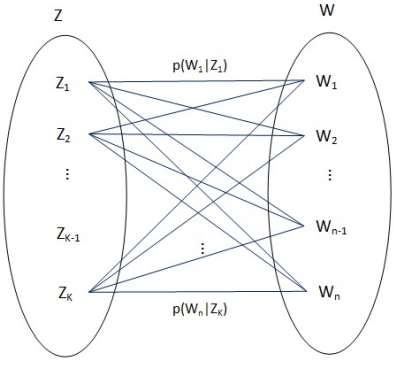


Fig. 1. Label correlation via hidden topics

high frequent labels under each topic lead to the fact that the adjacent matrix is sparse usually. More worse, it cannot reflect the global structure of the graph [9]. Therefore, we introduce random walk with restart to solve this problem.

3 Multi-label Image Annotation Based on Neighbor Pair Correlation

Random Walk with Restart(RWR) supply a excellent technique to define relevance score between arbitrary two nodes in a graph. It is defined as (3)[12]. The basic procedure assumes a random walker start from node w_i of a graph, in the next time-tick, the walker has two choices: follow an edge to get an adjacent node, or jump to another node w_j with a probability $c * \overrightarrow{e_i}$. In which, $\overrightarrow{e_i}$ is the restart vector, and c is the probability of restarting the walk.

RWR represent a node i as the vector e_i, and compute a rank vector over all nodes in the graph using the following equation:

$$\overrightarrow{r_i} = c\widetilde{\boldsymbol{W}}\overrightarrow{r_i} + (1-c)\overrightarrow{e_i} \tag{3}$$

After simple derivation, we can get:

$$\overrightarrow{r_i} = (1-c)(I - c\widetilde{\boldsymbol{W}})^{-1}\overrightarrow{e_i} \tag{4}$$

RWR requires a matrix inversion. There exists two solutions: The first one is to pre-compute, which pre-compute and store the low-rank approximation of a sparse matrix in equation (5); another is to compute matrix inversion on the fly: given a initial value, then equation (4) is iterated until convergence. When the graph is not large, the both work efficiently, we adopt the first and implement it.

For any word w_i in V_w in Fig. 1, we can define its association with another node w_j as follows:

$$asso(w_i, w_j) = r_{ij} \tag{5}$$

In which, r_{ij} is the jth element of $\overrightarrow{r_{ij}}$ in (4), representing the relevance score of node j is the steady-state probability of random walk from node i to node j. We can found $asso(w_i, w_j) \neq asso(w_j, w_i)$, so equation in (5) is not symmetric, but also conform to the fact, for instance, "elephant" appears in an image, which associate "zoo" (or "forest") more possibly, while the opposite is weakly somehow.

We'll recall the method concisely proposed in [14] as mentioned in section 1. The method proposed classifier chain of multi-label for image annotation by involving label correlation as follows: firstly, a randomized label chain is created, when classifier for a label is training, classifying results of its precedent labels in the chain are involved. For a new image, according to the label chain, each classifier for a label w is employed, in the meanwhile, precedent labels in the chain are considered, to predicate whether w is related to the image. The training process can be described in step 1 to step 6 in Fig. 2.

Given a graph, relevance with all the other nodes for an initial node w can be obtained with a score using RWR, most relevant node with w can be obtained after ranking. Inspired by the fact, we can create label chain as follows: given an initial word w_1, select $w_2 = \arg\max_2^n asso(w_1, w_j)$, likewise, w_3 is selected according to w_2 as an initial node, repeated this procedure induce a label chain "$w_1 \rightarrow w_2 \rightarrow \ldots \rightarrow w_n$". The label chain assure that adjoining labels are as semantic close as possible. Based on this, we proposed a new label chain classifier, and expect precedent label in the label chain can bring more positive information to the subsequent labels for annotating an image. The algorithm is described in Fig. 2.

In Fig. 2, step 4 compose both visual features and category features into a new feature vector. However, these two kinds of features are heterogeneous, and also dimense differ. Followed [7], scale factor f is introduced into the composition: x is scaled to $[0, f]$, while $(y_{w'_1}, y_{w'_2}, \ldots, y_{w'_{i-1}})$ is scaled to $[1-f, 1]$. In step 7, For label w_{i-1}, it is represented as restart vector, in which $i-1$th is 1, all other elements are 0. Using equation (4), descending $\overrightarrow{r}_i$ illustrate the most relevant label w_i.

4 Experiments

We use Corel5k[6] and IAPR TC-12[8] dataset to validate our methods. Corel5k dataset contains 5000 images and 374 labels in total, each image has 4-5 labels. IAPR dataset is more challenging, which contains 20,000 images, and having varying appearance, and keywords extracted from free-flowing text captions. We picked a subset of 25 labels (Fig. 3 and Fig. 4) for both datasets in our experiment, and get about 4500 images for corel5k and 10000 images for IAPR respectively. Images are qualified as follows: each image is segmented as 18*18 pixels block, for each block, hsv feature of 36 dimensions for color and LBP[17] feature of 51 dimensions for texture is extracted. Two kinds of features are both clustered into 500 blobs and make up the visual dictionary. Therefore, each image is represented by a vector of 1000 visual words.

We considered each image as a BOW of its labels, and use GibbsLDA++ [13] to train a LDA model with 100 topics and get label distribution for each topic,

Algorithm: Neighbor Pair Correlation Chain Classifier (NPC)

Input: Training dataset $D = \{(x_1, Y_1), (x_2, Y_2), \ldots, (x_n, Y_n)\}$, Initial label w'_1

Output: Label chain $w'_1, w'_2, \ldots, w'_{|L|}$, and according classifiers chain $\{C_{w'_1}, C_{w'_2}, \ldots, C_{w'_{|L|}}\}$

1. **for** $i = 2$ to $|L|$ **do**
2. $D' \leftarrow \{\}$
3. **for** $(x, Y) \in D$ **do**
4. $D' \leftarrow D' \cup ((x, y_{w'_1}, y_{w'_2}, \ldots, y_{w'_{i-1}}), y_{w'_i})$
5. **end for**
6. Train classifier $C_{w'_{i-1}}$
7. Compute most correlative label of w_{i-1} using RWR as w_i, which not appeared in current label chain $w_1, w_2, ..., w_{i-1}$
8. **end for**

Fig. 2. Algorithm NPC

mountain sky water tree beach boats people birds buildings
bear snow ice forest field flowers rocks plants sand house
cat tiger street stone frost desert

Fig. 3. Selected label set in Corel5k

duplicate topics are deleted because there exists two or more topics have same label distribution. We construct a graph as defined in section 2. We select the most "discriminative" label, which its meta-classifier has highest precision in BR, as the head of label chain, and implement random walk with restart algorithm ($c = 0.8$) to create a label chain. LibSVM [4] with RBF kernel is employed to train classifier for each label. The experiment is running on Win 7 platform.

Single classifier chain only envolved partial correlations among labels, therefore strategy of ensembling is considered in [14]. In our experiment, for each method we train 9 classifier chains as the ensembler to vote for each label, and use 10-fold cross validation to test our algorithm. We use two evaluation criterions which usually used in information retrieval: $microF_1$ and $macroF_1$ as follows:

$$F1_{micro} = \frac{2 * \frac{\sum_{i=1}^{Q} N_{i-true-pos}}{\sum_{i=1}^{Q} N_{i-pos}} * \frac{\sum_{i=1}^{Q} N_{i-true-pos}}{\sum_{i=1}^{Q} N_{i-true}}}{\frac{\sum_{i=1}^{Q} N_{i-true-pos}}{\sum_{i=1}^{Q} N_{i-pos}} + \frac{\sum_{i=1}^{Q} N_{i-true-pos}}{\sum_{i=1}^{Q} N_{i-true}}} \tag{6}$$

base bench bike bottle canyon condor cycling fence flagpole
garden girl harbour house jungle meadow pool river roof
ship shore stair statue tourist wall window

Fig. 4. Selected label set in IAPR

$$p_i = \frac{N_{i-true-pos}}{N_{i-pos}},\ r_i = \frac{N_{i-true-pos}}{N_{i-true}} \tag{7}$$

$$F1_{macro} = \frac{1}{|W|}\sum_{i=1}^{|W|}\frac{2p_i r_i}{p_i + r_i} \tag{8}$$

In which, $|W|$ is the number of annotation words, N_{i-true} is the number of images annotated with w_i in groundtruth, N_{i-pos} is the number of images annotated with w predicted by the method, and $N_{i-true-pos}$ is the number of images in groundtruth and both predicated by the method. Therefore, p_i and r_i is accuracy and recall of label w_i respectively.

Effect of Scale factor is shown in Fig. 5. Scale factor f is the coefficient to balance the visual feature with category feature. Bigger f, visual features effect the classifier more. As shown, BR keep stable because it doesn't introduce any category feature. When f is lower than about 0.4, both ECC and NPC work poor than BR, due to the lower-level visual features of image contribute less to the chain classifier, yet few category features are not discriminant enough to discern different images. As f higher, especially during 0.4 to 0.7, both ECC and NPC work better than BR, the most possible reason attributes to the fact that f get a well balance among visual features and semantic label features best. Therefore, most appropriate f is selected and the performance is illustrated in the Fig. 7. We can found that performance of ECC has fluctuates in a flat, due to the fact that category feature is picked randomly. We can see that, as f higher, the semantic label features play a relatively minor role, thus ECC and NPC get close to BR. Our method perform better than other methods overall, mostly because more label correlations are introduced.

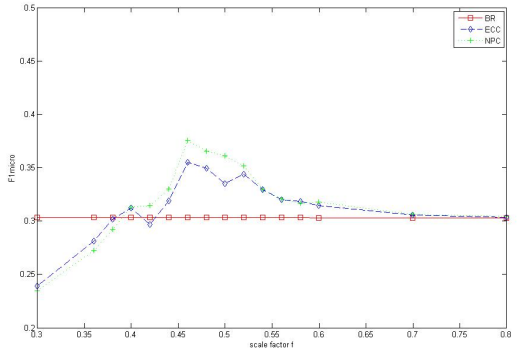


Fig. 5. $F1_{micro}$ performance with scale factor varying in corel5k

Fig. 6 gives a label chain obtained form our algorithm. We can found that RWR help us a lot, adjacent pair such as "snow→frost" can be mined. Even long

chain are linked together because of their semantic correlation, such as "sand→ rock→ mountain" and "tree→ tiger→ water→ plant". RWR is computed with probability, so there also exists several obscure sub-chains.

flower→ boat→ bird→ bear→ beach→ stone→ snow→ frost→ desert→ building→ sky→ ice→ cat→ people→ forest→ field→ house→ sand→ rock→ mountain→ street→ tree→ tiger→ water→ plant

Fig. 6. A label chain using RWR in Corel5k

Fig. 7 illustrate performance on the two dataset, our algorithm NPC outperform both in $F1_{macro}$ and $F1_{micro}$. We think that the fact is attributed to capacity of the meta-classifier which displayed in Fig. 8, in most cases, meta-classifier of NPC has higher precision and therefore benefit multi-label image annotations.

Dataset	Performance	BR	ECC	NPC
Corel5k	$F1_{macro}$	0.274	0.286	0.309
	$F1_{micro}$	0.303	0.335	0.346
IAPR	$F1_{macro}$	0.212	0.216	0.221
	$F1_{micro}$	0.245	0.263	0.279

Fig. 7. Performance of NPC compared with other methods

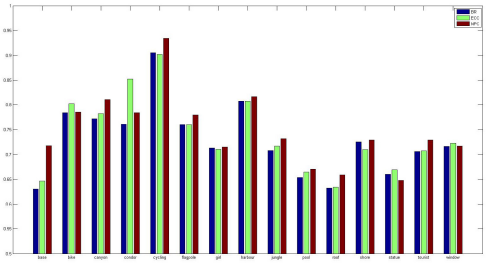


Fig. 8. 10-fold cross-validation precision of meta-classifier for each label in IAPR

5 Conclusions

In this paper, we give an approach to mining label correlation from the semantic space, and augment relationship among them using random walk with restart.

Our experiment illustrates that our method can conduct classifier chain more effectively for image multi-label annotation due to combination of bottom features and its semantic label features.

In the future work, we would like to try to introduce more corpus to mine semantic relations among annotation words. Furthermore, correlation information could be utilized into two related missions: annotation an image when little labels are given (label recommendation), and filtering noise label when too much labels are given(label filter).

References

1. Blei, D.M., Jordan, M.I.: Modeling annotated data. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, SIGIR 2003, pp. 127–134. ACM, New York (2003)
2. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)
3. Boutell, M., Luo, J., Shen, X., Brown, C.: Learning multi-label scene classification. Pattern Recognition 37(9), 1757–1771 (2004)
4. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011)
5. De Finetti, B.: Theory of Probability: A critical introductory treatment, vol. 2. Wiley (1990)
6. Duygulu, P., Barnard, K., de Freitas, J.F.G., Forsyth, D.: Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 97–112. Springer, Heidelberg (2002)
7. Godbole, S., Sarawagi, S.: Discriminative methods for multi-labeled classification. In: Advances in Knowledge Discovery and Data Mining, pp. 22–30 (2004)
8. Grubinger, M., Clough, P., Müller, H., Deselaers, T.: The iapr tc-12 benchmark-a new evaluation resource for visual information systems. In: International Workshop OntoImage, pp. 13–23 (2006)
9. He, J., Li, M., Zhang, H., Tong, H., Zhang, C.: Manifold-ranking based image retrieval. In: Proceedings of the 12th Annual ACM International Conference on Multimedia, pp. 9–16. ACM (2004)
10. Jin, Y., Khan, L., Wang, L., Awad, M.: Image annotations by combining multiple evidence & wordnet. In: Proceedings of the 13th Annual ACM International Conference on Multimedia, MULTIMEDIA, pp. 706–715. ACM, New York (2005)
11. Liu, X., Shi, Z., Li, Z., Wang, X., Shi, Z.: Sorted label classifier chains for learning images with multi-label. In: Proceedings of the International Conference on Multimedia, pp. 951–954. ACM (2010)
12. Pan, J., Yang, H., Faloutsos, C., Duygulu, P.: Automatic multimedia cross-modal correlation discovery. In: Proceedings of the Tenth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining, pp. 653–658. ACM (2004)
13. Phan, X.H., Nguyen, C.T.: Gibbslda++: A c/c++ implementation of latent dirichlet allocation, lda (2007)
14. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier Chains for Multi-label Classification. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009. LNCS, vol. 5782, pp. 254–269. Springer, Heidelberg (2009)

15. Tsoumakas, G., Vlahavas, I.: Random *k*-Labelsets: An Ensemble Method for Multilabel Classification. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 406–417. Springer, Heidelberg (2007)
16. Wang, C., Jing, F., Zhang, L., Zhang, H.J.: Content-based image annotation refinement. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 1–8. IEEE (2007)
17. Zhi-ping, S., Hong, H., Qing-yong, L., Zhong-zhi, S., Chan-lun, D.: Texture spectrum descriptor based image retrieval. The Journal of Software 16(6), 1039–1045 (2005)

Enhancing Image Retrieval by an Exploration-Exploitation Approach

Luca Piras[1], Giorgio Giacinto[1], and Roberto Paredes[2]

[1] Department of Electrical and Electronic Engineering University of Cagliari
Piazza D'armi, 09123 Cagliari, Italy
{luca.piras,giacinto}@diee.unica.it
[2] Universidad Politécnica de Valencia
Camino de Vera s/n, Edif. 8G Acc. B, 46022 Valencia, Spain
rparedes@dsic.upv.es

Abstract. In this paper, the Relevance Feedback procedure for Content Based Image Retrieval is considered as an Exploration-Exploitation approach. The proposed method exploits the information obtained from the relevance score as computed by a Nearest Neighbor approach in the *exploitation* step. The idea behind the Nearest Neighbor relevance feedback is to retrieve the immediate neighborhood of the area of the feature space where relevant images are found. The exploitation step aims at returning to the user the maximum number of relevant images in a local region of the feature space. On the other hand, the *exploration* step aims at driving the search towards different areas of the feature space in order to discover not only relevant images but also informative images. Similar ideas have been proposed with Support Vector Machines, where the choice of the informative images has been driven by the closeness to the decision boundary. Here, we propose a rather simple method to explore the representation space in order to present to the user a wider variety of images. Reported results show that the proposed technique allows to improve the performance in terms of average precision and that the improvements are higher if compared to techniques that use an SVM approach.

Keywords: Algorithms, Active Learning, max-min.

1 Introduction

Nowadays, the possibility for people to easily create, store, and share, vast amount of multimedia documents is a problem that the pattern recognition community is facing since several years. Digital cameras allow capturing an unlimited number of photos and videos, thanks to the fact that they are also embedded in a number of portable devices. This vast amount of media archives needs to be organized in order to ease future search tasks. It is easy to see that it is often impractical to automatically label the content of each image or different portions of videos by recognizing the objects in the scene, as we should have templates for each object in different positions, lighting conditions, occlusions, etc. [26].

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 355–365, 2012.

It is quite easy to see that each picture and video may be characterized by a large number of concepts depending on the level of detail used to describe the scene, or the focus in the description. Moreover, different users may describe an image using different categories, and the same user may classify the same image in different ways depending on the context. Sometimes, an image may contain one or more concepts that can be prevalent with respect to others, so that if a large number of people are asked to label the image, they may unanimously use the same label. Nevertheless, it is worth noting that a concept may be also decomposed in a number of "elementar" concepts. For example, an image of a car can have additional concepts, like the color of the car, the presence of humans or objects, etc. Thus, for a given image or video-shot, the same user may focus on different aspects. How the task of retrieving similar images or videos from an archive can be solved by automatic procedures? How can we design procedures that automatically tune the similarity measure to adapt to the visual concept the user is looking for? Once again, the target of the classification problem cannot be clearly defined beforehand but must be designed to explicitly take into account user needs [10].

In the field of content based multimedia retrieval, a number of review papers pointed out the difficulties in providing effective similarity measure that can cope with the broad domain of content of multimedia archives [6,21,28]. The shortcomings of current techniques developed for image and video has been clearly shown by Pavlidis [25]. While systems tailored to a particular image domain (e.g., medical images) can exhibit quite impressive performances, the use of these systems on unconstrained domains reveals their inability to adapt dynamically to new concepts [27]. One solution is to have the user manually label a small set of representative images as relevant or non relevant to the query (the so-called relevance feedback) that are used as training set for updating the similarity measure [33]. In this system is not uncommon that, after the first feedback iterations, the number of relevant images retrieved increases quickly. However, the system typically stops providing new relevant images despite of the user interaction. The reason lies in the way in which images are presented to the user. In fact, usually the best ranked images are retrieved after each round of feedback, and these images are usually retrieved in a small local area of the feature space. As a consequence, the search often converges towards a local optimum, without taking into account images in other areas of the feature space. In order to address this kind of problems, we propose an *Exploitation-Exploration* mechanism where the exploration step is inspired by *Active Learning* [4]. Our approach requires the system to choose not only the most relevant images according to the user judgement, but also the most informative images that allows driving the search in more promising regions of the feature space. The key issue is how to choose the most informative images. Usually this approach has been used in systems based on discriminative functions, i.e. system that builds a decision function which classifies the unlabelled data.

One method to select informative images is based on choosing the patterns closest to the decision boundary, as described in [31,16] where SVM based on

active learning are used. In [3], the authors proposed to learn two SVM classifiers in two uncorrelated feature spaces as color and texture. The classifiers have been then used to classify the images and the unlabelled ones that received a different label in the two feature spaces, have been chosen to be shown to the user. In addition, different criteria have been proposed over the years as the minimization of expected average precision [13], or the maximization of the entropy [19]. In the latter paper the authors learned an SVM on the labelled images, mapped the SVM outputs into probabilities and chose the images with the probability to belong to relevant class nearest to 0.5. In [17] the authors instead to use the proximity to the theoretical decision boundary as measure of the information capability of the training images, propose a clarity index that takes into account the rank of each image with respect to those of the known relevant and non relevant images. The images with the lowest values of clarity are chosen as training images.

Conventional SVM active learning is designed to select a single example for each learning iteration but, as suggested in [15], usually in a relevance feedback iteration the user labels multiple image examples as being relevant or non relevant. In this case it is possible that the system selects similar images to learn the SVM. The authors, to address this problem, proposed a *Batch Mode Active Learning* technique that chooses the most suitable unlabelled examples one at a time. An interesting approach has been proposed in [34] where the authors propose a novel paradigm of active learning, which is able to estimate the probability density function (pdf) of the underlying query distribution to avoid the risk of learning on a completely unknown distribution. The estimated pdf, together with the distribution of the classifier outcomes, is used to guide the sampling, in the way that it is possible to give priority to two types of instances to label in the next iteration, namely instances in the area where the probability to find relevant pattern is high (for boosting the retrieval) and instances in the uncertain area (for figuring out the new decision boundary). In [22] the authors, instead of using SVM, proposed a selective sampling for Nearest Neighbor classifiers. In order to choose the most informative patterns they suggest to consider not only the uncertainty of the candidate sample point, but also the effect of its classification on the remaining unlabelled points. For this reason, their *lookahead algorithm for selective sampling* considers sampling sequences of neighboring patterns of length k, and selects an example that leads to the best sequence. The best sequence is the one whose samples have the highest conditional class probabilities. Also in [18] the authors proposed a probabilistic variant of the k-Nearest Neighbor method for active learning in multi-class scenarios. After that they defined a probability measure, based on the pairwise distances between data points, they used the Shannon entropy as "uncertain" measure over the class labels in order to maximize the discriminating capabilities of the model.

In this paper we consider the most informative images as those that are distributed around the images that have been labelled by the user along all the representation space. This task can be accomplished by resorting to an Active

Learning approach based on hierarchical clustering of the data [5]. Although this technique allows obtaining good results, it is quite computationally expensive. In order to use a computationally cheap method, we performed the exploration phase through a *max-min* approach that showed good results in similar tasks such as the initialization of the *c-means* algorithm [20].

This paper is organized as follows. Section 2 briefly reviews the Nearest Neighbor approach used in order to assign a Relevance Score to the images. Section 3 introduces the proposed technique and describes how exploit it in the Relevance Feedback iterations. Experimental results are reported in Section 4. Conclusions are drawn in Section 5.

2 Nearest-Neighbor Relevance Feedback for Relevance Score

The use of the Nearest-Neighbor paradigm has been inspired by classification techniques based on the "nearest case", which are used in pattern recognition and machine learning for classification and outlier detection. In addition, nearest-neighbor techniques have been also used in the context of "active learning", which is closely related to relevance feedback [22]. In particular, recent works on outlier detection and one-class classification clearly pointed out the effectiveness of nearest-neighbor approaches to identify objects belonging to the target class (i.e., the relevant images), while rejecting all other objects (i.e., non relevant images) [2,30]. This approach is suited to cases when it is difficult to produce a high-level generalization of a "class" of objects.

This approach can be used for estimating image relevance in CBIR as each "relevant" image as well as each "non relevant" image can be considered as individual "cases" or "instances" against which the images of the database should be compared [12].

In this paper, a method proposed in [9] has been used, where a score is assigned to each image of a database according to its distance from the nearest image belonging to the target class, and the distance from the nearest image belonging to a different class. This score is further combined to a score related to the distance of the image from the region of relevant images. The combined score is computed as follows:

$$rel(\mathbf{x}) = \left(\frac{n/t}{1+n/t}\right) \cdot rel_{BQS}(\mathbf{x}) + \left(\frac{1}{1+n/t}\right) \cdot rel_{NN}(\mathbf{x}) \tag{1}$$

where n and t are the number of non-relevant images and the whole number of images retrieved after the last iteration, respectively. The two terms rel_{NN} and rel_{BQS} are computed as follows:

$$rel_{NN}(\mathbf{x}) = \frac{\|\mathbf{x} - NN^{nr}(\mathbf{x})\|}{\|\mathbf{x} - NN^{r}(\mathbf{x})\| + \|\mathbf{x} - NN^{nr}(\mathbf{x})\|} \tag{2}$$

where $NN^r(\mathbf{x})$ and $NN^{nr}(\mathbf{x})$ denote the relevant and the non relevant Nearest Neighbor of $\mathbf{x}$, respectively, and $\|\cdot\|$ is the metric defined in the feature space at hand,

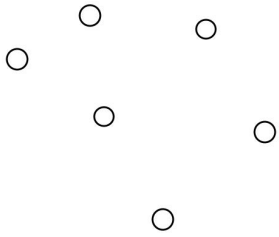
$$rel_{BQS}(\mathbf{x}) = \frac{1 - e^{1 - d_{BQS}(\mathbf{x}) \Big/ \max\limits_{i} d_{BQS}(\mathbf{x}_i)}}{1 - e} \tag{3}$$

where e is the *Euler's number*, i is the index of all images in the database and d_{BQS} is the distance of image $\mathbf{x}$ from a reference vector computed according to the Bayes decision theory (Bayes Query Shifting, BQS) [11].

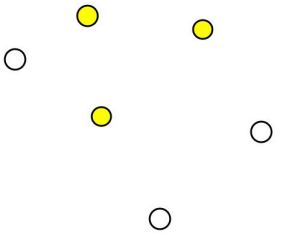
3 The Exploration Phase

Typically, an image retrieval system with relevance feedback works as follows: after the submission of a query, the system, according to a similarity measure, scores all images in the database and presents the k best scored ones to the user. One of the problems in this kind of behavior is that in the following iterations the search could be driven by the (probably few) relevant images retrieved so far, and the system could be trapped in a limited area of the feature space. Sometimes, neither the non relevant images, that are also considered in the evaluation of the relevance score, can help to get out of this situation. In fact, by always using the same set of relevant images iteration by iteration, the search can "take a wrong way". In order to face this problem, the proposed method selects within a certain number of best scored images, those images that are "not too close" to the "classical search area". The meaning of "not too close" and "classical search area" will be explained in the following.

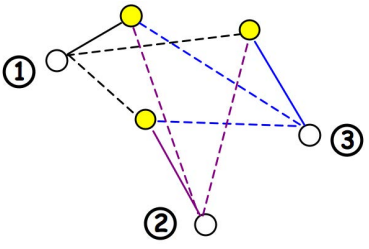
Let us define k as the number of the images to return to the user and $\lfloor k \cdot \alpha \rfloor$ as the fixed number of the "seed images". Let us also define $\lfloor (k - \lfloor k \cdot \alpha \rfloor) \cdot \beta \rfloor$ as the number of images among which to choose the most informative ones. In the previous formulas, the parameter α can assume values between $\frac{1}{k}$ and 1, and $\beta \geq 1$. Summing up, with the *Exploration-Exploitation* approach k images are shown to the user, the best scored $\lfloor k \cdot \alpha \rfloor$ are selected beforehand, while the other $(k - \lfloor k \cdot \alpha \rfloor)$ are chosen through a *max-min* approach between the $\lfloor (k - \lfloor k \cdot \alpha \rfloor) \cdot \beta \rfloor$ best scored images. It is clear that if $\alpha = \frac{1}{k}$ all the images, apart from the query, are selected in an "active" way, on the contrary, when it is equal to 1, they system shows to the user the best k scored images as in the classical Nearest Neighbor approach. The same happens when $\beta = 1$, as $(k - \lfloor k \cdot \alpha \rfloor)$ images will be chosen from a set of $(k - \lfloor k \cdot \alpha \rfloor)$ best scored images. The *max-min* approach selects an image from a set evaluating all the distances between the seed images and the images in the set and choosing for each of them the shortest. The images are then sorted according to these distances and that with the maximum distance it is selected. The idea behind the *max-min* technique is similar to that of pruning, that is, it pays attention to the relevant images that are most far apart from each other because they are more likely to be the most different from those that are usually used [20]. To better explain the algorithm, Figure 1 shows an example where $k = 5$, $\alpha = 0.6$, and $\beta = 1.5$.



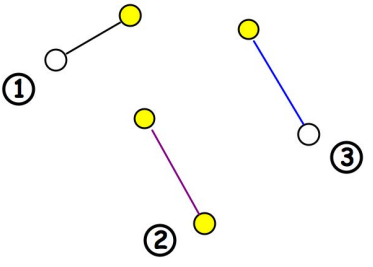
(a) The set of considered images



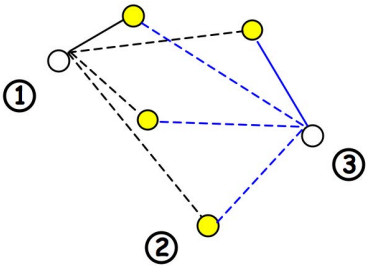
(b) $\lfloor k \cdot \alpha \rfloor$ best scored images



(c) Evaluation of all the distances



(d) *Max-min* distance



(e) New round of selection

Fig. 1. Exploration-Exploitation algorithm

(a) For each image in the database a score is computed according to Eq. (1);
(b) the three best scored images are used as seeds of the search ($\lfloor 5 \cdot 0.6 \rfloor = 3$);
(c) for each of the remaining images ($\lfloor (5 - \lfloor 5 \cdot 0.6 \rfloor) \cdot 1.5 \rfloor = 3$) the distances with the seed images are evaluated, and the minimum ones are chosen;
(d) the image with the largest minimum distance (image **(2)**) is chosen to be added to the seed images.
(e) in order to add the fifth image to be shown to the user ($k = 5$), the algorithm restarts from step **(c)**.

It is clear that if $\alpha = \frac{1}{k}$, all the images, apart from the query, are selected following the *Exploration* approach. On the other hand, when $\alpha = 1$ the best k scored images, as in the classical Nearest Neighbor approach, are shown to

the user. The same happens when $\beta = 1$, in fact in this situation $(k - \lfloor k \cdot \alpha \rfloor)$ images from a set of $(k - \lfloor k \cdot \alpha \rfloor)$ best scored images are chosen.

4 Experimental Results

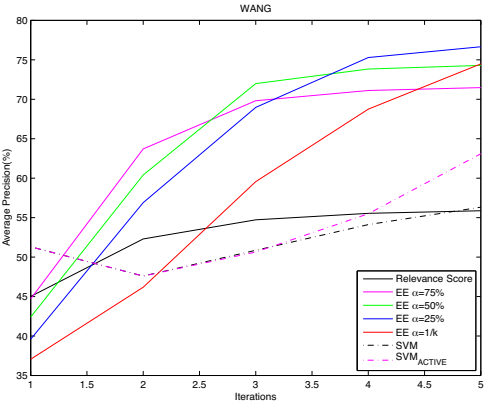
4.1 Datasets

Experiments have been carried out using three datasets, namely the Caltech-256 dataset, from the California Institute of Technology[1], the WANG dataset[2], and the Microsoft Research Cambridge Object Recognition Image Database[3] (in the following referred to as MSRC). The first dataset consists of 30607 images subdivided into 257 semantic classes [14], the WANG dataset consists of a subset of 1000 images of the Corel stock photo database which have been manually selected and which form 10 classes of 100 images each [32], and MSRC contains 4320 images subdivided into 17 "main" classes, each of which is further subdivided into subclasses, for a total of 33 semantic classes [35]. From Caltech-256, the *Edge Histogram* descriptor [1] has been extracted using the open source library LIRE (Lucene Image REtrieval) [24]. The images from the WANG dataset are represented by a 512-dimensional colour histogram and a 512-dimensional Tamura texture feature histogram [29] concatenated in a single vector [7]. The images of MSRC are represented by a vector of 4096 components of SIFT descriptors [23] extracted at Harris interest points [8,7]. The WANG, MSRC and Caltech-256 datasets represents image retrieval tasks of different complexity. In fact, the WANG dataset is usually considered an easy task in the Image Retrieval context. The MSRC dataset is mainly used in the object recognition domain, as the pictures usually contain one object "captured" from a particular point of view (front, side, rear, etc.) or at most more objects of the same type. The Caltech-256 dataset is also widely used in object recognition, and it can be considered a more difficult task than the one represented in the MSRC dataset: the semantic concepts in the Caltech dataset are more loosely related to the image content. As example, the class "marriages" contains images of newlyweds as well as images of wedding cakes.

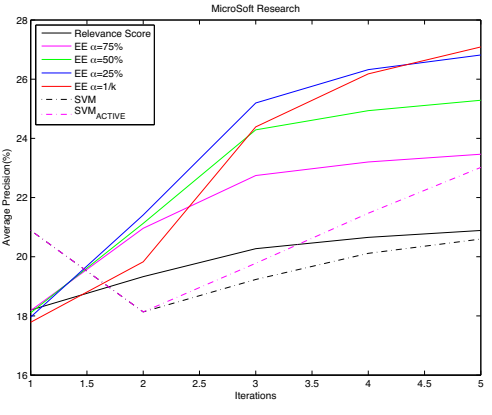
4.2 Experimental Setup

In order to test the performances of the proposed approaches, 500 query images from Caltech-256 dataset have been randomly extracted, so that they cover all the semantic classes. For the WANG and MSRC datasets, each image is used as query. Relevance feedback is performed by marking images belonging to the same class of the query as relevant, and all other images in the pool of k to-be-labelled images as non-relevant. Performance is evaluated in terms of mean average precision taking into account all the relevant images as they are ordered by the classifier.

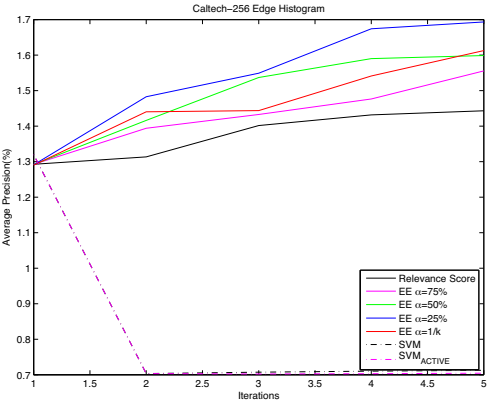
[1] http://www.vision.caltech.edu/Image_Datasets/Caltech256/
[2] http://wang.ist.psu.edu/docs/related.shtml
[3] http://research.microsoft.com/downloads



(a) WANG Dataset



(b) MSRC Dataset



(c) Caltech-256 Dataset

Fig. 2. Average Precision for 5 rounds of relevance feedback

In order to choose the most suitable values of the parameters α and β, a number of preliminary experiments have been performed. Accordingly, it has been fixed the value of β at 10, whereas different values of α have tested, and the related results are reported for comparison purposes. The relevance score at each image has been assigned according to Eq. (1), and in the graph it has been referred to as the *Relevance Score*, whereas the case $\alpha = \frac{1}{k}$ has been referred to as $\alpha = 0\%$. For comparison purposes, relevance feedback has been also computed by a "pure" SVM classifier with an RBF kernel and an SVM_{ACTIVE} as well. The Active Learning has been performed according to the approach proposed in [17].

4.3 Results

Figures 2(a), 2(b), and 2(c) show the average precision evaluated using the WANG, the MSRC, and the Caltech datasets, respectively. Observing the Figure 2 it is quite clear to see how the lines have a quite different trend w.r.t. the value of α. In fact, the lower the value of α in Figure 2, the better the measured average precision. The only exception in the above trend can be seen in the case of $\alpha = \frac{1}{k}$ (referred to as $\alpha = 0\%$) in Figure 2(a) and (c), where the average precision is lower than the values obtained with $\alpha = 0.25\%$. The reason of this behavior probably is due to the fact that only the query is not enough as "seed image" in order to begin the search. In the case of the WANG and MSRC dataset it is also easy to see how the lower improvement obtained using the Exploration-Exploitation approach ($\alpha = 0.75\%$) w.r.t. the method without "Exploration" phase, overcomes the improvement of the SVM_{ACTIVE} w.r.t the "pure" SVM.

5 Conclusion

In this paper the problem of low informative training sets has been faced exploiting the Nearest Neighbor paradigm in an Exploration-Exploitation task. The proposed algorithm subdivides the NN approach in two phases: *Exploitation* and *Exploration*. In the Exploitation phase a relevance score is assigned to each image in the database, while in the Exploration phase a certain number of best scored images is drawn, and by means of a max-min approach based on the distances among the images, the images that will be shown to the user to capture her feedback are chosen. The obtained results clearly show that the bigger the percentage of the images, proposed to the user, the larger is the obtained average precision. According to these results, it is possible to say that the proposed Exploration-Exploitation approach is able to move the search in areas of the feature space usually "unexplored".

References

1. Information technology - Multimedia content description interface - Part 3: Visual, ISO/IEC Std. 15938-3:2003 (2003)
2. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: Identifying density-based local outliers. In: Chen, W., Naughton, J.F., Bernstein, P.A. (eds.) SIGMOD Conference, pp. 93–104. ACM (2000)

3. Cheng, J., Wang, K.: Active learning for image retrieval with co-svm. Pattern Recognition 40(1), 330–334 (2007)
4. Cohn, D.A., Atlas, L.E., Ladner, R.E.: Improving generalization with active learning. Machine Learning 15(2), 201–221 (1994)
5. Dasgupta, S., Hsu, D.: Hierarchical sampling for active learning. In: Cohen, W.W., McCallum, A., Roweis, S.T. (eds.) ICML. ACM International Conference Proceeding Series, vol. 307, pp. 208–215. ACM (2008)
6. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys 40(2), 1–60 (2008)
7. Deselaers, T., Keysers, D., Ney, H.: Features for image retrieval: an experimental comparison. Inf. Retr. 11(2), 77–107 (2008)
8. Dorkó, G.: Selection of Discriminative Regions and Local Descriptors for Generic Object Class Recognition. Ph.D. thesis, Institut National Polytechnique de Grenoble (2006)
9. Giacinto, G.: A nearest-neighbor approach to relevance feedback in content based image retrieval. In: CIVR 2007: Proceedings of the 6th ACM International Conference on Image and Video Retrieval, pp. 456–463. ACM, New York (2007)
10. Giacinto, G.: Moving targets in computer security and multimedia retrieval. Trans. MLDM 4(1), 30–52 (2011)
11. Giacinto, G., Roli, F.: Bayesian relevance feedback for content-based image retrieval. Pattern Recognition 37(7), 1499–1508 (2004)
12. Giacinto, G., Roli, F.: Instance-based relevance feedback for image retrieval. In: Saul, L.K., Weiss, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems 17, pp. 489–496. MIT Press (2005)
13. Gosselin, P.H., Cord, M.: Active learning methods for interactive image retrieval. IEEE Transactions on Image Processing 17(7), 1200–1211 (2008)
14. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Tech. Rep. 7694, California Institute of Technology (2007), http://authors.library.caltech.edu/7694
15. Hoi, S.C.H., Jin, R., Zhu, J., Lyu, M.R.: Semisupervised svm batch mode active learning with applications to image retrieval. ACM Trans. Inf. Syst. 27(3), 16:1–16:29 (2009)
16. Hoi, S.C.H., Lyu, M.R.: A semi-supervised active learning framework for image retrieval. In: CVPR (2), pp. 302–309. IEEE Computer Society (2005)
17. Huang, T., Dagli, C., Rajaram, S., Chang, E., Mandel, M., Poliner, G., Ellis, D.: Active learning for interactive multimedia retrieval. Proceedings of the IEEE 96(4), 648–667 (2008)
18. Jain, P., Kapoor, A.: Active learning for large multi-class problems. In: CVPR, pp. 762–769. IEEE (2009)
19. Jing, F., Li, M., Zhang, H., Zhang, B.: Entropy-based active learning with support vector machines for content-based image retrieval. In: ICME, pp. 85–88. IEEE (2004)
20. Katsavounidis, I., Jay Kuo, C.C., Zhang, Z.: A new initialization technique for generalized lloyd iteration. IEEE Signal Processing Letters 1(10), 144–146 (1994)
21. Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content-based multimedia information retrieval: State of the art and challenges. ACM Trans. Multimedia Comput. Commun. Appl. 2(1), 1–19 (2006)
22. Lindenbaum, M., Markovitch, S., Rusakov, D.: Selective sampling for nearest neighbor classifiers. Machine Learning 54(2), 125–152 (2004)
23. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)

24. Lux, M., Chatzichristofis, S.A.: Lire: lucene image retrieval: an extensible java cbir library. In: MM 2008: Proceeding of the 16th ACM International Conference on Multimedia, pp. 1085–1088. ACM, New York (2008)
25. Pavlidis, T.: Limitations of content-based image retrieval (2008), http://theopavlidis.com/technology/CBIR/PaperB/vers3.htm
26. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: A database and web-based tool for image annotation. International Journal of Computer Vision 77(1-3), 157–173 (2008)
27. Sivic, J., Zisserman, A.: Efficient visual search for objects in videos. Proceedings of the IEEE 96(4), 548–566 (2008)
28. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Trans. Pattern Anal. Mach. Intell. 22(12), 1349–1380 (2000)
29. Tamura, H., Mori, S., Yamawaki, T.: Textural features corresponding to visual perception. IEEE Trans. Systems, Man and Cybernetics 8(6), 460–473 (1978)
30. Tax, D.M.: One-class classification. Ph.D. thesis, Delft University of Technology, Delft, The Netherlands (June 2001)
31. Tong, S., Chang, E.Y.: Support vector machine active learning for image retrieval. In: ACM Multimedia, pp. 107–118 (2001)
32. Wang, J.Z., Li, J., Wiederhold, G.: Simplicity: Semantics-sensitive integrated matching for picture libraries. IEEE Trans. Pattern Anal. Mach. Intell. 23(9), 947–963 (2001)
33. Wang, J., Hua, X.S.: Interactive image search by color map. ACM TIST 3(1), 12 (2011)
34. Wei, X.Y., Yang, Z.Q.: Coached active learning for interactive video search. In: Candan, K.S., Panchanathan, S., Prabhakaran, B., Sundaram, H., Chi Feng, W., Sebe, N. (eds.) ACM Multimedia, pp. 443–452. ACM (2011)
35. Winn, J.M., Criminisi, A., Minka, T.P.: Object categorization by learned universal visual dictionary. In: ICCV, pp. 1800–1807. IEEE Computer Society (2005)

Finding Correlations between 3-D Surfaces: A Study in Asymmetric Incremental Sheet Forming

M. Sulaiman Khan, Frans Coenen, Clare Dixon, and Subhieh El-Salhi

Department of Computer Science, Ashton Building, Ashton Street, Liverpool L69 3BX, United Kingdom
{mskhan,coenen,cldixon,salhi}@liverpool.ac.uk

Abstract. A mechanism for describing 3-D local geometries is presented which is suitable for input into a classifier generator. The objective is to predict the springback that will occur when Asymmetric Incremental Sheet Forming (AISF) is applied to sheet metal to produce a desired shape so that corrective measures can be applied. The springback is localised hence the desired before shape and the actual after shape are expressed using the concept of a Local Geometry Matrix (LGMs). The reported evaluation demonstrates that the LGM idea can be usefully employed to capture local geometries with respect to individual shapes.

Keywords: Classification, 3-D Surface Modelling, Asymmetric Incremental Sheet Forming.

1 Introduction

An investigation into data mining techniques for identifying correlations between 3-D surfaces, and then to predict likely correlations with respect to "new" 3-D surfaces is presented[1]. More specifically, the investigation is directed at predicting the springback that occurs during Asymmetric Incremental Sheet Forming (AISF); a manufacturing process used to shape sheet metal. The advantages of AISF are that it is comparatively inexpensive and does not require heating of the metal (heating introduces potential fracture points and adds an additional financial overhead). The disadvantage of AISF metal forming is that springback is introduced into the shape. The AISF process commences with a desired *input* shape, defined in terms of a set of 3-D coordinates, and produces an *output* shape which, as a result of the process, is a "variation" of the desired input shape because of the springback that has been introduced. The nature of the resulting output shape can be recorded using an optical measuring system[2] to generate a

[1] The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 266208.

[2] In our case the GOM (Gesellschaft fr Optische Messtechnik) optical measuring tool produced by GOM mbH was used.

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 366–379, 2012.

second set of 3-D coordinates. Thus we have before and after *coordinate clouds* (input and output). Therefore, given a desired shape T, a process P and a result T' we wish to learn the correlation A between T and T' so that given a new shape S we can predict the outcome S' and consequently attempt to redefine S so as to minimise the springback. A simple answer to the problem can be expressed as $A = \frac{T+T'}{2}$. However, the springback introduced by process P is not evenly spread across the entire output shape; it is conjectured by domain experts that the nature of the springback may be dependent on a number of factors such as tool head shape, tool head speed, tool head pitch, lubricant, blank holder, type of alloy, sheet thickness, sheet size, shape geometry and the forming process used. However, it is suggested that a key influencing factor is the geometry of the desired shape. The nature of the springback/correlation between T and T' as a result of the process P is localised according to the geometry of T (and by extension T').

The proposed technique uses a grid representation for both T and T' so that by registering and superimposing T' over T we can determine the springback between the two surfaces for each grid point contained in T. We then numerically define the "local surface" surrounding each grid point in T in terms of the change in elevation (the z coordinate) of each of the eight neighbouring grid points compared to the z coordinate of the "centre" grid point. This then gives us a 3×3 Local Geometry Matrix (LGM) for each grid point. Any given 3-D surface can then be described in terms of a set of records (one per grid point) such that each record comprises an LGM. If we describe T in this way and for each record include an error value e obtained by comparing correlated grid points in T and T' we can produce a "training set" set that can be used to train a classifier. The fundamental idea is then, given a new shape S, to use the classifier to predict the springback (S') so that corrective measures can be applied to S to compensate for the springback to give S'' (a corrected definition of S' to be fed back into the AISF process).

We evaluated the proposed technique by generating a set of records, using the process described above, and applying a standard Ten-fold Cross Validation (TCV) technique where we built the classifiers using nine tenths of the data and tested on the remaining tenth (using a different tenth as the test set on each occasion). For the evaluation we used a large and a small flattened square based pyramid. As will be demonstrated later in this paper the experiments produced excellent results; a classification best accuracy of 100% was obtained.

The rest of this paper is structured as follows. In section 2 a brief overview of some related previous work is presented. Sections 3 and 4 describe respectively our LGM representation and the mechanism for measure deformation between T and T'. The processing of the shape representation to produce a training data set from which classifiers can be generated is described in Section 5. The actual generation of our desired classifiers is then considered in Section 6, followed by the evaluation of the proposed technique in Section 7. Finally some conclusions are presented in Section 8.

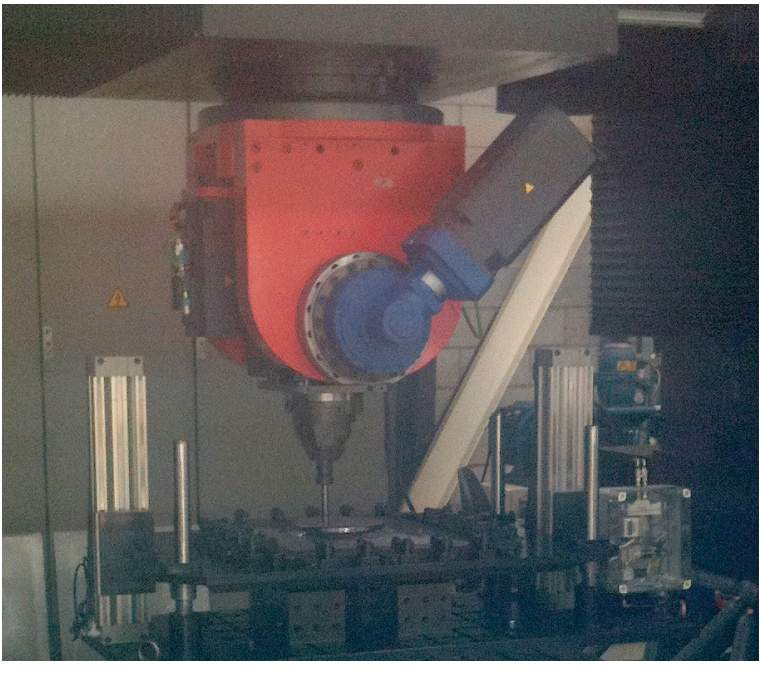


Fig. 1. Asymmetric Incremental Sheet Forming (AISF), the work piece is clamped in position while the tool head "pushes out" the desired shape, on release springbuck occurs as a result of which the final shape is not the desired shape

2 Previous Work

When manufacturing parts using AISF a metal sheet is clamped into a holder and the desired shaped is produced using the continuous movement of a simple round-headed forming tool. A typical AISF machine is shown in Figure 1. The forming tool is provided with a "tool path" generated by a CAD model and the part is "pressed" out according to the co-ordinates of the tool path. However, due to the nature of the metal used and the manufacturing process, *springback* occurs which means that the geometry of the shaped part is different from the geometry of the desired part, i.e. some deformation is introduced. In [1] the authors consider a number of products that could potentially be formed using AISF and demonstrated that the accuracy of the formed part needs to be improved before this process could be used in a large scale production. In [13] the authors considered two drawbacks of the AISF process relating to the metal thickness and the geometric accuracy of the resulting shape.

There has been substantial reported work on dynamic tool path correction in the context of laser guided tools (see for example [5] and [8]). However, AISF requires that the tool path is specified in advance rather than as the process develops. In [2] the authors propose a multi-stage forming technique, i.e. rather

than a single pass by the machine tool, several are made so that the process can take into account the deformatiuon that is introduced by springback. As a case study a square based pyramid shape was considered (similar to those considered in this paper). From [2] it is interesting to note that the initial geometry with a corner radius larger than the desired shape and a number of forming stages produced the least deformation.

For several years the Finite Element Method (FEM) has been used as an industry standard for calculating the springback of sheet metals in forming processes [19]. However, the results of FEM calculations are not very accurate because of the involvement of complex non-linear factors [25]. A data mining approach is advocated in the paper. Not unexpectedly data mining techniques have been applied to sheet metal forming. There are many examples of the use of neural networks to support sheet metal forming [7,14,15,16,18,21,24]. Considering one example only, in [21] a neural network is trained to predict springback. Several inputs were used for the neural network to train on, such as thickness, radius, springback etc. It was observed that the predictions made by the neural networks were very close to the simulation results. Rule based learning techniques have also been popular. For example in [26] rule based mining is used to extract knowledge from data generated by Finite Element Analysis (FEA). A four part knowledge discovery model was proposed that included: (i) product design and development, (ii) data-collection, (iii) knowledge discovery and (iv) management and reuse. In the fourth part the extracted knowledge was filtered with the aim of supporting the design process. Another similar approach was proposed in [28] for the U-draw bending process where a rule based system was used to extract knowledge from FEA simulation data. The nature of the material and various process parameters were used to study their effect on springback. However, there has been very little reported work on the use of data mining techniques to address the AISF springback problem as formulated in this paper. The approach proposed advocated in this paper is not only concerned with extracting knowledge from the sheet metal forming data but also proposing a classification model to predict and apply it in order to minimise the springback effect.

3 Grid Representation

The inputs to the proposed procedure are an input "coordinate cloud" C_{in} (representing T) and an output coordinate cloud C_{out} (representing T'). Each coordinate cloud comprises a set of N, (x, y, z) coordinate triples, such that $x, y, z \in \mathbb{R}$. The number of coordinates per cm^2 (within the X-Y plane) in each coordinate cloud varies between 120 points per cm^2 to 20 points per cm^2 depending on how the data is generated/collected. The C_{in} coordinate cloud is typically obtained from a tool path specification generated using a CAD model, while C_{out} is collected using an optical measuring system, $|C_{out}|$ is typically less than $|C_{in}|$. Both coordinate clouds were registered to the same reference origin and orientation.

We first cast C_{in} into a grid representation (Figure 2) such that each grid point is defined by a $\langle x_i, y_j \rangle$ coordinate value pair. The number of grid lines is

defined by some grid spacing d. Each coordinate pair $\langle x_i, y_j \rangle$ in the grid has a z value calculated by averaging the z values associated with the part of the input coordinate cloud contained in the $d \times d$ grid square centered on the point $\langle x_i, y_j \rangle$ (Figure 3). We then cast the C_{out} coordinate cloud into the same grid format so that we end up with two grids, G_{in} and G_{out}, describing the before and after surfaces (T and T').

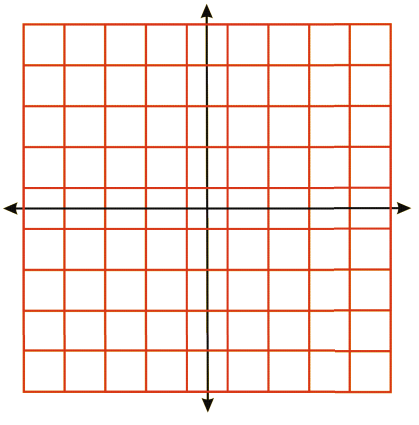


Fig. 2. Example grid referenced to a central origin (grid spacing = d)

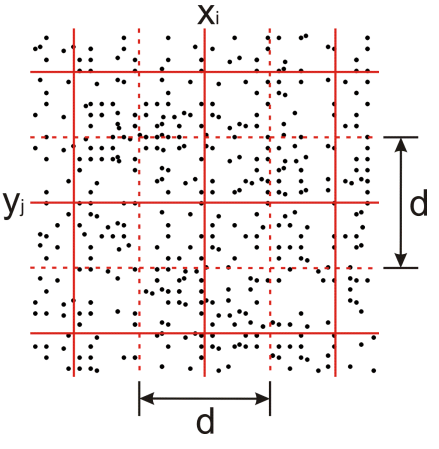


Fig. 3. Coordinate cloud points associated with a grid point $\langle x_i, y_j \rangle$

4 Springback Measurement

A simple mechanism for establishing the degree of springbok (e) at a particular grid point is simply to measure difference between the z values in G_{in} and G_{out} (Figure 4). However, a more accurate measure is to determine the length of the surface normal from each grid point in G_{in} to the point where it intersects G_{out}. The distance between any two three dimensional points can be calculated using the point to point Euclidean distance formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{1}$$

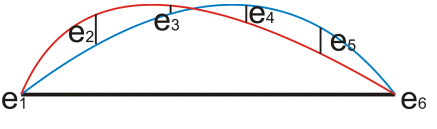


Fig. 4. Cross section at a grid line showing simple vertical springback error calculation between a before and after shape

However, application of equation (1) first requires knowledge of the x, y, z coordinates of the point where the normal intersects G_{out}. With respect to the work described in this paper we have used the line plane intersection method [9] to determine the length of the normal between two surfaces. Using this approach we find the normal to a plane by calculating the cross product of two orthogonal vectors contained within the plane. Once we have the normal we can calculate the equation for the line that includes the start and end points of the normal and then determine the point at which this line cuts G_{out}. We can then calculate the length of the normal separating the two planes. The process is as follows (with reference to Figure 5):

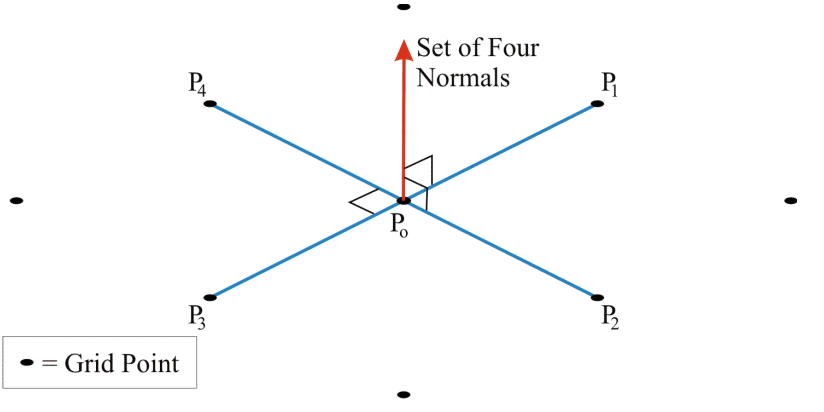


Fig. 5. Error calculation using the line plane intersection method

1. For each grid point in G_{in} first identify the four neighbouring grid points in the X and Y planes as shown in Figure 5 (except at edges and corners where three and two neigbouring grid points will be identified respectively).
2. Define a set of four vectors $V = \{v_1, \ldots, v_4\} = \{\langle p_\emptyset, p_1\rangle, \langle p_\emptyset, p_2\rangle, \langle p_\emptyset, p_3\rangle, \langle p_\emptyset, p_4\rangle\}$, each described in terms of its x-y-z distance from $p_\emptyset$ (the origin for the vector system).
3. Using the four vectors in V, four surface normals are calculated, $N = \{n_1 \ldots n_4\}$, by determine the cross product between each pair of vectors: $v_1 \times v_2$, $v_2 \times v_3$, $v_3 \times v_4$, $v_4 \times v_1$. Note that to validate a surface normal n_i, the dot product of one of its associated vectors v_j and n_i must be equal to zero ($n_i \cdot v_j = 0$).
4. For each normal $n_1 \ldots n_4$ calculate the local plane equation in G_{in} that includes $P_\emptyset$ (thus using, in turn, points $\{P_1, P_\emptyset, P_2\}$, $\{P_2, P_\emptyset, P_3\}$, $\{P_3, P_\emptyset, P_4\}$ and $\{P_4, P_\emptyset, P_1\}$). The plane equation is given by Equation 2.

$$ax + by + cz + d = 0 \tag{2}$$

5. For each plane equation identified in (4) determine the parametric equations (a set of equations/functions which describe the x, y and z coordinates of

the graph of some line in a plane) [9] of the surface normal as a straight line according to the identities given in equation 3.

$$x = a + i(t), \qquad y = b + j(t), \qquad z = c + k(t) \tag{3}$$

where t is a constant; a, b and c are the x-y-z coordinates for the point $p_\emptyset$; and i, j and k are the normal components. The constant t is calculated by substituting the parametric equations in plane equation 2 for x, y and z.

6. Once the parametric equations for each surface normal are found, they are then used to compute the points of intersection of each normal with G_{out}.
7. We then use the coordinates for each of the four points of intersection and $p_\emptyset$ to calculate the Euclidean distance (the error) between $p_\emptyset$ and each intersection point to give four error values $E = \{e_1 \dots e_4\}$
8. We now have four error values for each grid point (except at the corners and edges where we will have two or three respectively), we then find the "overall" error e simply by computing the average error:

$$e = \frac{\sum_{i=1}^{i=|E|} e_i}{|E|} \tag{4}$$

On completion of the process our input grid, G_{in}, will comprise a set of (x, y, z) coordinates describing the N grid points, each with an associated springback (error) value e.

5 Surface Representation (The Local Geometry Matrix)

In this section we describe how local geometries can be represented using the concept of a Local Geometry Matrix (LGM). From the foregoing it has already been noted that the value of e is particularly influenced by the nature of the geometry of the desired surface (shape). We can model this according to the change in the δz value of the eight grid points surrounding each grid point. (Of course along the edges and at the corners of the grid we will have fewer neighbouring grid points). Thus we generate n records (where n is the number of grid points) each typically comprising nine values, eight δz values and an associate e value. We, then coarsen the δz values by describing them using qualitative labels taken from a set L to describe the nature of the "slope" in each of the eight neighbouring directions. Therefore we can describe $|L|^8$ different "local geometries" if we take orientation into consideration. Thus if we have a label set $\{negative, level, positive\}$ we can describe $3^8 = 6561$ different local geometries.

Example 1. Considering the flattened square based pyramid shape in Figure 6 and a section of the surface, measuring 3×3 grid points, covering an edge as shown, then the z coordinate matrix associated with the grid point might be as shown in Table 1. The δz values are then calculated by subtracting the centre z value from each of the surrounding z values in turn. With respect to the example the δz matrix result would be as shown in Table 2 (the centre grid reference

point always has a value of 0). We refer to this matrix as a Local Geometry Matrix (LGM). Assuming $L = \{negative, level, positive\}$, and ordering the matrix elements (grid points) in a clockwise direction from the top left, would give us a record of the following form where e is the error value associated with the grid point that the record describes:

$$(positive, positive, positive, level, negative, negative, negative, level, e)$$

where e is the error value.

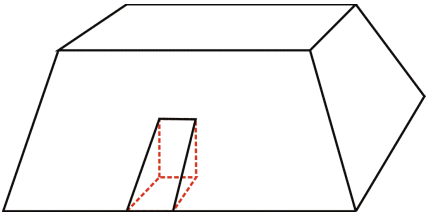


Fig. 6. Square Based Pyramid With Side Section (Example 1)

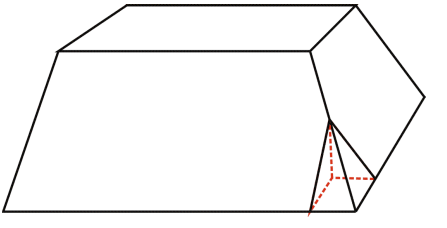


Fig. 7. Square Based Pyramid With Corner Section (Example 2)

Table 1. Z matrix for Example 1

20	20	20
10	10	10
0	0	0

Table 2. LGM for Example 1

10	10	10
0	0	0
-10	-10	-10

Example 2. Again considering a flattened square based pyramid shape but now looking at a section of the surface, measuring 3×3 grid points, located at the corner of the shape as shown in Figure 7, the z coordinates associated with the grid point might be as shown in Table 3. The LGM would then be as shown in Table 4. Again assuming $L = \{negative, level, positive\}$ the resulting record would be:

$$(positive, level, negative, negative, negative, negative, negative, level, e)$$

The proposed representation can be used to capture all local geometries. Given a suitable test shapes (in this paper we have used two flattened square based

Table 3. Z matrix for Example 2

20	10	0
10	10	0
0	0	0

Table 4. LGM for Example 2

10	0	-10
0	0	-10
-10	-10	-10

pyramid shapes, one substantially larger than the other) we can link error values all the possible geometries. It should be noted that, at least conceptually, the use of LGMs is akin to the use of Local Binary Patterns (LBPs) as applied in the context of image texture analysis [12,20].

The set of error values was also discretised using a set of five qualitative labels each describing a particular sub-range of error values. The five sub-ranges used were of equal size and designed to encompass the full range of error values from the recorded minimum to the recorded maximum.

6 Classifier Generation

There are a number of classification mechanisms that can be applied to data, pre-processed in the manner described above, so as to generate a classifier that can be applied to unseen data. In the work described here we favour a classifier that generates rules. Rule base representations offer two principal advantages:

1. Rule representations are intuitive; they are simple to interpret and understand.
2. Because of (1), the validity of rules can easily verified by domain experts.

It is possible to generate rules using many of the available classifier generation techniques, although some are more suited to rule generation than others. Classification Association Rule (CAR) generators directly generate rule sets. There are a number of well established CAR Mining (CARM) algorithms that can be adopted: examples include CPAR [27], CMAR [17] and TFPC [3,4]. Although the principle is the same each of these operates in slightly different manner. It is also fairly straightforward to generate rule sets using decision tree classifiers such as the ID3 Algorithm [23], C4.5 [22] or the MARS Algorithm [11]. Generating rules from Neural Network based classification techniques or Support Vector Machines is less straight forward but can be done [10,6]. In the evaluation section (Section 7) we compare the operation of all three of CPAR, CMAR, TFP and C4.5. Using these algorithms the required input is a set of binary valued attributes. Thus given our representation (see above) we will use $|L| \times 8$ attributes. Thus if $|L| = 5$ the input training data will comprise 45 columns, 5×8 attributes plus the class (error) attributes (1×5).

6.1 Classifier Application

Once we have generated our desired classifier we will wish to apply it to unseen data, i.e. a new shape S so that we can predict S'. To do this the coordinate cloud describing S must be expressed in terms of its components in the same manner as used to define the training data. Thus the coordinate cloud for S must be expressed as a grid using the same value of d as that used to generate the classifier, which must then be converted in to a set of records comprising $L \times 8$ attributes so as to be compatible with the generated classification rule representation (again there will be some missing data at edges and corners).

7 Evaluation

This section reports on the outcomes of the evaluation, using a small (S1) and a large (S2) square based pyramid (similar to that used in [2]), of the proposed approach. The two pyramids were constructed using the AISF process. In each case the before cloud was the input to the AISF process. The resulting after clouds were obtained using a GOM optical measuring tool. The objective of the evaluations were:

1. To identify the most appropriate value for d, the grid spacing, so as to maximise the descriptive accuracy of the rules.
2. To identify the most appropriate value for $|L|$, the number of qualitative labels used to describe local geometries, again so as to maximise the descriptive accuracy of the rules.
3. To determine the overall effectiveness of the proposed approach, in terms of classification accuracy.

A range of values for d and $|L|$ were tested; d values from 5 to 20 in steps of 5 were used (the units are millimeters), and $|L|$ values of 3, 5 and 7. Some statistics regarding the size of the resulting data sets are presented in Table 5. As noted above we tested a number of CARM algorithms (CMAR, CPAR, TFPC) and the C4.5 decision tree classifier. We used Ten-fold Cross Validation (TCV) throughout.

Table 5. Number of records using a range of values for d (S1 = Small Pyramid, S2 = Large Pyramid)

d	S1	S2
5	756	4833
10	182	1260
15	90	552
20	56	306

The results from the TCV evaluations are presented in Table 6. Best accuracy figures are highlighted in bold font. From the table the following can be noted:

1. We can predict the springback (error) to a high level of accuracy (best accuracy of 92% for the small pyramid, and 100% for the large pyramid).
2. Decision tree classifier worked the best with respect to both pyramids.
3. A high size of L seems to be beneficial (the best value for $|L|$ was $|L| = 7$).
4. An argument can be made that a small gird size (d=5 or d=10) is also beneficial.

The fact that a high value for $|L|$ is beneficial is not suprising because the greater the value of $|L|$ the more expressive the label descriptors. However, if $|L|$ becomes too large there are implications for the runtime complexity of the approach; and, more significantly, may result in "overfitting" of the training data. Overall

it can be seen that some very good accuracies were obtained and that best accuracy results were obtained using C4.5. This was a very encouraging result. The experiments indicate that we can generate classifiers (as demonstrated) for given shapes, and that this classification approach can provide a sound AI platform for (say) an Intelligent Process Model (IPM).

Table 6. TCV Classification Results (S1 = Small Pyramid, S2 = Large Pyramid)

	$\|L\| = 3$				$\|L\| = 5$				$\|L\| = 7$			
	$d = 5$											
	CMAR	CPAR	TFPC	C4.5	CMAR	CPAR	TFPC	C4.5	CMAR	CPAR	TFPC	C4.5
S1	71.60	63.45	57.91	83.28	70.29	62.70	73.65	88.64	72.56	63.50	75.37	**90.38**
S2	92.98	-	93.21	97.85	93.31	-	93.06	99.01	95.38	-	93.23	**99.52**
	$d = 10$											
S1	64.67	56.89	67.63	80.28	78.50	77.39	77.05	91.67	79.61	77.94	77.60	**91.67**
S2	92.38	91.98	92.70	96.35	93.17	91.75	94.21	98.65	93.17	91.35	94.37	**99.37**
	$d = 15$											
S1	73.33	65.56	67.78	77.78	73.33	62.22	70.00	84.44	77.78	67.78	66.67	**87.78**
S2	93.27	90.91	92.00	96.73	91.64	90.18	91.09	99.09	92.36	90.36	91.45	**99.27**
	$d = 20$											
S1	45.82	59.82	70.00	88.00	56.73	66.55	71.67	90.00	39.82	57.64	71.33	**92.00**
S2	93.67	95.67	91.24	97.33	92.33	95.00	90.57	99.67	92.67	95.33	92.22	**100.0**

8 Conclusions and Perspectives

In this paper we have described a mechanism for discovering correlations between 3-D surfaces. More specifically we have described a mechanism for discovering local correlations between a target shape T and a shape T' produced as a result of the application of an AISF process. We have demonstrated that the mechanism we have proposed to represent local geometries, using the LGM concept, can be used to generate accurate classifiers to predict (and consequently apply) errors in shapes produced using AISF.

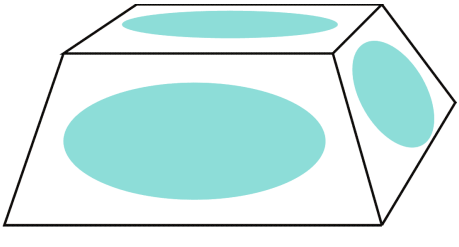


Fig. 8. Areas of greatest springback in a flattened square based pyramid shape

Given the above it is suggested that classification is an appropriate technology for building Intelligent Process Models (IPMs). However, we believe our current representation still needs further refinement. Firstly the ranging mechanisms used to discretize LGM values may not be the most appropriate if we wish to apply a classifier built using one shape to another type of shape. It may also be the case that the current representation needs to be augmented with additional information regarding the proximity of grid points to edges and/or corners. The reason for this is that it is conjectured that the error magnitude of the springback increases as we move away from edges (Figure 8). This means that the errors should be greater in the large pyramid than in the small pyramid. Two possible mechanisms whereby we may augment our current representation are suggested. The first involves using two or more d values so that we capture both the "big picture" as well as the "small picture". Alternatively we can include an edge/corner proximity measure (p). Currently we describe shapes using a grid. For each grid point (except at edges and corners) we have eight surrounding grid points. We have established that local geometry can be described by the difference in z values between the center grid points and the surrounding eight points. In each case this gives a 3×3 Local Geometry Matrix (LGM) describing the δz values (with the value 0 at the center representing the grid point). Some of these LGM configurations will indicate the presence of edges and corners provided that the grid distance (d) is sufficient to capture this. Given a "bank" of LGMs describing edge and corner configurations we can use pattern matching to identify the corners and edges in any given piece. We can then use this knowledge to determine values for p for each grid point. The long term goal is to produce a generally applicable classifier that can be applied to any shape (of course other influencing factors such as material and tool head speed must be kept constant).

Currently errors are defined as the distance along the normal from the before surface to where it intersects the after surface. We calculate four normals for each grid point and consequently four error values are obtained. The specific error associated with a grid point is then the average of these four error values. To produce a new coordinate cloud, S'', we can simply reverse these errors. The reverse errors can either be applied to the grid points or the before coordinate cloud. If we apply to the cloud and if there is a significant difference between the error associated with adjacent grid points, we may get a stepping effect (especially if d is large); in which case some sort of smoothing may be required. If we apply to the grid coordinates we may not have sufficient points to allow a new shape to be manufactured. We will therefore need to use small values of d, $d = 1$ seems to be a good value. It should also be noted that we believe that simply reversing the error is unlikely to produce a good S'', we therefore propose to apply a factor f to the errors. The intention is that the nature of f will be dependent on the local geometry as defined so far, but augmented by the additional work on representing local geometries (as described above) that we intend to undertake.

Acknowledgements. The authors would particularly like to thank Markus Bambach, Babak Taleb and David Bailly, from RWTH-IBF (Germany) for their

support in the preparation and provision of the test data used to evaluate the proposed mechanism described in this paper. The authors would also like to thank Mariluz Penalva, Asun Rivero and Antonio Rubio from Tecnalia-IS (Spain) for comments on an earlier draft of this paper; and Nicolas Guegan from AIRBUS (France) and Joachim Zettler from EADS (Germany) for their extremely helpful advice on various aspects of the work described.

References

1. Allwood, J.M., King, G.P.F., Duflou, J.: A structured search for applications of the incremental sheet-forming process by product segmentation. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 219(2), 239–244 (2005)
2. Bambach, M., Taleb Araghi, B., Hirt, G.: Strategies to improve the geometric accuracy in asymmetric single point incremental forming. Production Engineering Research and Development 3(2), 145–156 (2009)
3. Coenen, F., Leng, P.: Obtaining best parameter values for accurate classification. In: Proc. IEEE Int. Conf. on Data Mining (ICDM 2005), pp. 597–600 (2005)
4. Coenen, F., Leng, P., Zhang, L.: Threshold Tuning for Improved Classification Association Rule Mining. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 216–225. Springer, Heidelberg (2005)
5. Dearden, G., Edwardson, S.P., Abed, E., Bartkowiak, K., Watkins, K.G.: Correction of distortion and design shape in aluminium structures using laser forming. In: 25th International Congress on Applications of Lasers and Electro Optics(ICALEO 2006), pp. 813–817 (2006)
6. Diederich, J.: Rule extraction from support vector machines. Springer New York Inc. (2008)
7. Dunston, S., Ranjithan, S., Bernold, E.: Neural network model for the automated control of springback in rebars. IEEE Expert: Intelligent Systems and Their Applications, 45–49 (1996)
8. Edwardson, S.P., Watkins, K.G., Dearden, G., Magee, J.: Generation of 3D shapes using a laser forming technique. In: Proceedings of ICALEO 2001, pp. 2–5 (2001)
9. Egerton, P.A., Hall, W.W.: Computer graphics: Mathematical first steps. Simon and Schuster International (1998)
10. Elalfi, A.E., Haque, R., Elalami, M.E.: Extracting rules from trained neural network using GA for managing e-business. Applied Soft Computing 4(1), 65–77 (2004)
11. Friedman, J.H.: Multivariate adaptive regression splines. The Annals of Statistics 19(1), 1–67 (1991)
12. Guo, G., Zhang, L., Zhang, D.: A completed modeling of local binary pattern operator for texture classification. IEEE Transactions on Image Processing 19(6), 1657–1663 (2010)
13. Hirt, G., Ames, J., Bambach, M., Kopp, R., Kopp, R.: Forming strategies and process modelling for cnc incremental sheet forming. CIRP Annals - Manufacturing Technology 53(1), 203–206 (2004)
14. Inamdar, M., Date, P.P., Narasimhan, K., Maiti, S.K., Singh, U.P.: Development of an artificial neural network to predict springback in air vee bending. International Journal of Advanced Manufacturing Technology 16(5), 376–381 (2000)
15. Kim, D.J., Kim, B.M.: Application of neural network and fem for metal forming processes. International Journal of Machine Tools and Manufacture 40(6), 911–925 (1999)

16. Kinsey, B., Cao, J., Solla, S.: Consistent and minimal springback using a stepped binder force trajectory and neural network control. Journal of Engineering Materials and Technology 122(1113), 113–118 (2000)
17. Li, W., Han, J., Pei, J.: Cmar: Accurate and efficient classification based on multiple class-association rules. In: Proc. IEEE Int. Conf. on Data Mining (ICDM 2005), pp. 369–376 (2001)
18. Manabe, K., Yang, M., Yoshihara, S.: Artificial intelligence iidentification of process parameters and adaptive control system for deep drawing process. Journal of Materials Processing Technology 80-81, 421–426 (1998)
19. Narasimhan, N., Lovell, M.: Predicting springback in sheet metal forming an explicit to implicit sequential solution procedure. Finite Elements in Analysis and Design 33(1), 29–42 (1999)
20. Ojala, T., Inen, M.P., Maéenpaé, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7), 971–987 (2002)
21. Pathak, K.K., Panthi, S., Ramakrishnan, N.: Application of neural network in sheet metal bending process. Defence Science Journal 55(2), 125–131 (2005)
22. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc. (1993)
23. Quinlan, J.R.: Induction of decision trees. Machine Learning 1(1), 81–106 (1986)
24. Rufni, R., Cao, J.: Using neural network for springback minimization in a channel forming process. Journal of Materials and Manufacturing 107(5), 65–73 (1998)
25. Xu, J., Zhang, Z., Wu, Y.: Application of data mining method to improve the accuracy of springback prediction in sheet metal forming. Journal of Shanghai University (English Edition) 8(3), 348–353 (2004)
26. Yin, J.L., Li, D.Y.: Knowledge discovery from finite element simulation data. In: Proceedings of 2004 International Conference on Machine Learning and Cybernetics, pp. 1335–1340 (2004)
27. Yin, X., Han, J.: Cpar: Classification based on predictive association rules. In: Proc. SIAM Int. Conf. on Data Mining (SDM 2003), pp. 331–335 (2003)
28. Zhang, S., Luo, C., Peng, Y.H., Li, D.Y., Yang, H.B.: Study on factors affecting springback and application of data mining in springback analysis. Journal of Shanghai Jiaotong University E-8(2), 192–196 (2003)

Combination of Physiological and Behavioral Biometric for Human Identification

Emdad Hossain and Girija Chetty

Faculty of Information Sciences and Engineering, University of Canberra, Australia
emdad.hossain@canberra.edu.au

Abstract. In this paper we propose a novel person-identification scheme based on gait biometric information in surveillance videos using simple PCA-LDA features, and RBF-MLP and SMO-SVM classifier. The experimental evaluation on resolution surveillance video images from a publicly available database [1] showed that the combined PCA-MLP and LDA-MLP technique turns out to be a powerful method for capturing identity specific information from walking gait patterns.

Keywords: multimodal, MLP, PCA, LDA, identification, SMO, naïve bayes, J48.

1 Introduction

Human identification from arbitrary views is a very challenging problem, especially when one is walking at a distance. Over the last few years, recognizing identity from gait patterns has become a popular area of research in biometrics and computer vision, and one of the most successful applications of image analysis and understanding. Also, gait recognition is being considered as a next-generation recognition technology, with applicability to many civilian and high security environments such as airports, banks, military bases, car parks, railway stations etc. For these application scenarios, it is not possible to capture the frontal face, and is of low resolution. Hence most of traditional approaches used for face recognition fail; however, several studies have shown that humans can identify a person from a distance from their gait or the way they walk. Even if frontal face is not visible, it is possible to establish the identity of the person using certain static and dynamic cues such as from face, ear, walking style, hand motion during walking etc. If automatic identification systems can be built based on this concept, it will be a great contribution to surveillance and security area. However, each of these cues or traits captured from long range low resolution surveillance videos on its own are not powerful enough for ascertaining identity. A combination or fusion of each of them, along with an automatic processing technique can result in satisfactory recognition accuracies. In this paper, we propose usage of full profile silhouettes of persons without frontal faces from visible range and infrared range, for capturing inherent multi-modality available from static and dynamic cues from the gait patterns of the walking human. This also addresses the problems with frontal faces,

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 380–393, 2012.

such as vulnerability to pose, illumination and expression variations. In addition, one of the biggest shortcomings of frontal face is; user cooperation is mandatory upon data collection. On other hand, long range biometric information from surveillance videos captures several biometric traits such as side face, ear, body shape, and gait, which are a combination of physiological and behavioral biometrics resulting in robust identification approaches. Further, by using certain automatic processing techniques for extracting salient features based on multivariate statistical techniques and learning classifiers, it is possible to enhance the performance in real world operating scenarios. Here, we use simple feature extraction techniques based on principle component analysis (PCA) and linear discriminant analysis (LDA) with different types of learning classifiers. The experimental evaluation of the scheme on a publicly available CASIA [1] database with visible and infra-red gait information shows promising performance improvement.

2 Background

Current state-of-the-art video surveillance systems, when used for recognizing the identity of the person in the scene, cannot perform very well due to low quality video or inappropriate processing techniques. Though much progress has been made in the past decade on visual based automatic person identification through utilizing different biometrics, including face recognition, iris and fingerprint recognition, each of these techniques work satisfactorily in highly controlled operating environments such as border control or immigration check points, under constrained illumination, pose and facial expression variations. To address the next generation security and surveillance requirements for not just high security environments, but also day-to-day civilian access control applications, we need a robust and invariant biometric trait [3] to identify a person for both controlled and uncontrolled operational environments. In this case, trait selection can play vital role. According to authors in [4], the expectations of next generation identity verification involve addressing issues related to application requirements, user concern and integration. Some of the suggestions made to address these issues were use of non-intrusive biometric traits, role of soft biometrics or dominant primary and non-dominant secondary identifiers and importance of novel automatic processing techniques. To conform to these recommendations; often there is a need to combine multiple physiological and behavioral biometric cues, leading to so called multimodal biometric identification system.

Each of the traits, physiological or behavioral have distinct advantages, for eg; the behavioral biometrics can be collected non-obtrusively or even without the knowledge of the user. Behavioral data often does not require any special hardware (other than low cost off the shelf surveillance camera), so , it is very much cost effective. While most behavioral biometrics is not unique enough to provide reliable human identification they have been proved to be sufficiently high accurate [5]. Gait, is such a powerful behavioral biometric, but on its own it cannot be considered as a strong biometric to identify a person. But, if we combine some other equally

nonintrusive biometric with gait; it is expected to be strong combination for human identification. We have taken profile (side) images containing side face and ear biometric traits and used with gait. Here side-face and ear images from the physiological component. Both can be collected unobtrusively without user concern which is very much important especially in the public surveillance. Let us look at some of current as well as traditional biometric identification technologies to put this work in perspective.

For face recognition systems, the performance of 2D face matching systems depends on capability of being insensitive of critical factors such as facial expression, makeup and aging, but also relies upon extrinsic factors such as illumination difference, camera viewpoint, and scene geometry [6]. The 2D face recognition systems are vulnerable to pose, and illumination variations. Use of 3D face can make systems robust to pose and illumination variations. The state of the art 3D face recognition technique using isogeodesic stripes was proposed in [6], 3D face recognition from single image using single reference face shape was proposed in [7], where researchers proposed a novel method for 3D shape recovery of faces that exploits the similarity of faces. It also should be mention that; a number of limitations of 3D identification are; not applicable to public surveillance, high costs, and limited availability of databases [8].

Furthermore, there have been several works reporting use of fingerprints for authenticating identity. A fingerprint is made of a number of ridges and valleys on the surface of the finger [9]. The uniqueness of a fingerprint can be determined by the pattern of ridges and furrows as well as the minutiae points. There are five basic fingerprint patterns: arch, tented arch, left loop, right loop and whorl. Loops make up 60% of all fingerprints, whorls account for 30%, and arches for 10%. Fingerprints are usually considered to be unique, with no two fingers having the exact same dermal ridge characteristics [9]. In fact, there has been a debate on how stable is the uniqueness of fingerprints? Further, due to increasing use of fingerprints for criminal identification, there have been cases of abuse [10]. According to most researchers, Iris and retina are not changeable, but still not out of limitation. The fail to enroll (FTE) rate brings up another important problem. Not all users can use any given biometric system. People without hands cannot use fingerprint or hand-based systems. Visually impaired people have difficulties using iris or retina based techniques. As not all users are able to use a specific biometric system, the authentication system must be extended to handle users falling into the FTE category. This can make the resulting system more complicated, less secure or more expensive [11]. The authors in [11] clearly indentified undeniable limitations for biometric person authentication using fingerprint, iris and retina. Same might goes to person authentication using signature, some systems may also compare visual images of signatures, but the core of a signature biometric system is behavioral, i.e. how it is signed rather than visual, i.e. the image of the signature [11]. It means it has limitations of usage with persons with disability, and it can't be applied to authenticate for large population due to behavioral nature of the trait. Another possible biometric trait is use of hand geometry. In large populations, hand geometry is not suitable for so-called one-to many applications, in which a user is identified from his biometric without any other

identification [12]. Some extreme biometric traits have also been proposed such as use of ear canal. Researchers found that one of the most promising techniques is use of multimodality or combination of biometric traits. Using PCA on combined image of ear and face, researchers in [6, 13] have found that multi-modal recognition results in significant improvement over either individual biometric. But, that has been taken into completely control environment.

Recently, few attempts have been expended on combining various biometrics in a bid to improve upon the recognition accuracy of classifiers that are based on a single biometric. Some biometric combinations which have been tested include face, fingerprint and hand geometry [14]; face, fingerprint and speech [15]; face and iris [16]; face and ear [17]; and face and speech [18]. The fusion of face-ear and gait however, did not attract much attention from the research community. This could be due to difficulty in processing and making sense out of them.

3 Multimodal Identification Scheme

For experimental evaluation of our proposed face-gait identification scheme, we used CASIA Gait Database collected by Institute of Automation, Chinese Academy of Sciences [1]. It is a large multi-view gait database, which is created in January 2005. There are more than 300 subjects. We used two different set of data known as dataset B and Dataset C. Dataset B was captured from 11 views with normal video camera, and 11 different views know as view angles. We used the data captured only in 90 degree view angle. The dataset C was captured with an infrared (thermal) camera. It takes into account four walking conditions: normal walking, slow walking, fast walking, and normal walking with a bag. The videos were all captured at night. Figure 1 shows the sample images in different view angles.

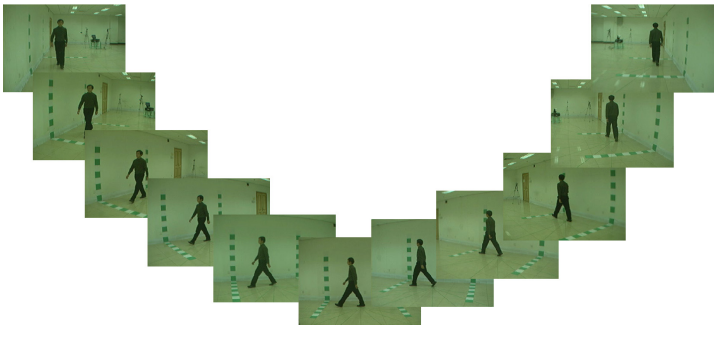


Fig. 1. Sample images

However, we used 50 subjects with a set of extracted silhouettes from Dataset B and another set of extracted silhouettes from Dataset C. Each set consists of 16 images and in total 1600 images for 100 subjects (people). Figure 2 shows the extracted silhouettes from dataset B and C.

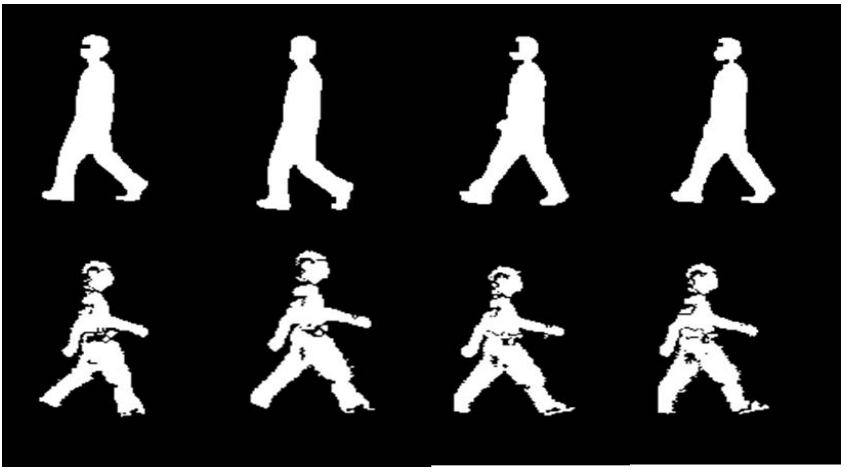


Fig. 2. Extracted silhouettes

Further, we extracted the feature vector for each of the dataset separately by suing PCA (principal component analysis) and Linear Discriminant Analysis (LDA). And classified with different classifiers. So the tests involved PCA-MLP, LDA-MLP, PCA-SMO, LDA-SMO, LDA-Naïve Bayes, LDA-J48. Each of them are described briefly here.

3.1 PCA- LDA

Principle component analysis is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in data can be hard to find in data of high dimension, where the luxury of graphical representation is not available, PCA is a powerful tool for analysing data. The other main advantage of PCA is that once we have found these patterns in the data, and we can compress the data, e.g. by reducing the number of dimensions, without much loss of information. Basically this technique used in image compression [19]. In the image analysis it works like;

$$X=(x1, x2, x3\ldots\ldots N2)\ldots\ldots \quad \ldots\ldots\ldots\ldots\ldots\ldots(1)$$

where the rows of pixels in the image are placed one after the other to form a one dimensional image. Each image is *N* pixels high by *N* pixels wide. For each image it creates an image vector. And then it counts all the images together in one big image-matrix like;

$$\text{Matrix} = (v1, v2, v3\ldots\ldots vN)\ldots\ldots\ldots\ldots \quad \ldots\ldots\ldots\ldots (2)$$

On the other hand, the LDA also closely related to principal component analysis (PCA) and factor analysis in that they both look for linear combinations of variables which best explain the data. LDA explicitly attempts to model the difference

between the classes of data. PCA on the other hand does not take into account any difference in class, and factor analysis builds the feature combinations based on differences rather than similarities. Discriminant analysis is also different from factor analysis in that it is not an interdependence technique: a distinction between independent variables and dependent variables (also called criterion variables) must be made. LDA works when the measurements made on independent variables for each observation are continuous quantities. When dealing with categorical independent variables, the equivalent technique is discriminant correspondence analysis [20]. And in our experiment, LDA shows prominent than PCA. However, The Figure 3 shows the extracted feature (Eigen value) using PCA in Dataset B. Next Section describes several classifiers we examined.

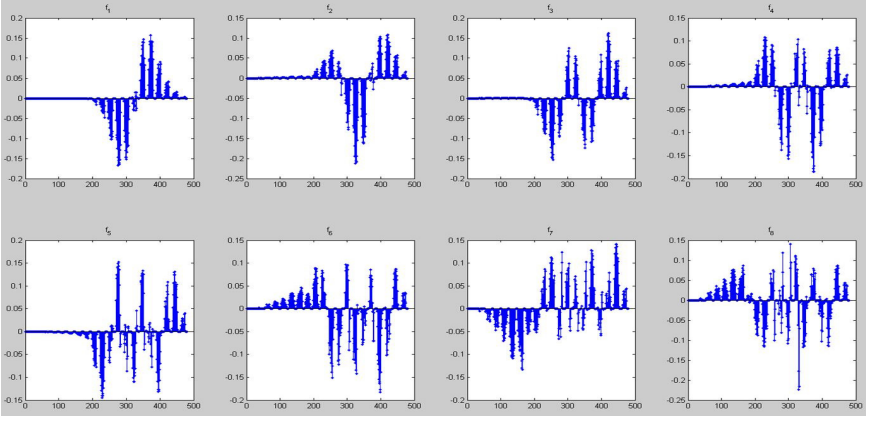


Fig. 3. PCA Eigen Value extracted from silhouettes

3.2 MLP

Multi Layer perceptron (MLP) is a feedforward neural network with one or more layers between input and output layer. Feedforward means that data flows in one direction from input to output layer (forward). This type of network is trained with the backpropagation learning algorithm. MLPs are widely used for pattern classification, recognition, prediction and approximation. Multi Layer Perceptron can solve problems which are not linearly separable [21]. Figure 4 shows that sample layer representation in MLP;

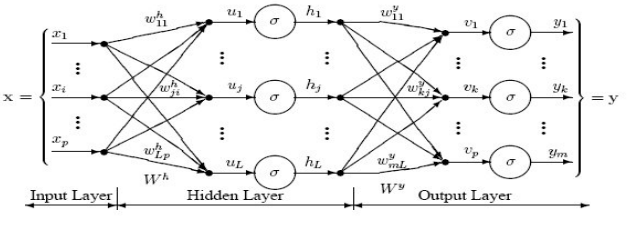


Fig. 4. Perceptron network with three layers [22]

The network diagram shown above is a full-connected, three layer, feed-forward, perceptron neural network. “Fully connected” means that the output from each input and hidden neuron is distributed to all of the neurons in the following layer. “Feed forward” means that the values only move from input to hidden to output layers; no values are fed back to earlier layers (a Recurrent Network allows values to be fed backward) [22]. In our experiment we had 49 input layers, 800 hidden layers (for each data set) and 50 output layer. This is basically based on dimension, instance and the given class.

3.3 J48 Classifier

J48 classifier is same as C4.5 algorithm. It is used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier. It builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set $S = s_1,s_2,...$ of already classified samples. Each sample $s_i = x_1,x_2,...$ is a vector where $x_1,x_2,...$ represent attributes or features of the sample. The training data is augmented with a vector $C = c_1,c_2,...$ where $c_1,c_2,...$represent the class to which each sample belongs [23]. At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. Its criterion is the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recourses on the smaller sub lists [24].

3.4 Naïve Bayes Classifier

A naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature, given the class variable. In plain English it works like [23];

$$\text{Posterior} = (\text{Prior}*\text{Likelihood})/\text{Evidence} \ldots\ldots \qquad \ldots\ldots\ldots (3)$$

All model parameters (i.e., class priors and feature probability distributions) can be approximated with relative frequencies from the training set. These are maximum likelihood estimates of the probabilities. A class prior may be calculated by assuming equiprobable classes (i.e., priors = 1 / (number of classes)), or by calculating an

estimate for the class probability from the training set (i.e., (prior for a given class) = (number of samples in the class) / (total number of samples)). To estimate the parameters for a feature's distribution, one must assume a distribution or generate nonparametric models for the features from the training set [23].

3.5 SMO Classifier

Finally we examined the SVM classifier with SMO. The Sequential Minimal Optimization (SMO) is a simple algorithm in the machine learning area. SMO decomposes the overall QP problem into QP sub-problems, using Osuna's theorem to ensure convergence [21]. Unlike the other methods, SMO chooses to solve the smallest possible optimization problem at every step. The advantage of SMO lies in the fact that solving for multi instance multipliers can be done analytically. In addition, SMO requires no extra matrix storage at all. There are two components to SMO: an analytic method for solving for the two Lagrange multipliers, and a heuristic for choosing which multipliers to optimize [25].

$$y1 /= y2 \Rightarrow \alpha 1 - \alpha 2 = k \tag{4}$$

$$y1 = y2 \Rightarrow \alpha 1 + \alpha 2 = k \tag{5}$$

However, the multi instance multipliers must fulfil all of the constraints of the full problem. The linear equality constraint causes them to lie on a diagonal line. Therefore, one step of SMO must find an optimum of the objective function on a diagonal line segment [25].

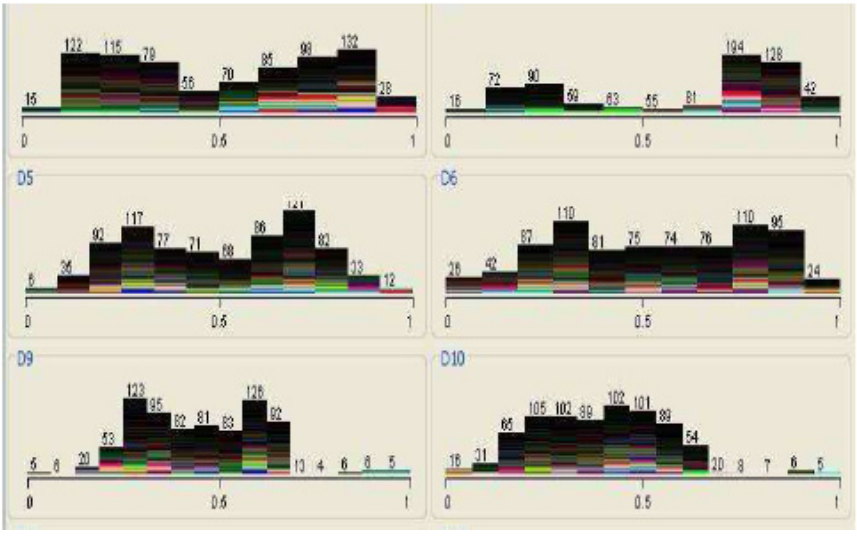


Fig. 5. Normalized feature vectors

4 Experimental Results and Discussion

Once we normalized the data (Figure 5), we examined different classifiers for their recognition accuracy; we examined the performance of difference classifiers using different sets of data. Figure 6 shows average recognition rated for different classifiers studied.

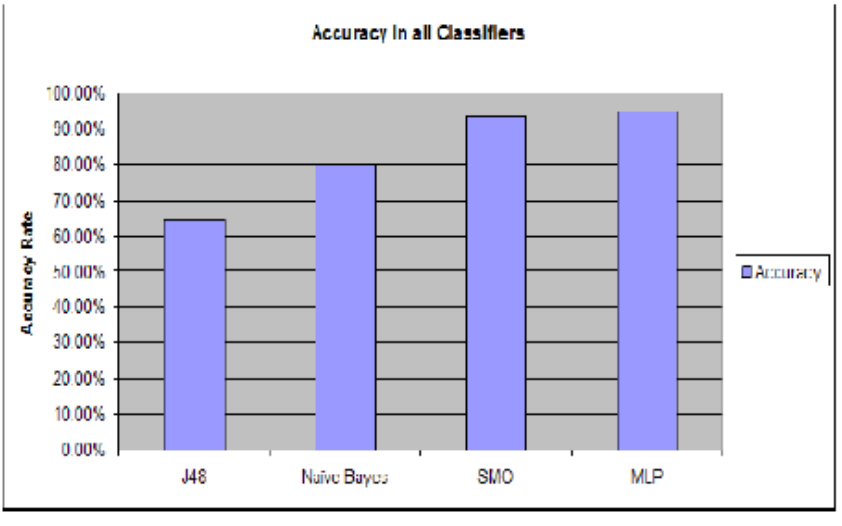


Fig. 6. Accuracy Difference in Applied Classifiers

For this experiment, we used Dataset C which has been captured on infrared camera. For feature extraction we used linear discriminant analysis (LDA) technique. The result shows J48 classifier providing poor result in comparison to all other classifiers which are around 64%. Further, MLP and SMO show significant improvement in classification and they classification accuracy is 94% and 93% respectively. Moreover, to compare both dataset B and C we applied PCA-MLP and LDA-MLP separately. Figure 7 shows the result of Dataset B with LDA-MLP and PCA-MLP.

The result shows, more than 92% accuracy was achieved by using LDA, whereas, only 79.5% accuracy was achieved by using PCA. LDA features providing good results as compared to PCA features. Figure 8 shows the result achieved from Dataset C Moreover, the results show; that we received more than 94% accuracy with LDA features by using MLP classifier, and 83% accuracy with PCA features. These experiments show that overall LDA features working much better as compared to PCA features, for both Dataset B (Visible) and Dataset C (Infrared). The result shows the dataset taken by infrared camera providing 83% and 94% for PCA, LDA respectively.

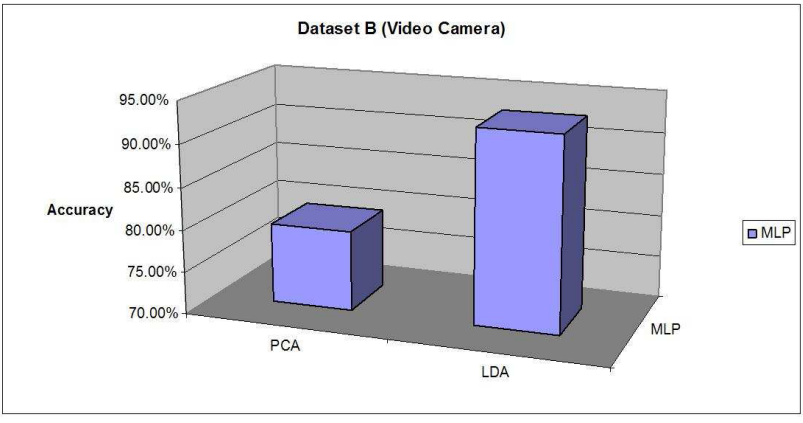


Fig. 7. Accuracy differences in PCA-LDA with MLP

Furthermore, to validate our scheme we applied different folds of cross-validation. Cross-validation, sometimes called rotation estimation is a technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the validation set or testing set). To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds [22]. In this experiment we applied 5, 10, 15, 20, and 30 folds cross-validations. We found that the overall accuracy changing over crossvalidation size. Figure 8 shows the result achieved from dataset B.

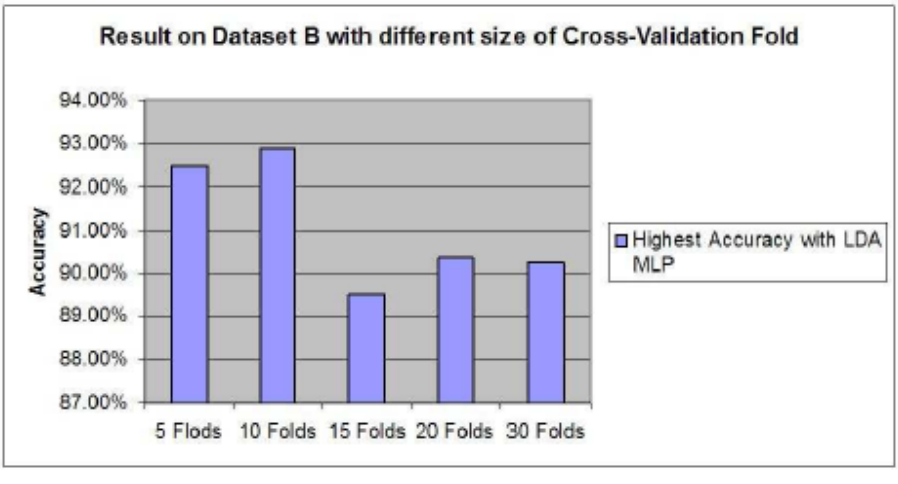


Fig. 8. Accuracy difference in different folds LDA-MLP with dataset B

The result is fluctuating on size of folds. It shows 10 folds provided better result in compare to other applied folds. The best result is around 93% with 10 folds and 15 folds cross-validation given poor result that is around 89% accuracy. However, figure 9 shows the result in dataset C with different size of cross-validation.

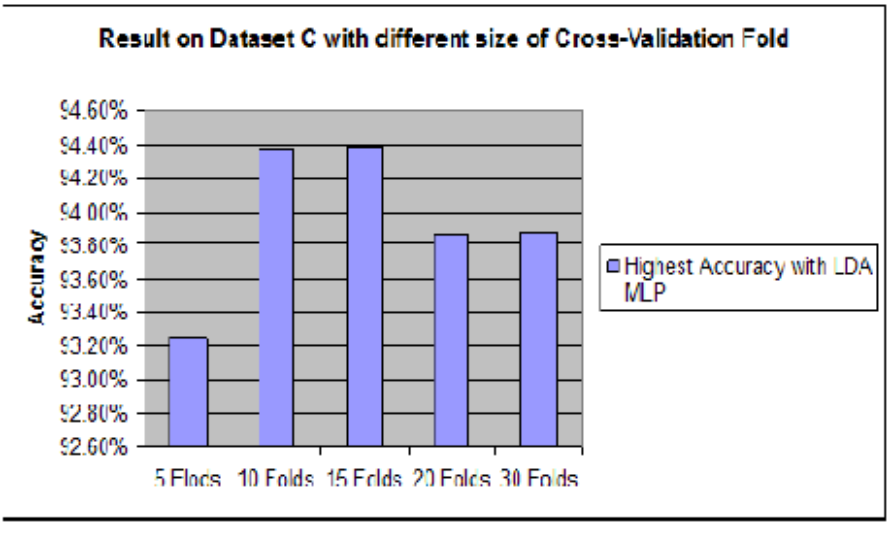


Fig. 9. Accuracy difference in different folds LDA-MLP with Dataset C

With dataset C, 15 folds provided better result which is around 94.5% and 5 folds cross-validation provided 93.3% which the lowest accuracy in this experiment. The dimensions of the PCA and LDA features for examining the influence of different folds of cross-validation were 49 dimensions. The final set of experiments involved influence of different dimensions of PCA and LDA features, as dimensionality can affect the speed of the recognition system. For this set of experiments, we fixed number of folds for cross validation to 10 folds, as this seems to be optimal from Figure 8 and 9. The results are shown in Table 1 here for SMO classifier. As can seen in Table 1, the best recognition accuracy is achieved. for dataset C with LDA features with 40 dimensions and is of the order of 93.88%. The LDA features perform better with lesser dimensions as compared to PCA features.

To summarize our experimental evolution we can say that; performance depends on different parameters, such as type of features, type of classifiers, dimensionality of the features and number of cross- validation folds used. As can be seen, for dataset C (which has been captured in an infrared camera), the linear discriminant analysis (LDA) - Multilayer Percertron (MLP) - 15 folds cross-validation turns to be the best combination.

Table 1. Influence of dimensionality of PCA-LDA features on the accuracy

Dataset	Features	Dimensions	Accuracy (%)
Dataset B	PCA	10	4.39
Dataset B	PCA	20	10.38
Dataset B	PCA	30	19.88
Dataset B	PCA	40	37.76
Dataset B	PCA	49	52.63
Dataset B	LDA	10	2.13
Dataset B	LDA	20	17.88
Dataset B	LDA	30	39.63
Dataset B	LDA	40	54.66
Dataset B	LDA	49	68.13
Dataset C	PCA	10	76
Dataset C	PCA	20	88
Dataset C	PCA	30	88
Dataset C	PCA	40	90.63
Dataset C	PCA	49	90.68
Dataset C	PCA	10	91.25
Dataset C	LDA	20	86.88
Dataset C	LDA	30	93.23
Dataset C	LDA	40	93.88
Dataset B	LDA	49	93.38

5 Conclusions and Further Plan

In this paper we proposed a novel person identification approach based on using different type of datasets based on visible and infrared gait images with side or profile views, and set of feature extraction and classification techniques Basically we used all data which are in 90 degree view angle. Because of 90 degree view angle, all of our expected traits (ear, side face, and gait) had clear view. Combination of dimensionality reduction approach PCA-LDA with different classifiers we received promising results. Significant outcome of this experiment is; for surveillance applications infrared camera will work better then normal video camera and that is what we proved by our results.

Acknowledgement. We are very much pleased and thankful to publicly available tools and databases for this paper. We would like to convey our gratitude to Institute of Automation, Chinese Academy of Sciences, for their excellent Database called "CASIA gait database". We also grateful to Machine Learning Group at University of Waikato for their "Weka" machine learning software. This is really massive software especially in machine learning area. We received expected outcome by utilizing both CASIA database and WEKA machine learning software.

References

1. Zheng, S.: CASIA Gait Database collected by Institute of Automation. Chinese Academy of Sciences, CASIA Gait Database, http://www.sinobiometrics.com
2. Weka, machine learning software, Machine Learning Group at University of Waikato, http://www.cs.waikato.ac.nz
3. Bringer, J., Chabanne, H.: Biometric Identification Paradigm Towards Privacy and Confidentiality Protection. In: Nichols, E.R. (ed.) Biometric: Theory, Application and Issues, pp. 123–141 (2011)
4. Jain, A.K.: Next Generation Biometrics, Department of Computer Science & Engineering, Michigan State University, Department of Brain & Cognitive Engineering, Korea University (2009)
5. Yampolskiy, R.V., Govindaraja, V.: Taxonomy of Behavioral Biometrics. In: Behavioral Biometrics for Human Identification, pp. 1–43 (2010)
6. Berretti, S., Bimbo, A., Pala, P.: 3D face recognaition using isogeodesic stripes. IEEE Transaction on Pattern Analysis and Machine Intelligence 32(12) (2010)
7. Shlizerman, I.K., Basri, R.: 3D Face Reconstruction from a Single Image Using a Single Reference Face Shape. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(2) (2011)
8. Human Face Recognition, Advantages and disadvantages of 3D face recognition, http://www.tutorial.freehost7.com/human_face_recognition/biometrics_and_human_biometrics.htm
9. Fingerprint Identification Technology, Principles of fingerprint biometrics, http://www.biometricvision.com
10. Feng, J., Jain, A.K.: Fingerprint alteration. Submitted to IEEE TIFS (2009)

11. Bengio, S., Mariethoz, J.: Biometric Person Authentication IS A Multiple Classifier Problem, Google Inc, Mountain View, CA, USA, bengio@google.com, IDIAP Research Institute, Martigny, Switzerland, marietho@idiap.ch
12. Biometric technology, Hand Geometry Identification Technology, http://www.biometricvision.com
13. Yuan, L., Mu, Z., Xu, Z.: Using Ear Biometrics for Personal Recognition, School of Information Engineering, Univ. of Science and Technology Beijing. Beijing 100083, yuanli64@hotmail.com
14. Ross, A., Jain, A.K.: Information fusion in biometrics. Pattern Recognition Letters 24, 2115–2125 (2003)
15. Jain, A.K., Hong, L., Kulkarni, Y.: A multimodal biometric system using fingerprints, face and speech. In: 2nd Int'l Conf. AVBPA, pp. 182–187 (1999)
16. Wang, Y., Tan, T., Jain, A.K.: Combining Face and iris Biometrics for Identity Verification. In: Kittler, J., Nixon, M.S. (eds.) AVBPA 2003. LNCS, vol. 2688, pp. 805–813. Springer, Heidelberg (2003)
17. Chang, K., et al.: Comparison and Combination of Ear and Face Images in Appearance-Based Biometrics. IEEE Trans. PAMI 25, 1160–1165 (2003)
18. Kittler, J., et al.: On combining classifiers. IEEE Trans. Pattern Anal. Mach. Intell. 20, 226–239 (1998)
19. Smith, L.I.: A tutorial on Principal Components Analysis (2002)
20. Linear discriminant analysis, Wikipedia, http://www.wikipedia.org
21. Platt, J.C.: Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines, Microsoft Research, jplatt@microsoft.com, Technical Report MSR-TR-98-14 (1998)
22. Cross-Validation (Statistics), http://www.wikipedia.org

Detecting Actions by Integrating Sequential Symbolic and Sub-symbolic Information in Human Activity Recognition

Michael Glodek, Friedhelm Schwenker, and Günther Palm

Institute of Neural Information Processing, Ulm University,
James Frank Ring 1, Ulm, Germany

Abstract. Detecting human activities is a challenging field for sequential algorithms in machine learning and several approaches have already been proposed. One approach is to make use of the hierarchical structure of the activities to be classified by subdividing them into more elementary actions [12]. Alternatively the fusing of additional context information has been investigated to obtain a more meaningful feature space [10]. Within this work both approaches are pursued by utilizing the layered architecture proposed by Oliver et al. [13] with the conditioned hidden Markov model (CHMM) [8]. The model is evaluated using a dataset containing sequential sub-symbolic information (i.e. the position of body parts) and symbolic information (i.e. the detected object the person interacts with). The results outperform the classical approach making no use of the additional symbolic information.

Keywords: Action Recognition, Markov Models, Machine Learning, Layered Architecture.

1 Introduction

Human activity recognition is an emerging field of research due to the challenges in variety, complexity and multi-modality of the classes to be detected. The achievements give insights in handling sparse occurrences in modalities which are also of interest in other related research fields such as affective computing in which classes are often only implicitly observable. Since the recognition can be approached from different directions the methods of resolution are as well very diverse [18,19]. For instance, Park et al. [14] propose the application of hierarchical Bayesian networks. Static body poses are estimated by a Bayesian network using features of detected body parts. Based on these poses basic actions are recognized utilizing a dynamic Bayesian network (e.g. the location and the stretch of the arm). A two-person interaction is then detected by additionally incorporating domain knowledge about relative poses and event causality. In summary, a complete hierarchical architecture composed of multiple layers using static and sequential models, as well as a symbolic layer to handle the large variety on the top-most layer is proposed. However, no objects are involved in this study and the temporal sequence of the observations is only partially used. Nguyen et al. [12] detect complex behaviors based on

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 394–404, 2012.

composed primitive behaviors which are themselves detected based on movement trajectories. The behaviors are detected using an extended hierarchical hidden Markov model (HHMM) [5] which removes the limitation to tree-like structures and enables the sharing of structures like primitive behavior patterns. The extension increases the expressiveness, but using a holistic architecture of the HHMM has the downside that the complete model needs to be retrained in case the setting changes or an additional class is introduced. Furthermore, it is difficult to add new information to the model in order to improve the detection of levels being high within the hierarchy. Sung et al. [17] detects human activity by dividing them into sub-activities. The sub-activities are estimated using a Gaussian mixture model and then used to train a maximum-entropy Markov model which captures the intuition that activities are composed of consecutive sub-activities. The proposed approach is in many aspects similar to the work presented. However, Sung et al. [17] make no use of the temporal characteristics of the sub-activities and do not train a generic model for a set of sub-activities which occur in different distinct activities (e.g. pick up). Furthermore, no symbolic context such as the object being manipulated is taken into account. Ben-Arie et al. [1] recognize activities by combing votes for an activity casted by each body part (i.e. all four limbs and torso). The intermediate results of the body parts are then combined using sequencing. The approach can easily be extended by new activities since the lower layer need not to be re-trained. However, the architecture and fusion approach is rather basic. Oliver et al. [13] propose a layered hidden Markov model (HMM) architecture to detect activities such as *phone conversation*, *face to face conversation* or *presentation* based on atomic events detected by utilizing the modalities: audio (e.g. human speech, phone ring), video (e.g. nobody present, one person present) and computer activity (e.g. keyboard or mouse used). Within this architecture the outputs of the classifiers from the first layer are fed into the consecutive layer of classifiers. Oliver et al. [13] utilize HMM to detect the classes within a layer and pass crisp class assignments to the next layer. Due to the layered architecture every consecutive layer has a coarser time granularity and increased level of abstraction than the previous layer. Lower layers can be re-trained to adapt to a new scene while keeping the more abstract upper layers unchanged. Furthermore, new activities can be added by training on the already trained lower layers.

The presented work studies the layered architecture proposed by Oliver et al. [13] but focuses on the first layer in which elementary actions are recognized. The action recognition follows a new approach in which sequential sub-symbolic information (i.e the position of body parts) is integrated with the symbolic information (i.e the recognized object) by a conditioned HMM (CHMM) [8]. The name of the CHMM is originated from conditioning to the set of nodes considered as labels. However, in the presented study only the likelihood of the model is of interest and, therefore, the conditioning is not performed. The CHMM is inspired by the latent-dynamic conditional random field (LDCRF) proposed by Morency et al. [11]. In contrast to the LDCRF the CHMM is based on a Bayesian network. Due to the missing conditioning, the CHMM can be related to the coupled HMM [3]. In the formalism of the coupled HMM a hidden state at time

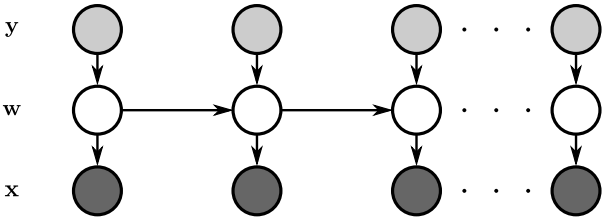


Fig. 1. Graphical model of the CHMM. The gray nodes are illustrating the random variable **y** influencing the hidden variables **w**, which in turn influencing the observations **X** shown as dark gray nodes.

step t of one HMM chain is additionally influenced by the hidden state in time step $t-1$ of another HMM chain. However, in comparison to coupled HMM, the CHMM does not use multiple chains of HMM. Instead, the hidden state of this model is influenced by an outer cause which is realized by the outcome of an independent classifier.

The application of the CHMM is studied using a dataset recorded with the Kinect™ camera[1] and labeled with six actions on the first layer and four activities on the second layer. The sub-symbolic information is extracted from a skeleton fitted in real-time by the camera while the symbolic information is obtained by an object recognition performed on the RGB image.

We proceed as follows. Section 2 introduces the concept of the CHMM. The dataset and the arising problem setting are explained in Section 3 while the results achieved are summarized in Section 4. Finally, Section 5 draws conclusions from the results and gives an outlook on future work.

2 Methods

The CHMM extends the HMM by the assumption that the selection of the hidden states are influenced by an outer cause which is the symbolic information given by an external classifier [8,7]. The Markov chain of the CHMM is shown in Figure 1 and is composed of a sequence of hidden random variables $\mathbf{w} = (\mathsf{w}_1, \ldots, \mathsf{w}_T)$ influencing the sequence of observations $\mathbf{X}$. The hidden variables themselves are influenced by a sequence of random variables $\mathbf{y}$ which are modeling the aforementioned causes inducing the selection of the hidden random variables. The model likelihood of the sequence $\mathbf{X}$ and $\mathbf{y}$ is given by

$$p(\mathbf{X}|\mathbf{y},\lambda) = \sum_{\mathbf{w}\in\mathcal{W}} p(\mathsf{w}_1 = w_1|\mathsf{y}_1 = y_1, \boldsymbol{\pi}) \cdot \Big(\prod_{t=1}^{T} p(\mathbf{x}_t = \mathbf{x}_t|\mathsf{w}_t = w_t, \theta) \Big) \cdot \Big(\prod_{t=2}^{T} p(\mathsf{w}_t = w_t|\mathsf{w}_{t-1} = w_{t-1}, \mathsf{y}_t = y_t, \mathbf{A}) \Big),$$

[1] The Kinect™ camera is a novel input device for video games by Microsoft®. A more detailed description of the camera's properties can be found at http://www.xbox.com/en-US/Kinect (24/10/2011).

where the set $\mathcal{W}$ contains all possible states of $\mathbf{w}$ and the set $\lambda = \{\boldsymbol{\pi}, \mathbf{A}, \theta\}$ holds the set of parameters. The elements of the matrix $\boldsymbol{\pi} \in \mathbb{R}^{|\mathcal{W}|\times|\mathcal{Y}|}$ correspond to the probability to enter a hidden state in the first time step depending on the random variable y_1. The transition probability is parameterized by the matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{W}|\times|\mathcal{W}|\times|\mathcal{Y}|}$ in which the probability to be in hidden state at the time step t depends on the former hidden state w_{t-1} and the current discrete state y_t. The literature suggests different ways to model continuous distributions. However, among the parameterized distributions the Gaussian mixture model (GMM) is still frequently used. Within this study we therefore focus on modeling the emission probability $p(\mathbf{x}_t = \mathbf{x}_t|\mathsf{w}_t = j, \theta)$ by a GMM such that every hidden state $j \in \mathcal{W}_t$ is modeled by a GMM having K mixture components $\theta_j = \left\{\{\boldsymbol{\phi}_{j,k}\}_{k=1}^K, \{\boldsymbol{\mu}_{j,k}\}_{k=1}^K, \{\boldsymbol{\Sigma}_{j,k}\}_{k=1}^K\right\}$. The elements of the set θ_j contains the mixing components $\{\boldsymbol{\phi}_{j,k}\}_{k=1}^K$, the means $\{\boldsymbol{\mu}_{j,k}\}_{k=1}^K$ and the covariance matrices $\{\boldsymbol{\Sigma}_{j,k}\}_{k=1}^K$. The parameters of the classic HMM are iteratively re-estimated using the well-known expectation-maximization (EM) algorithm [9,2]. The CHMM parameters are estimated in the same manner using a slightly modified EM algorithm. Since a detailed description of the algorithm would be out of scope, only the essential evaluation of forward-backward variables is addressed. For each time step t and hidden state j the forward variable $\alpha_{t,\mathbf{y}}(j)$ and backward variable $\beta_{t,\mathbf{y}}(j)$ is recursively determined based on the given sequence $\mathbf{y}$ using the equation

$$
\begin{aligned}
\alpha_{t,\mathbf{y}}(j) &= p(\mathbf{X}_{1..t}, \mathsf{w}_t = j|\mathbf{y}_{1..t}) \\
&= p(\mathbf{x}_t|\mathsf{w}_t = j)p(\mathbf{X}_{1..t-1}, \mathsf{w}_t = j|\mathbf{y}_{1..t}) \\
&= p(\mathbf{x}_t|\mathsf{w}_t = j) \sum_{i\in\mathcal{W}_{t-1}} p(\mathbf{X}_{1..t-1}, \mathsf{w}_{t-1} = i, \mathsf{w}_t = j|\mathbf{y}_{1..t}) \\
&= p(\mathbf{x}_t|\mathsf{w}_t = j) \sum_{i\in\mathcal{W}_{t-1}} p(\mathbf{X}_{1..t-1}, \mathsf{w}_{t-1} = i|\mathbf{y}_{1..t-1})p(\mathsf{w}_t = j|\mathsf{w}_{t-1} = i, y_t) \\
&= p(\mathbf{x}_t|\mathsf{w}_t = j) \sum_{i\in\mathcal{W}_{t-1}} \alpha_{t-1,\mathbf{y}}(i)\ p(\mathsf{w}_t = j|\mathsf{w}_{t-1} = i, y_t)
\end{aligned}
$$

and

$$
\begin{aligned}
\beta_{t-1,\mathbf{y}}(j) &= p(\mathbf{X}_{t..T}|\mathsf{w}_{t-1} = j, \mathbf{y}_{(t)..T}) \\
&= \sum_{i\in\mathcal{W}_t} p(\mathbf{X}_{t..T}, \mathsf{w}_t = i|\mathsf{w}_{t-1} = j, \mathbf{y}_{t..T}) \\
&= \sum_{i\in\mathcal{W}_t} p(\mathbf{X}_{(t+1)..T}|\mathsf{w}_t = i, \mathbf{y}_{(t+1)..T})p(\mathbf{x}_t|\mathsf{w}_t = i)p(\mathsf{w}_t = i|\mathsf{w}_{t-1} = j, y_t) \\
&= \sum_{i\in\mathcal{W}_t} \beta_{t,\mathbf{y}}(i)p(\mathbf{x}_t|\mathsf{w}_t = i)p(\mathsf{w}_t = i|\mathsf{w}_{t-1} = j, y_t).
\end{aligned}
$$

The start and termination of the recursions are given by

$$\alpha_{1,\mathbf{y}}(j) = p(\mathbf{x}_1|\mathsf{w}_1 = j)p(\mathsf{w}_1 = j|y_1)$$
$$\beta_{T,\mathbf{y}}(j) = 1$$
$$\beta_{0,\mathbf{y}}(j) = \sum_{i \in \mathcal{W}_1} \beta_{1,\mathbf{y}}(j)p(\mathbf{x}_1|\mathsf{w}_1 = i)p(\mathsf{w}_1 = i|y_1).$$

To infer the probability for the sequences $(\mathbf{X}, \mathbf{y})$ the sum over the forward variable $\alpha_{T,\mathbf{y}}(i)$ at time step T can be evaluated

$$p(\mathbf{X}|\mathbf{y}) = \sum_{i \in \mathcal{W}_T} \alpha_{T,\mathbf{y}}(i)$$

where $\alpha_{t,\mathbf{y}}(i) := p(\mathsf{w}_t = i, \mathbf{X}|\mathbf{y})$ and $\mathcal{W}_t$ contains all possible states of the hidden state at time step t. Since the dataset used is rather small, the CHMM has further been modified by an additional independence assumption. The conditioned probability of $p(\mathsf{w}_t = w_t|\mathsf{w}_{t-1} = w_{t-1}, \mathsf{y}_t = y_t, \mathbf{A})$ is divided into two matrices $\mathbf{A} \in \mathbb{R}^{|\mathcal{W}|\times|\mathcal{W}|}$ and $\mathbf{C} \in \mathbb{R}^{|\mathcal{W}|\times|\mathcal{Y}|}$ such that fewer parameters need to be learned. Furthermore, the parameter q is introduced which forces the parameters of the matrix $\mathbf{C}$ to be more similar using the update formula

$$\hat{C}_{ij} = \frac{C_{ij}^q}{\sum_{k \in \mathsf{w}} C_{kj}^q}.$$

where $i \in \mathcal{W}$ and $j \in \mathcal{Y}$. Choosing $q = 1$ will let the matrix $\mathbf{C}$ unchanged while choosing $0 < q < 1$ forces the parameters to a uniform distribution. In case the random variable y_t has only one possible state the CHMM is equivalent to the HMM. If more than one state is given for y_t, the CHMM can model complexer distributions having for instance additional context information. Hence, the CHMM promises to have a more robust behavior to estimate the likelihood of the observation sequence $\mathbf{X}$. However, the increased flexibility to adapt to these distributions comes at the expense of additional parameters which need to be learned from a larger set of trainings-data.

3 Dataset

In the given setting, activities always involve an object and can be subdivided into the actions: pick up object (PU_1), manipulate or hold the object (MA_1), move object to head (TH_1), use object close at the head position (HP_1), move object away from head (FH_1), lay object back to the table (LB_1). For instance, the activity of eating an apple is composed of the actions: pick up the apple from the table, moving the apple towards the head, take a bite of the apple, move the apple from the head and lay the apple back to the table.

The data acquisition was accomplished using the Kinect™ camera which is capable of collecting a RGB image, a corresponding depth map, a bit mask of tracked users and a fitted skeletons for up to two users. The video data has

been re-sampled to 10Hz because of varying frame rates. Based on the skeleton delivered by the camera, a graph is extracted using the position of the head, shoulders, torso center, hips, elbows and hands, as shown in Figure 2. The feature vector is then extracted from the graph based on the euclidean distances between the body parts. Figure 2 illustrates the usage of the hand positions to extract sub-images of the object the person holds in his hand. Features of these sub-images are obtained by subdividing the image into a 2×2 grid and calculating for each bag an orientation histogram having 45° bins [6].

Based on these features the discrimination of many actions will in principle be possible. However, two questions arise: can the object detection help to enhance the classification result of the actions, and can the confusion between similar actions be reduced? The first question can be exemplified by assuming that the object at hand is a book. In this case it is rather unlikely that the object will be moved towards the head. Using the information of the object detection therefore can prevent a misclassification. The second question, which will not be addressed in the presented work, is based on the fact that some actions might look similar but are only likely for certain objects. For instance, the attempt of filling a cup standing on the table might be confused with laying back an object since both actions are comprise of a movement towards the table. With the knowledge of holding for instance an apple, such an action is unlikely to occur.

The dataset contains four different activities which are decomposed into six actions.

Based on the activities, the objects *apple*, *cup*, *book*, *phone* and *empty hand* are labeled. The object class *empty hand* is included for later studies. The recorded activities in the given scenario are only requiring one hand to interact with the object. The samples of the class *empty hand* are therefore given by the hand which is not interacting with the labeled object.

Figure 3 shows the first layer of the architecture. The left-hand side illustrates the extraction of the symbolic and sub-symbolic sequences $\mathbf{y}$ and $\mathbf{X}$. The sub-images are classified by ν support vector machines (ν-SVM) with probabilistic outputs [15,16]. They are trained in a one-vs-one manner with a subsequent linear mapping to the five object classes. The crisp results of the object recognition are concatenated to form the sequence $\mathbf{y}$. The skeletons delivered by the camera are processed as described above and concatenated to the sequence $\mathbf{X}$. The right-hand side shows the CHMM which are trained for each action such that the

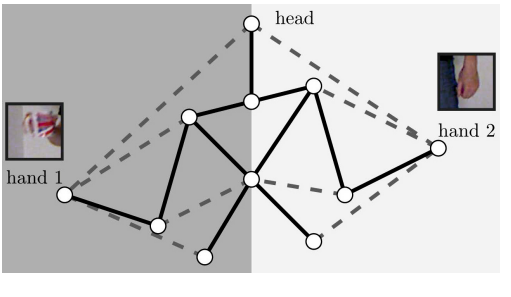


Fig. 2. Schema of the upper part of the skeleton as detected by the Kinect camera. For each hand a graph (dashed lines) is extracted with an additional sub-image based on the position of the hand.

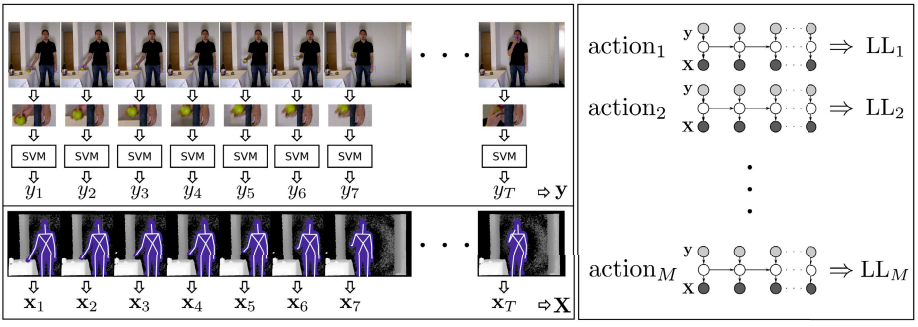


Fig. 3. Feature extraction and utilization of the CHMM. The left-hand side shows the feature extraction. The upper-left part illustrates the extraction of the symbolic features **y**, while the lower part shows how corresponding sub-symbolic features are extracted from the skeleton. The right-hand side shows the phalanx of CHMM processing the sequences. See text for a more details.

decision of the action to be observed is done by comparing the likelihoods (a uniform prior is assumed for all actions).

4 Results

First the results of the object recognition are presented. Subsequently, the results of the action recognition are summarized and compared to an alternative, classical approach. The classification is evaluated using a 3-fold cross-validation based on parameters optimized by an inner 3-fold cross-validation.

4.1 Object Recognition

The frame-wise F_1-measures, recall and precision per class of the testing set are shown in Table 1. The overall per frame error rate of the five-class problem is 33.1%. The F_1-measures indicate that there is no class which is not recognized. The class performing best is *empty hand*, followed by *apple* and *cup*. The good result of the class *empty hand* might be related to the high number of samples and the simple appearance of the pattern since it is in general not moving. The class *phone* has the worst performance which is related to the similarity to the

Table 1. Frame-wise object recognition. Overall error rate achieved is 33.1%. All results in percent with standard deviation in brackets.

Object	Apple	Cup	Book	Phone	Empty hand
↑F_1	61.60 (2.3)	58.40 (5.1)	49.20 (7.5)	36.10 (3.8)	82.00 (3.1)
↑Recall	56.5 (2.2)	62.6 (6.7)	69.0 (3.0)	31.1 (6.0)	80.4 (4.4)
↑Precision	68.1 (5.7)	54.9 (4.6)	38.5 (8.3)	44.2 (4.0)	83.8 (4.6)

class *book*. The object recognition for a complete action sequence is enhanced by multiplying the probability of each frame decision. The fusion improves the error rate to 26.2%. The SVM results for the training and development sets are calculated in a leave-one-out manner in order to avoid over-fitting.

4.2 Action Recognition

Two different experiments have been performed for comparison. Within the first experiment the architecture using the CHMM, as already explained in the previous sections, is evaluated. The second experiment replaces the CHMM by a classical HMM such that the action recognition is done with disregard of the additional object information.

The parameter search has shown that a CHMM using 5 hidden states and 4 mixture components for the Gaussian mixture model (GMM) and the parameter $p = 10^{-3}$ performed best. The results are listed in Table 2a. Each recognition result is divided into the fields: action correct (AC), action correct/object correct (AC/OC), action wrong/object correct (AW/OC), action correct/object wrong (AC/OW), action wrong/object wrong (AW/OW) and object correct (OC). The fields AC and OC are equivalent to the accuracy of the corresponding recognition task. While the other shares give insight in the confusion between object and action recognition. Furthermore, the last column lists the per class F_1-measures. All actions are recognized with a high accuracy above 93.8% (the overall action recognition error rate is 3.3%).

Table 2. Action and object recognition. All results in percent with standard deviation in brackets. Highlighted results performed better than in the comparative experiment. A detailed description can be found in the text.

(a) Performance of the CHMM action and object recognition

Action	↑AC	↑AC/OC	↓AW/OC	↑AC/OW	↓AW/OW	↑OC	↑F_1
PU_1	**99.1**(1.6)	**77.7**(4.0)	**0.0**(0.0)	21.4(3.4)	0.9 (1.6)	77.7(4.0)	**96.70**(1.6)
MA_1	**95.2**(4.3)	**62.0**(11.4)	**2.4**(2.2)	**33.2**(15.1)	**2.4**(2.2)	64.4(13.2)	**95.60**(4.4)
TH_1	98.8(2.1)	68.4(5.9)	1.2(2.1)	30.3(5.7)	0.0(0.0)	69.7(5.7)	97.80(2.0)
HP_1	93.8(10.8)	80.6(26.8)	4.2(7.2)	13.2(16.2)	2.1(3.6)	84.7(19.7)	96.60(6.0)
FH_1	97.0(5.2)	66.2(16.6)	1.0(1.7)	30.7(12.0)	2.0(3.5)	67.2(15.0)	**97.20**(2.5)
LB_1	95.1(3.6)	79.5(5.9)	2.9(3.0)	15.6(2.3)	2.0(1.7)	82.4(3.7)	96.50(1.7)

(b) Performance of the HMM action and object recognition

Action	↑AC	↑AC/OC	↓AW/OC	↑AC/OW	↓AW/OW	↑OC	↑F_1
PU_1	99.0(1.7)	76.7(2.4)	1.0(1.7)	**22.3**(4.0)	**0.0**(0.0)	77.7 (4.0)	95.50(3.1)
MA_1	84.6(10.3)	57.8(14.9)	6.6(4.8)	26.7(11.5)	8.8(5.7)	64.4(13.2)	88.00(9.5)
TH_1	98.8(2.1)	68.4(5.9)	1.2(2.1)	30.3(5.7)	0.0(0.0)	69.7(5.7)	**98.30**(1.9)
HP_1	93.8(10.8)	80.6(26.8)	4.2(7.2)	13.2(16.2)	2.1(3.6)	84.7(19.7)	96.60(6.0)
FH_1	97.0(5.2)	66.2(16.6)	1.0(1.7)	30.7(12.0)	2.0(3.5)	67.2(15.0)	95.20(2.5)
LB_1	**96.1**(1.7)	**80.5**(4.1)	**1.9**(1.7)	15.6(5.0)	2.0(3.4)	82.4(3.7)	**97.10**(1.3)

The performance of the proposed architecture is compared with a classical approach. The CHMM is replaced by a plain HMM such that the action recognition does not make use of the knowledge about the objects the user interacts with. The best performing HMM used 4 hidden states and 3 mixture components for the GMM and achieved an overall action recognition error rate of 4.8% on the test set. The results are listed in Table 2b and are very close to the ones of the CHMM. According to accuracy AC and further stated by the F_1-measure the action MA_1 is performing worse compared to the CHMM. A closer look to Table 2a shows that the improvement of the CHMM was made in the shares AC/OC, but also in AC/OW. Since the CHMM learns the action with respect to the detected objects, the quality of the object recognition is less important than the consistent behavior. Therefore, the performance of the CHMM can be explained although the object recognition achieves only an accuracy of 64.4%. Furthermore, the CHMM outperforms the classical approach with respect to the F_1-measure in detecting the actions PU_1 and FH_1. However, while PU_1 achieves this results by means of an improved accuracy, the FH_1 action only benefits from the improved recall.

5 Conclusions

Human activity recognition is a challenging task in machine learning [14,12,1]. Promising approaches make use of additional information such as the context of the scene and hierarchical structuring of the classes to be observed. This work investigates actions as part of a larger framework to detect activities. The goal is to extend the action recognition such that not only sequential data of body part positions can be used but also the recognized class of object the subject is interacting with. In order to accomplish that task, the application of conditional hidden Markov models (CHMM) has been proposed [8]. The presented experiment shows an improvement compared to a classical approach using hidden Markov models (HMM) in which the additionally information is neglected. As already stated in Section 3, the additional information about the object is necessary to resolve the ambiguity of two similar actions (e.g. lay back an object and start filling a cup) and to prevent the detection of an action which is not typical for an object (e.g. moving a book towards the head). While the given dataset addresses the second question, the first ambiguity is not present in the current dataset. Future work, will aim at augmenting the dataset in order to address also the second question, to evaluate the architecture on the second layer and to introduce an additional layer of reasoning on top of the activity layer.

Acknowledgment. The presented work was developed within the Transregional Collaborative Research Centre SFB/TRR 62 “Companion-Technology for Cognitive Technical Systems” funded by the German Research Foundation (DFG).

Special thanks also to the authors of the *libsvm* framework for making their software available [4].

References

1. Ben-Arie, J., Wang, Z., Pandit, P., Rajaram, S.: Human activity recognition using multidimensional indexing. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(8), 1091–1104 (2002)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
3. Brand, M., Oliver, N., Pentland, A.: Coupled hidden markov models for complex action recognition. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), pp. 994–999. IEEE (1997)
4. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2(3), 27 (2011)
5. Fine, S., Singer, Y., Tishby, N.: The hierarchical hidden Markov model: Analysis and applications. Machine Learning 32(1), 41–62 (1998)
6. Freeman, W.T., Roth, M.: Orientation histograms for hand gesture recognition. Technical Report TR94-03, Mitsubishi Electrical Research Laboratories, originally published at the International Workshop on Automatic Face and Gesture Recognition (1995)
7. Glodek, M., Bigalke, L., Schels, M., Schwenker, F.: Incorporating uncertainty in a layered HMM architecture for human activity recognition. In: Proceedings of the Joint Workshop on Human Gesture and Behavior Understanding (J-HGBU), pp. 33–34. ACM (2011)
8. Glodek, M., Scherer, S., Schwenker, F.: Conditioned hidden markov model fusion for multimodal classification. In: Proceedings of the Annual Conference of the International Speech Communication Association (ISCA), Interspeech, pp. 2269–2272 (2011)
9. Koller, D., Friedman, N.: Probabilistic graphical models: Principles and techniques. The MIT Press (2009)
10. Moore, D.J., Essa, I.A., Hayes III., M.H.: Exploiting human actions and object context for recognition tasks. In: Proceedings of the Seventh International Conference on Computer Vision, vol. 1, pp. 80–86. IEEE (1999)
11. Morency, L.P., Quattoni, A., Darrell, T.: Latent-dynamic discriminative models for continuous gesture recognition. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8. IEEE (2007)
12. Nguyen, N.T., Phung, D.Q., Venkatesh, S., Bui, H.: Learning and detecting activities from movement trajectories using the hierarchical hidden Markov models. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 955–960. IEEE (2005)
13. Oliver, N., Garg, A., Horvitz, E.: Layered representations for learning and inferring office activity from multiple sensory channels. Special Issue on Event Detection in Video: Computer Vision and Image Understanding 96(2), 163–180 (2004)
14. Park, S., Aggarwal, J.K.: A hierarchical Bayesian network for event recognition of human actions and interactions. Journal of Multimedia Systems 10(2), 164–179 (2004)
15. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in Large Margin Classifiers 10(3), 61–74 (1999)
16. Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L.: New support vector algorithms. Journal of Neural computation 12(5), 1207–1245 (2000)

17. Sung, J., Ponce, C., Selman, B., Saxena, A.: Human activity detection from RGBD images. In: Proceedings of the AAAI Workshop on Pattern, Activity and Intent Recognition (2011)
18. Turaga, P., Chellappa, R., Subrahmanian, V.S., Udrea, O.: Machine recognition of human activities: A survey. IEEE Transactions on Circuits and Systems for Video Technology 18(11), 1473–1488 (2008)
19. Yu, C., Ballard, D.H.: A multimodal learning interface for grounding spoken language in sensory perceptions. ACM Transactions on Applied Perception 1(1), 57–80 (2004)

Computer Recognition of Facial Expressions of Emotion

Ewa Piątkowska[1] and Jerzy Martyna[2]

[1] Institute of Applied Computer Science, Jagiellonian University, Reymonta 4, 30-059 Cracow, Poland
[2] Institute of Computer Science, Faculty of Mathematics and Computer Science, Jagiellonian University, Prof. S. Łojasiewicza 6, 30-348 Cracow, Poland

Abstract. In this paper, we study the computer recognition of emotions involved in facial expressions. We propose a recognition system based on a support vector machine (SVM) system as a classifier for detecting of spontaneous emotions. Using a face detection algorithm we created the-face representation. Then, the face texture is encoded with Local Binary Patterns (LBP) and used as a feature set in emotion recognition. The presented classifier can be useful a.o. for aggression classification and automatic emotion exploration.

1 Introduction

Recognizing human emotions is a difficult task. The main reason is that people mainly rely on meaning recognition in daily communication. This is why speech recognition analysis has long-treated emotions contained in speech as fluctuations or noise. Although integrating speech recognition with emotional expressions provides important data towards explaining emotion space, progress in the field of machine perception and machine learning is to recognize emotions even if emotional expression is unconsciously mixed with the meaning of speech.

Moreover, machine learning approaches to facial expression recognition provide a unique opportunity to explore the compatibility or incompatibility of different theories of emotion representation. We addressed this issue in the present paper by comparing human behavioral data to a computer model that was trained to make a choice between basic expressions plus neutral faces. Thus emotional experience has been characterized as a set of discrete dimensions coding activation of specific states, such as defined human emotions. What is more, the idea of computer emotion recognition systems became eligible since Ekman *et al.* [3] introduced the theory, that there are six basic emotions universal for all people, despite of culture or nation. Those emotions are: joy, sadness, anger, fear, disgust and surprise.

Facial expressions recognition systems (FERS) are used in many fields starting with human computer interaction applications such as user mood driven software, through the medical support in pain detection or newborn children monitoring, ending with surveillance software, like drivers fatigue detection systems. Although

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 405–414, 2012.

emotion recognition is highly applicable, it is still very challenging problem in computer vision and much effort is put to improve the performance of FERS.

During the last two decades facial expression recognition was an active research topic in the area of computer science. The task of facial expression analysis involves three main phases: face detection, feature extraction and expression recognition. Many different approaches were proposed for each phase however in this paper we will only mention few of them to outline the basic idea of emotion recognition problem. More detailed description of work that was done can be found in [12], [17]. Face detection is the first stage of facial expression recognition system in which face is to be localized in the input image. Face can be perceived as an integral object - holistic approach or as a set of facial landmarks (eyes, mouth, nose) - analytic approach. An example of holistic face representation can be found in work by Huang [7] who introduced Point Distribution Model (PDM) based on mean geometry of human face. Moreover, Pantic and Rothkrantz [13] detected face by analysis of vertical and horizontal projections as well as skin color segmentation. Analytic face detection was used in work by Kobayashi and Hara [9] where face region was determined by iris location in monochrome images, and in Kimura and Yachida [8] who searched for eyes and mouth corners. Face could be also represented by a set of features for instance Haar-like features in Viola and Jones [15] algorithm or eigenfaces in the paper by Essa and Pentland [5] approach. Depending on face representation different feature extraction techniques are used to describe facial expression. In Pantic and Rothkrantz [13] face is regarded as a set of points and expression is measured by displacement of those points in the initial and peak image. Littlewort *et al.* [10] used Gabor wavelets to encode facial texture to describe emotion by appearance features. The Active Appearance Model, introduced by Edwards *et al.* [4], combines shape and texture information.

The final stage of the system is the classification task where expression described by a set of features is assigned to one of several classes. Emotions are usually categorized in terms of six basic emotions: anger, sadness, joy, surprise, disgust and fear, however sometimes there is seventh class included for neutral expression. Different machine learning methods can be used at this stage for instance: k-nearest neighbors [7], neural networks [9], expert systems [13], support vector machines (SVM) classifier [14] or boosting algorithms [10].

Main goal of the paper is to computer recognition of human emotions, paying special attention to the detection of aggression. Using the method introduced here, face representation was defined by means of the set of points, which are tracked in the video sequence. Thanks to using the special filter and the developed coding of a texture, an extraction of face features was made. Then, the use of the SVM classifier allowed us to detect human emotion in the examined face. The experimental results showed that the proposed method could analyze and classify the human emotion efficiently.

The paper is organized as follows: in the second section we provide some theoretical background about the method used in the proposed system. In the

Fig. 1. A flowchart of the emotion recognition system

third section, the results of numerical experiments and discussion for proposed method are given. Finally, conclusions are presented in section 4.

2 Emotion Recognition Problem

The proposed system for emotion recognition consists of three steps, namely: face detection, feature extraction and emotion recognition (Fig. 1). In the first step, the input image is processed in order to detect the occurrence of face and to create face model. Next step includes emotion representation with a set of well-suited features.

In our system, emotion is described by texture of face region. The last step of a system is concerned with classification task where detected emotion is assigned to one of the seven classes (neutral plus six basic emotions). The output of the system is properly labeled image.

2.1 Facial Detection

Face detection algorithm used in first stage of the system is based on work by Viola and Jones [15] who proposed a method for rapid object detection. In this approach image is represented by a set of rectangular Haar-like features (see Fig. 2), which are calculated by subtracting a sum of pixels covered by white

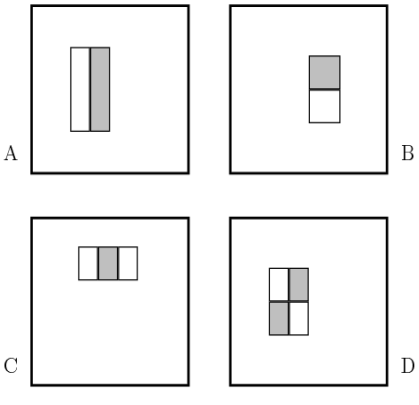


Fig. 2. Examples of Haar-like features

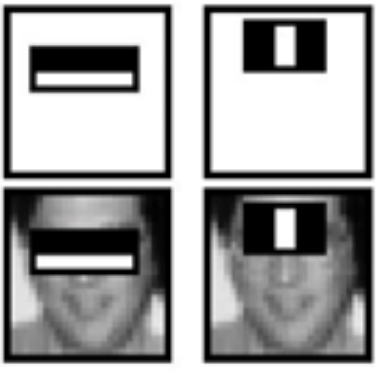


Fig. 3. Features detected on face region

rectangle from a sum of pixels covered by gray rectangle. Depending on the type of feature we can detect different elements in the image.

Two-rectangular features detect contrast between two vertically or horizontally adjacent regions. Three-rectangular features detect contrasted region placed between two similar regions and four-rectangular features detect similar regions placed diagonally (Fig. 3). To reduce computational effort put into feature calculation, input image is transformed into integral image in which each pixel is a sum of pixels above and to the left. Pixels in integral image are calculated by the formula

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \tag{1}$$

where $ii(x, y)$ is integral image and $i(x, y)$ is input image.

Using integral image improves the efficiency of the algorithm because the value of each rectangle (in terms of Haar-like features) requires up to four pixel references. Considering image representation, the number of features is much higher that number of pixels in the original image. However it was proven, that even small set of well-chosen features can build a strong classifier for object detection. Viola and Jones [15] used the Adaboost algorithm which iteratively selects the most discriminative feature to separate positive and negative examples in training set. The Viola and Jones [15] method is widely used in the area of face detection because of its efficiency and robustness.

2.2 Facial Expression Representation

Second stage of the proposed system is concerned with the task of emotion description by the set of well-chosen features. In this paper, we examine the significance of texture information in emotion recognition. To encode texture we use the Local Binary Patterns (LBP) method, proposed originally by Ojala et al. [11] and later extended by Ahonen [1] and Hadid [6]. The LBP method allows

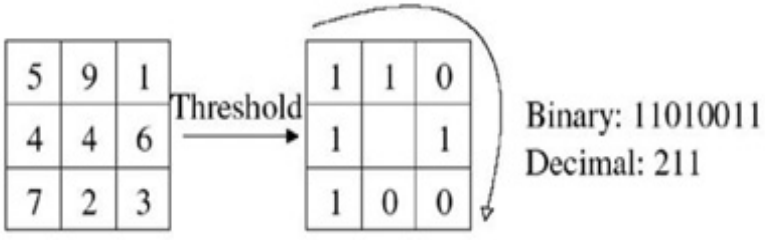


Fig. 4. LBP encoding scheme

for transforming the image into a representation, thanks to the application of the special operator which assigns the value on the basis of a value of the particular pixel P. The LBP operator is given by

$$\forall_{n \in N} t(n) = \begin{cases} 1, & n \leq P \\ 0, & n > P \end{cases} \tag{2}$$

where N is the number of pixels in the neighborhood. Values of pixels from the neighborhood, making a binary sequence, constitute a code which, when transformed into the decimal system, is assigned to a pixel.

The classical LBP operator has analyzed the neighborhood with dimensions 3×3, yet the relatively small size of the operator was its basic limitation. For the purposes of the features extraction, a new kind of LBP operator was adopted here as well as new dimensions of the neighborhood for the operator, e.g., a circular neighborhood with radius R and any number of pixels P.

As a result of LBP transformation carried out in this way, binary standards, coded local primitives of texture, the micro-patterns or textons also called. Examples of such micro-patterns are as follows: spot, spot/flat, line end, edge, corner. Sliding window is applied to the face region detected in the previous stage to transform it into LBP representation (see Fig. 4). Value of each pixel in the neighborhood is thresholded with the value of the central pixel in the sliding window.

Neighborhood pixels after thresholding are formed into a binary code and the decimal value of this code is assigned to the central pixel in the corresponding LBP image. Binary codes are called 'micro-textons' because they represent texture primitives such as curved edges, flat or convex areas.

2.3 Emotion Recognition

The last stage of the proposed system is emotion recognition in which the support vector machine (SVM) [14] with radial based kernel (RBF) function is used as a classifier. The support vector machine is a machine learning system which receives labeled training data and transforms it into higher dimensional feature space. Then separating hyperplane with respect to margin maximization is computed to determine the best separation between classes. The greatest advantage

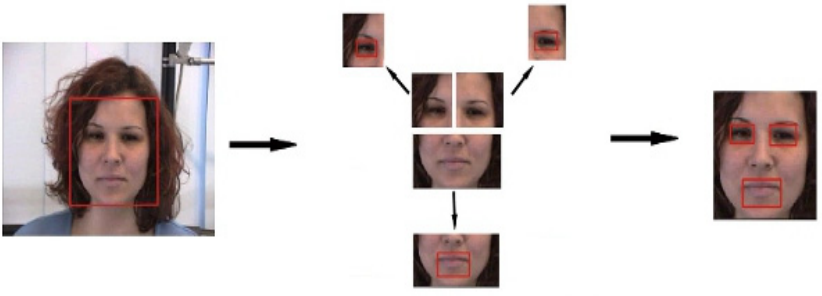


Fig. 5. Face detection and representation

of SVM is that even with small set of training data it has good performance in generalization.

3 Experimental Results

In the proposed system, the algorithm was used for face, eyes and mouth detection. After the acquisition, input image is being searched in order to detect face. In case of finding more than one face in the image, the biggest one is chosen for emotion analysis. Inversely, when there is no face in the image, further processing is omitted.

If the face is detected in the image (see Fig. 5), the classifiers for landmark detections are applied to find mouth, left and right eye. To improve algorithm efficiency, each landmark is searched in narrowed region, for instance, the left eye is searched only in upper left region of the detected face. Having locations of the face and its landmarks, we can form the face representation which is used in the next stage of the proposed system.

In the second stage of our system, we encode face texture in an analytic way to use also the spatial information about texture. In order to reduce the size of feature set, only the parts of the face which are highly involved in emotion expressions are encoded. Those parts are chin - mouth - cheeks and forehead eye regions (see Fig. 6). Particular face regions are normalized to the same size,

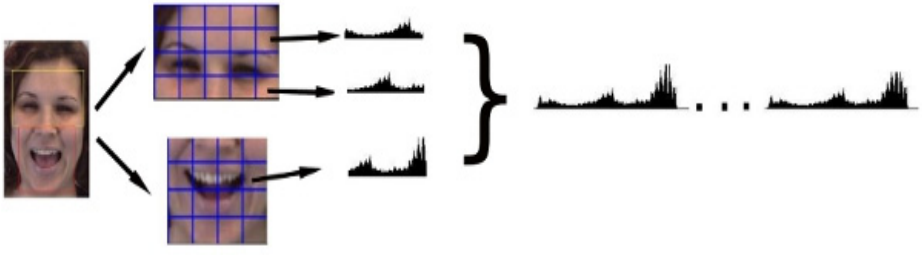


Fig. 6. Feature extraction scheme

Table 1. Confusion matrix for emotion classification

%	neutral	joy	sadness	surprise	anger	fear	disgust
neutral	61	0	33	0	0	6	0
joy	3	78	3	10	0	3	3
sadness	3	1	91	0	4	1	0
surprise	3	3	3	76	0	12	3
anger	0	2	36	0	51	4	7
fear	2	2	28	4	0	64	0
disgust	0	0	20	0	7	7	64

namely: 90×48 for upper region and for lower face region. Next, regions are divided into grids of sizes: 4×4 in the lower part, in the upper part. Each patch in the grid is then encoded with LBP operator.

We apply the basic version of LBP which uses 3×3 sliding window thus the range of possible codes are from 0 to 255. The original image texture is described by 256-bin histogram of the corresponding LBP image. Therefore, the feature set in our system consists of 36 histograms and particular emotion is described by 9216 features.

For the purpose of training we used the data contained in the Facial Expression and Emotion Database (FEED) [16] base (prepared in the framework of the FG-NET project at the University of Munich), which consists of MPEG video files with spontaneous emotions recorded. Database contains examples gathered from 18 subjects (9 female and 9 male). What is more, each subject shows particular emotion three times, thus first two samples are used for training while third is used for testing.

The proposed system was implemented in the environment of MATLAB using the Image Processing and Bioinformatics Toolboxes. It was trained with captured video frames in which the displayed emotion is very representative. Training set contains 675 images and testing set - 330. Both training and testing sets consist of images with seven states: neutral, surprise, fear, disgust, sadness, happiness and anger.

System's performance was measured with accuracy rate that is the proportion of properly classified images to all images in the test set. Proposed system recognizes emotions with accuracy rate of 71%. Additionally, the recognition results were presented by confusion matrix (see Table 1), which not only shows recognition accuracy of each emotion but also indicates the emotions which are commonly confused. The best recognition rate was obtained for sadness (91%), the worst one for anger (51%) that was usually confused with sadness.

A difficult task for applying computer recognition system of facial expression of emotion, is the detection of human aggression. According to the Facial Action Coding System Action Unit (FACS AU) recognition, given by Ekman [2], four AUs describe aggression, namely

Fig. 7. Facial expressions corresponding to human aggression

1) AU4 lowering eyebrows,
2) AU5 lifting upper eyelids,
3) AU7 tightening of eyelids,
4) AU23 tightening of mouth.

Action Units that describe facial expressions corresponding to aggression are presented in Fig. 7. The system was trained using 10 video sequences, demonstrating aggression and 10 sequences expressing other emotions. The set of tests contained 4 examples, 2 positive cases (aggression) and 2 negative cases (no aggression). In addition, the ability of the method for generalization was checked, using recordings of the persons, who have not been taken into account in the process of learning.

Effectiveness in detecting the expression is measured by the correctness recognition coefficient of the received results, namely

$$R = \frac{\text{number of correctly recognized examples}}{\text{number of all examples in the test}} \times 100\% \tag{3}$$

Presented results are in conformity with the tested configurations of the vector of features. For the analysis of the face points movements trajectory, a vector was defined, with dimensions of 1000 features, containing the information on the change in the face geometry in the determined time slice (frames). The correctness recognition coefficient obtained here is equal to 78%.

The texture of the face was acquired from the most representative frame of the sequence, where the presented emotion is in the phase of culmination. When coding the texture with the Gabor filters the system achieved a correctness recognition coefficient of 81%, while when using the LBP method 72%. Combining the two categories, or the methods describing both the geometry (in the dynamic take), and the face texture, the tested examples were classified with a correctness recognition coefficient of 82.5%. The overall results of the classification in different configurations of the vector of features are presented in Table 2.

The ability of the system for generalization was also tested, which here means the detection of aggression in examples which were not included in the test set (new persons). In this case vector of features was used which consisted of the

Table 2. The overall results of the classification in different configurations of features

Vector of features in time	Number of features	Correctness recognition coefficient
Geometrical features	1000	78%
Coding the texture with Gabor filter	432 000	81%
Coding the texture with LBP method	1280	72%
Geometrical features in time plus Gabor filter	433 000	80%
Geometrical features in time plus LBP method	2 280	85%

geometric-dynamic features as well as the LBP histograms. Two samples (negative and positive) were presented from four different persons. The system correctly classified all of them.

4 Conclusion

In this paper, we proposed fully automatic system for spontaneous emotion recognition. The system consists of 3 stages: face detection, feature extraction and classification. Firstly, acquired image is processed in order to detect occurrence of face and to create its representation with use of Viola and Jones [15] algorithm. Then, face region is encoded by Local Binary Patterns [11] method and passed to the SVM classifier for emotion recognition. Our main goal was to investigate the performance of Local Binary Patterns as a texture descriptor. The system can recognize emotions with accuracy rate of 71% therefore information encoded in facial texture is significant. The correctness recognition coefficient of aggression expression is equal to 78%.

In the future, we intend to improve the performance of classifier by reducing the feature set to contain the most discriminative features for particular emotion description. Some other classification methods could be considered as well.

References

1. Ahonen, T., Hadid, A., Pietikäinen, M.: Face Recognition with Local Binary Patterns. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 469–481. Springer, Heidelberg (2004)
2. Ekman, P., Friesen, W.: Facial Action Coding Systems: A Technique for the Measurement of Facial Movement. Consulting Psychologists Press, Palo Alto (1978)
3. Ekman, P., Huang, T., Sejnowski, T., Hager, J.: Final Report To NSF of the Planning Workshop on Facial Expression Understanding (1992), http://face-and-emotion.com/dataface/nsfrept/nsf_contents.html

4. Edwards, G.J., Cootes, T.F., Taylor, C.J.: Face Recognition Using Active Appearance Models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 581–695. Springer, Heidelberg (1998)
5. Essa, I., Pentland, A.: Coding, Analysis Interpretation, Recognition of Facial Expressions. IEEE Trans. Pattern Analysis and Machine Intelligence 19(7), 757–763 (1997)
6. Hadid, A., Pietikainen, M.T., Ahonen, T.: A Discriminative Feature Space for Detecting and Recognizing Faces. In: Proc. Computer Vision and Pattern Recognition, pp. 797–804 (2004)
7. Huang, C.L., Huang, Y.M.: Facial Expression Recognition Using Model-Based Feature Extraction and Action Parameters Classification. J. Visual Comm. and Image Representation 8(3), 278–290 (1997)
8. Kimura, S., Yachida, M.: Facial Expression Recognition and Its Degree Estimation. In: Proc. Computer Vision and Pattern Recognition, pp. 295–300 (1997)
9. Kobayashi, H., Hara, F.: Facial Interaction between Animated 3D Face Robot and Human Beings. In: Proc. Int'l Conf. Systems, Man, Cybernetics, pp. 3, 732–3, 737 (1997)
10. Littlewort, G.C., Bartlett, M.S., Chenu, J., Fasel, I., Kanda, T., Ishiguro, H., Movellan, J.R.: Towards Social Robots: Automatic Evaluation of Human-Robot Interaction by Face Detection and Expression Classification. In: Advances in Neural Information Processing Systems, vol. 16, pp. 1563–1570 (2004)
11. Ojala, T., Pietikainen, M., Harwood, D.: A Comparative Study of Texture Measures with Classification Based on Featured Distribution. Pattern Recognition 29(1), 51–59 (1996)
12. Pantic, M., Rothkrantz, L.: Automatic Analysis of Facial Expressions: The State of the Art. IEEE Trans. on Pattern Analysis and Machine Intelligence 22(12), 1424–1445 (2000)
13. Pantic, M., Rothkrantz, L.: Expert System for Automatic Analysis of Facial Expression. Image and Vision Computing Journal 18(11), 881–905 (2000)
14. Vapnik, V.N.: Statistical Learning Theory. John Wiley, New York (1998)
15. Viola, P., Jones, M.J.: Robust real-time object detection. International Journal of Computer Vision 57(2), 137–154 (2004)
16. Wallhoff, F.: Facial Expressions and Emotion Database, Technische Univesität München (2006), http://www.mmk.ei.tum.de/~waf/fgnet/feedtum.html
17. Zeng, Z., Pantic, M., Roisman, G.I., Huang, T.S.: A Survey of Affect Recognition Methods: Audio, Visual and Spontaneous Expressions. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(1), 39–58 (2009)

Outcome Prediction for Patients with Severe Traumatic Brain Injury Using Permutation Entropy Analysis of Electronic Vital Signs Data

Konstantinos Kalpakis[1], Shiming Yang[1], Peter F. Hu[2], Colin F. Mackenzie[2], Lynn G. Stansbury[2], Deborah M. Stein[2], and Thomas M. Scalea[2]

[1] Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250
{kalpakis,shiming1}@umbc.edu
[2] R Adams Cowley Shock Trauma Center; Shock Trauma and Anesthesiology Research Center, University of Maryland School of Medicine, Baltimore, MD 21201
{phu,lstansbury,dstein,tscalea}@umm.edu, cmack003@umaryland.edu

Abstract. Permutation entropy is computationally efficient, robust to noise, and effective to measure complexity. We used this technique to quantify the complexity of continuous vital signs recorded from patients with traumatic brain injury (TBI). Using permutation entropy calculated from early vital signs (initial 10~20% of patient hospital stay time), we built classifiers to predict in-hospital mortality, and mobility measured by 3-month Extended Glasgow Outcome Score (GOSE). Sixty patients with severe TBI produced a skewed dataset that we evaluated for accuracy, sensitivity and specificity. With early vital signs data, the overall prediction accuracy achieved 91.67% for mortality, and 76.67% for 3-month GOSE in testing datasets, using the leave-one-out cross validation. We also applied Receiver Operating Characteristic analysis to compare classifiers built from different learning methods. Those results support the applicability of permutation entropy in analyzing the dynamic behavior of biomedical time series for early prediction of mortality and long-term patient outcomes.

1 Introduction

Continuous vital signs (VS), such as heart rate (HR), blood pressure (BP), and oxygen saturation (SpO_2), among others, are sequential assessments of important physiological functions, providing basic evidence of patients' status. Because VS are an early-warning-system of physiologic perturbation, they are usually recorded hourly in the intensive care unit (ICU) setting. However, in most modern ICUs, automated electronic instrumentation is gathering these data continuously, and the massive quantities of high-quality data produced create both a challenge to store, analyze, and interpret and an opportunity to explore novel advanced analytic methods for predicting outcomes. Such predictive algorithms can support advanced instrumentation and decision-assist tools that have the potential to significantly improve clinical outcome for these very ill patients.

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 415–426, 2012.

A number of approaches have been suggested for utilization of VS data for prediction of adverse outcomes. These analyses attempt to discover the intrinsic patterns that characterize continuous, multivariate, time-series systems. One strategy is to embed the time series into higher dimensional space and then compute various entropies for the elements of the embedded time series. Conventional entropies such as Shannon entropy, Renyi entropy and Tsallis entropy can be calculated given the distribution of elements of the embedded time series. The Renyi entropy of a time series has been used to detect spatially varying multivariate relationships [9] and to study brain injuries [8] and heart rate variability [4,5]. The Tsallis entropy of the elements of a time series has been used to monitor brain injuries after cardiac arrests [2,24], and to improve the accuracy of gene regulatory networks inference [15].

The initial applications of ordinal pattern and permutation entropy demonstrate this to be very promising in quantifying and analyzing the dynamic behavior of biomedical and other time series. Introduced by Bandt and Pompe [1] in 2002, permutation entropy is a new measure of complexity of time series, and extracts qualitative information from non-linear time series. Examples include identifying temporal gene expression profiles [22], measuring the anesthetic drug effect from electroencephalograms (EEGs) [13,17], characterizing brainwave data of epileptic patients [14,18] and sleep EEGs [3,16], change detection in dynamic systems [5], and financial time series [23].

In this research effort, we have focused on VS classification. Given a number of VS sequences and their corresponding outcomes, we want to train a model to predict the outcome for a new sequence of VS. Permutation-based distribution estimation is used to calculate the Renyi entropy of the multivariate VS series, and to predict the in-hospital mortality and the three-month Extended Glasgow Outcomes Scale (GOSE). The early prediction is achieved by using the continuous automatically collected and stored electronic VS data collected in the first 10~20% of patient hospital stay time. To evaluate the results, we calculated accuracy, sensitivity, and specificity to quantify the performance of classifiers, especially for the imbalanced training/testing data sets. The Areas Under the Curve (AUCs) of the receiver operating characteristic (ROC) are also used to compare classifiers constructed by different learning methods. Using the first 3 days' VS of 5-minute time resolution, overall 91.67% prediction accuracy for mortality (classifier AUC= 0.84, $p < 0.001$), and 76.67% accuracy for 3-month GOSE (classifier AUC=0.71, $p = 0.001$) were achieved with the testing data set.

The remainder of this paper is organized as follows. In section 2, we briefly introduce the permutation entropy and the entropy map that we used for quantifying the characteristics of the dynamic system. In section 3, we describe the dataset and experiment design. We apply the permutation entropy to predict mortality and 3-month GOSE, and present experiment results, evaluated by accuracy and the area under the receiver operator characteristic (ROC) curve. Finally, in section 4, we provide discussions and summary.

2 Method

2.1 Ordinal Pattern and Permutation Entropy

That the physiological status of living things is dynamic but has identifiable and repeated patterns is assumed. Likewise, we assume that these patterns will be different in the healthy, injured, and/or ill individuals and that the patterns will be discernibly different from each other. For example, the VS of healthy individuals fall generally within a range of normal, whereas those of patients suffering from severe traumatic brain injury (TBI) have VS that fall outside of these norms. For instance, if the patient is also losing blood, blood pressure (BP) will fall. Heart rate (HR) increases to compensate for the decreased BP to ensure adequate circulation and oxygenation of the brain, and the increase in HR usually increases the BP, at least temporarily. If blood loss continues, BP falls, and clinicians will usually give fluid, including blood, to raise the BP and insure adequate oxygenation. These changing patterns of HR and BP are accompanied by changes in intracranial pressure (ICP), cerebral perfusion pressure (CPP), and so on.

Bandt and Pompe [1] suggested an approach to time series analysis in which they embedded a continuous timeseries as a symbolic sequence into another space, a process which they called "permutation entropy." One major ingredient of permutation entropy is the ordinal pattern. The ordinal pattern of a sequence of elements $x_1, \ldots, x_n$ is the permutation (re-arrangement) $\pi = (i_1, i_2, \ldots, i_n)$ that sorts the amplitude values in ascending order so that $x_{i_1} \leq x_{i_2} \leq \ldots \leq x_{i_n}$.

The order L permutation entropy of a timeseries $x_{1\ldots N}$ is calculated as follows. Let π_t be the ordinal pattern (i.e. the sorting permutation) for the segment of the timeseries under the sliding window of length L that ends at x_t, i.e. the subsequence $x_{t-L+1}, \ldots, x_t$. Let $S_L = \{\pi_k\}$ be the set of all those unique (alphabet) ordinal patterns π_t. To the timeseries $x_{1\ldots N}$ there corresponds the sequence $\langle \pi_t : t = L, \ldots, N - L + 1 \rangle$ of $N - L + 1$ ordinal patterns from the alphabet S_L. The entropy of this sequence of ordinal patterns is the permutation entropy of the timeseries $x_{1\ldots,N}$. For example, the Shannon permutation entropy is defined in equation (1),

$$H_L = -\sum_{k \in \mathcal{S}_L} P(\pi_k) \log(P(\pi_k)). \tag{1}$$

where $P(\pi_k)$ is the frequency of π_k in the sequence $\langle \pi_t \rangle$. In the work presented here, we use the Renyi entropy with parameter α of the sequence $\langle \pi_t \rangle$ defined as

$$R_L^{(\alpha)} = \frac{1}{1-\alpha} \log(\sum_{k \in \mathcal{S}_T} P(\pi_k)^\alpha). \tag{2}$$

The parameter α in the Renyi entropy acts as a selector of probabilities. It assigns almost equal weight to each possible probability when α is sufficiently close to zero. When α is larger, it puts more weights on higher probabilities. With this property, Renyi entropy can filter out the small probability events, and better capture the essence of the system.

2.2 Multivariate Time Series

In real applications, a single variable is generally insufficient to sketch the profile of complex dynamic systems, because they respond to multiple factors in a non-linear manner. For example, many VS are used to monitor TBI patient status – HR, systolic BP (SBP), SpO_2, ICP, CPP, etc. Suppose there are M variables. Given a window size L, vital signs within that window are viewed as one slice of size $M \times L$. Figure 1 demonstrates one example of finding ordinal patterns from a finite sequence of time series. Suppose that there are 3 vital signs ($M = 3$) available for inclusion: ICP, HR, and SBP. Let the window size be $L = 2$. Therefore, one slice constitutes 6 points, which means that we embed VS in a window of size 2 into a higher dimension 6. There are two choices to permute in a slice. The first one considers one slice as one bag. All values in this bag are sorted in an ascending order. For example, in Figure 1, slice 1 can be written linearly as the sequence: (ICP)12.36, 14.44; (HR)59.54, 59.48; (SBP)142.0, 138.6. Labeling each value 1~6, the values of this sequence can then be sorted into ascending order by applying a permutation $\langle 1\ 2\ 4\ 3\ 6\ 5 \rangle$. Another choice is to sort within each variable, then concatenate them. For the same example, if we sort ICP, HR, SBP in slice 1 separately, and concatenate their local permutation index, we obtain the pattern $\langle 1\ 2\ 2\ 1\ 2\ 1 \rangle$. The second method would help keep each variable isolated even if they may have similar range, and hence maintain the ordinal patterns from each variable.

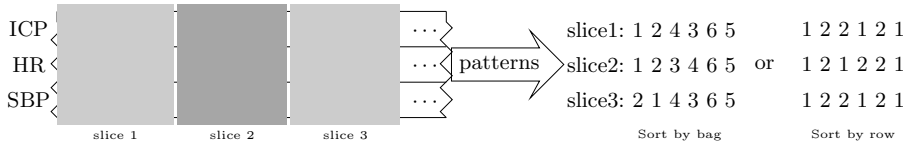


Fig. 1. Illustration of ordinal patterns built by permutation in two ways. The exemplary time series snippet comes from 6 points of 5-min smoothed data from one patient.

With the permutation entropy, we can construct a feature for each patient, and apply the supervised learning methods, such as decision tree, support vector machines, and discriminant analysis to build models from known outcomes. Furthermore, instead of using a single feature, we can create a family of features using different parameters in the entropy calculation. This strategy is more practical, for the following reasons. First, a family of features will leave the learning methods to select the most appropriate features with the data provided. This is always desired since we have limited knowledge to determine optimal window size and the parameter values for entropy calculation. Besides, using a different set of parameters may help us find more patterns that exist in other different spaces.

2.3 Evaluations

To evaluate results, not only the accuracy, but also the sensitivity, specificity and ROC analysis are utilized to compare performance of different classifiers. The ROC is a tool to depict the tradeoff between sensitivity and specificity. One major reason we adopt the ROC AUC for classifier comparison is that the dataset is skewed, and the ROC AUC is insensitive to the skewness of data sets [7]. Such property of ROC curves provides us a way to evaluate the classifiers without worrying about the datasets from which they were trained. Instead of using one single point, we can use the instance statistics to produce a full ROC curve by calculating the class label scores [7]. Provost *et al.* [20] described a method of calculating the ROC by assigning a score to each instance that reaches the leaf of the decision tree. That score is equal to the ration of positive class labels assigned to that leaf during training. Platt [19] suggested a way of estimating posterior probability from the output of a support vector machine by fitting a sigmoid function.

3 Experiments and Results

3.1 Data and Setup

After removal of patient identifiers, continuous, automated electronic VS data collected over the course of hospitalization from patients with severe TBI were analyzed using permutation entropy to predict in-hospital mortality and 3-month GOSE outcomes. These patient data were part of a larger study of prediction factors after severe TBI that is ongoing at the R Adams Cowley Shock Trauma Center, Baltimore, Maryland. Our dataset was collected during 2008 and 2009 from 60 sequentially admitted individuals, 9 female and 51 male, 8 of whom died while in hospital. The average duration of stay in hospital was 16 days (range, 1.5 to 53 days); 52 patients remained in the hospital longer than 1 week; and 27 patients stayed longer than 2 weeks. Among the 52 patients discharged from the hospital alive, follow-up interviews were carried out at 3 months post-discharge to assess functional outcomes of patients after treatment in terms of an 8-category scale [10]: dead, vegetative state, lower severe disability, upper severe disability, lower moderate disability, upper moderate disability, lower good recovery, and upper good recovery. Categories 1 to 4 are defined as "unfavorable" (value 1) and categories 5 to 8 as "favorable" outcomes (value 0). For 3-month GOSE in our dataset, 25 individuals had "favorable" outcome and 35 had "unfavorable" outcome, which, for our purposes, give a relatively balanced data set.

The raw, every-6-seconds data were preprocessed to deal with noise due to unstable attachment of sensors, patients' movement and missing values. To reduce the negative effect of noise, VS data were smoothed in a 5-minute tumbling window, as previously described [11]. In addition, gaps often occur at the start of the vital sign sensor placement or because patients were moved between hospital units (ICU, Operating Room, etc.). Table 1 shows the percentage of missing points of six selected VS. To utilize all information, we perform some impute

techniques, by using the k-nearest neighbors' average as the surrogate values. Another approach is to use the average values of the VS as the fill-in value.

Determining the optimal selection of VS with which to set up the experiment parameters can be difficult, that is, which values are optimal for the window size and the α range of the Renyi entropy. Therefore, our parameters were selected based on the following considerations. First, a group of VS that are frequently used in clinical diagnosis were chosen, such as ICP, CPP, SBP, SpO_2, etc. Those VS with the lowest percentage of missing points and missing data were selected to increase the chances of preserving more patterns, therefore more accurately characterizing the changing physiologic dynamics. A dataset was also tested for change of accuracy with and without removing a given vital sign. Correlated or dependent variables may be included in the dataset for ordinal pattern finding. However, it will not be redundant to include those variables when the relationship among those correlated variables are not order preserving. Hence, for simplicity, the rule of thumb is followed to select VS.

Using the above criteria and tests, a group of five VS were selected (see Table 1) and tested iteratively. Then the range of window size was selected for a block of vital signs among 3, 6, 12, equivalent to VS collection durations 15, of 30 and 60 minutes. In addition, the range for the Renyi entropy parameter α was selected as 0.1 to 2.0 with step size 0.01.

Table 1. Percentage of available values for selected vital signs

Vital signs	Percentage of available points			
	First 1 day	First 2 days	First 3 days	All
HR	90.07%	93.05%	94.53%	87.60%
SpO_2	87.04%	90.79%	92.38%	85.20%
SBP	88.71%	91.80%	93.23%	81.65%
SI=HR/SBP	88.71%	91.80%	93.23%	81.65%
ICP	68.63%	78.14%	79.81%	37.72%
CPP *	65.69%	74.51%	76.48%	36.45%

*not included due to its limited contribution to accuracy.

3.2 Prediction for Mortality and 3-Month GOSE

With the above setting, experiments were conducted to predict in-hospital mortality and 3-month GOSE. Since the sample size of 60 instances does not form a very large dataset, the leave-one-out cross validation method was used for training and testing.

For each individual patient, a collection of features based on entropy are built as follows. First, selected VS of a certain length (i.e. 3 days VS) are aligned by time and filled in for missing values with the k-nearest neighbor imputation method. Next, given a slice window size L, the VS within a moving window of length L are sorted in bag and are represented by permutations. Such collection

of permutations makes an alphabet, where the frequency of each "word" (permutation pattern) is calculated. With a vector of instantiation of parameter α in the equation (2), a set of entropy values are calculated for the window size L. Then the second step is repeated for different parameter values for L. So far, a group of new features are created for individual patients, which are different measurement of their physiological status complexity. With those features, various kinds of classification methods are applied to predict outcomes of clinical interest.

Tables 2a and 2b show confusion matrices and overall accuracy for predicting mortality and 3-month GOSE. The a priori knowledge is that 13.3% died in hospital, and 58.3% have unfavorable 3-month GOSE. Using early VS as defined above, a classification tree built upon permutation entropy achieved 62.50% in true positive rate (91.67% in overall accuracy) in predicting death, and 82.86% in true positive rate (76.67% in overall accuracy) in predicting unfavorable cases for 3-month GOSE, which are all higher than the a priori. This suggests that the permutation entropy is capable of classifying patients of different physiological status, and can handle imbalanced class distribution. On the other hand, the permutation entropy also demonstrates good performance of prediction using early VS. This has potential clinical importance in providing medical care providers with timely prognostic information.

Table 2. Confusion matrices for classification trees built upon features created by permutation entropy on the testing set

(a) In-hospital mortality

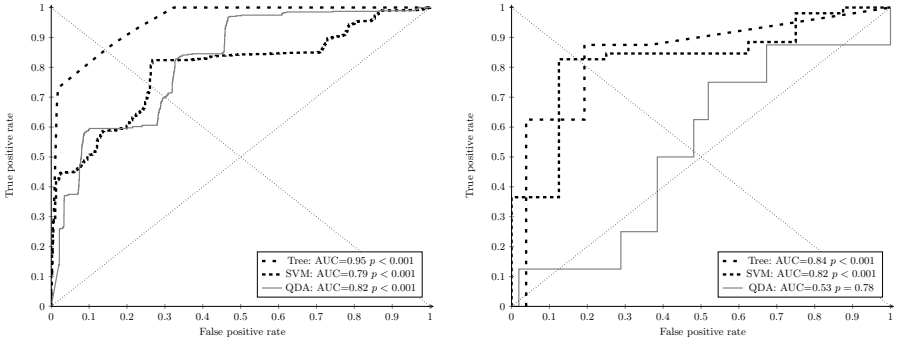
Predicted / True	First 1 day		First 2 days		First 3 days	
	(A)	(D)	(A)	(D)	(A)	(D)
(A)live	94.23%	5.77%	86.54%	13.46%	96.15%	3.85%
(D)ead	62.50%	37.50%	75.00%	25.00%	37.50%	62.50%
Overall	86.67%		78.33%		**91.67%**	

(b) 3-month GOSE

Predicted / True	Last 3 days		Last 2 days		Last 1 day	
	(G)	(B)	(G)	(B)	(G)	(B)
(G)ood	68.00%	32.00%	44.00%	56.00%	52.00%	48.00%
(B)ad	17.14%	82.86%	37.14%	62.86%	48.57%	51.43%
Overall	**76.67%**		55.00%		51.67%	

We then applied two other different learning methods, the support vector machine (SVM) and the quadratic discriminant analysis. The ROC AUC is employed to assess the performance of different classifiers. As noted above, ROC graphs depict the tradeoff between sensitivity and specificity for each classifier in both training and testing data sets, and the AUC measures the probability of the classifier assigning a higher score to the positive than to the negative case, if

one positive and one negative case were to be randomly drawn. Figures 2a and 2b show the in-hospital mortality prediction on the training and testing sets, using the first three days' VS. Figures 3a and 3b compare prediction power of three classifiers for 3-month GOSE using the last three days' VS. Note that the classifier built by the classification tree has the best discrimination for mortality prediction on both the training and the testing sets. The classification tree also has good discrimination capability on the 3-month GOSE outcomes.



(a) Training (b) Testing

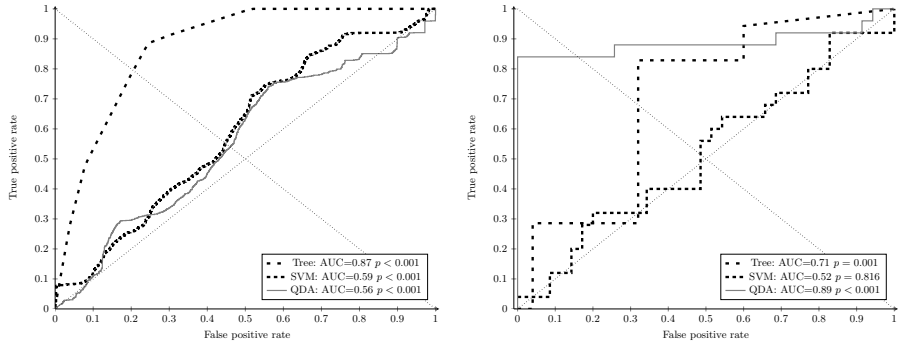
Fig. 2. ROCs of mortality classifiers built by three learning methods, using 3 days training set

3.3 Baseline

In this section, we compare our results with other models created from clinical experience to demonstrate that permutation entropy method has stable and comparable performance.

Many empirical models have been studied and reported to estimate patients' current and future status. With computer assistance, more statistical metrics can be calculated from long duration vital signs records. Previous work by our group [12,21] on this same dataset studied cumulated dose of ICP $>$ 20mm Hg, CPP $<$ 60mm Hg and Brain Trauma Index (BTI=CPP/ICP) as features to predict functional outcomes for patients of severe BTI, using ROC analysis and observed good predictive power for 3-month GOSE 1-4 (AUC=0.65~0.75, $p < 0.05$) [21].

To compare with features built from the permutation entropy, up to 5 features from the 5 vital signs in Table 1 were selected. Mean values of HR, SpO_2, SBP, shock index (SI=SPB/HR), and ICP were calculated using the first 3 days data for the in-hospital mortality prediction, and the last 3 days for the 3-month GOSE. Table 3 compares the performance of classification tree built on features from the permutation entropy and the top 3 classification trees built on subsets of features out of total $\sum_{k=1}^{5} C_5^k = 31$ combinations from the Table 1.



(a) Training (b) Testing

Fig. 3. ROCs of 3M-GOSE classifiers built by three learning methods, using 3 days training set

Table 3. Comparison between permutation entropy and baseline models on testing set

Decision tree features	Mortality Accu.	AUC	p-value	Decision tree features	3-month GOSE Accu.	AUC	p-value
Entropy	91.67%	0.84	< 0.001	Entropy	76.67%	0.71	0.001
ICP/SPO2/HR	85.00%	0.71	0.057	SPO2/HR	68.33%	0.69	0.005
SBP	81.67%	0.82	< 0.001	SPO2/SBP/HR	65.00%	0.68	0.009
SI/SBP	80.00%	0.78	0.005	SI/SPO2/HR	63.33%	0.67	0.013

It can be observed that the classification tree built upon features created by the permutation entropy demonstrated better performance in terms of overall accuracy and values of AUC for both in-hospital mortality and 3-month GOSE prediction.

4 Conclusion

4.1 Summary

Using a large collection of continuous, automated, electronic patient VS data, we derived features to quantify the complexity of this dynamic system using permutation entropy and found that VS features can predict in-hospital mortality and 3-month GOSE, despite a skewed dataset from relatively few instances. These features created by permutation entropy demonstrated promising results. Among 13.3% deaths (58.3% unfavorable cases), we observed 91.67% overall accuracy (62.5% for deaths) for in-hospital mortality prediction, and 76.67% in 3-month GOSE prediction (82.86% for bad outcomes). In comparison with other classifiers on the same dataset, permutation entropy predicted in-hospital mortality and 3-month GOSE with greater accuracy and area under the receiver

operating characteristic curves (ROC AUC=0.84, $p < 0.001$ for mortality, and ROC AUC=0.71, $p = 0.001$ for 3-month GOSE on testing sets).

Permutation entropy is capable of capturing the essentials of dynamic systems described by time series, which can be used to create interpretable decision rules. The capability that this method displays in our study to identify within the first 12 hours of care changes in VS associate with long-term outcome, offers clinicians the potential for early interventions, which may improve outcome.

4.2 Future Work

In this study, we used features created by permutation entropy to compare the capabilities of this technique with AUC in prediction of outcome. The accuracy of the prediction models can be improved by including extra descriptive features, such as those features studied in comparison. Furthermore, patients can be categorized into refined subgroups, for which more specific models can be built by categorizing by age or types of injury.

Higher frequency data can be used to enhance early prediction. Optimal calculation of entropy requires time series of sufficient length for a reasonable estimation of ordinal pattern distribution. Using higher frequency data, such as waveform data, permutation entropy may be able to create features to describe the system complexity in earlier time series, such as the first 12 hours in the hospital.

4.3 Clinical Implication

Access to valid clinical prognosis is important in the first 72 hours of care among a group of patients typically hospitalized for several weeks. However, the overall mean time to death for people who died of TBI in our system is 24 hours [6]. Our long-term goal in this work is to provide the critical care team with access to valid clinical prognosis in the first 12 hours after hospital admission and even, if possible, during pre-hospital care and transport, maximizing the potential for timely therapeutic interventions that can save lives and, more importantly, improve long-term clinical outcome.

Acknowledgments. The authors would like to thank the reviewers for their valuable comments. This research was funded in part by the grants W81XWH-07-2-0118 (Early Support of Intracranial Perfusion) and FA8650-11-2-6D01 (Continuing non-invasive monitoring and the development of predictive triage indices for outcomes following trauma). The authors thank Karen Murdoch, RPT; Melissa Binder, BS; Betsy Kramer RN, MS; and the ONPOINT investigators group* for their support.

*The ONPOINT investigators group includes: Tom Grissom, Amechi Anazodo, Patrick Boyle, Anthony Herrera, Chein-I Chang, Chris Stephens, Colin Mackenzie, Deborah Stein, George Hagegeorge, Cris Imle, Jay Menaker, John Blenko, John Hess, Peter Hu, Joseph duBose, Karen Murdock, Linda Goetz, Lisa Gettings, Victor Guistina, Lynn Smith, Lynn Stansbury, Matthew Lissauer,

Raymond Fang, Sarah Saccicchio, Robert Sikorski, Stacy Shackelford, Steven J. Barker, Theresa Dinardo, Thomas Scalea, Tim Oates, and Yvette Fouche.

References

1. Bandt, C., Pompe, B.: Permutation entropy – a natural complexity measure for time series. Phys. Rev. Lett. 88(17) (April 2002)
2. Bezerianos, A., Tong, S., Thakor, N.: Time-dependent entropy estimation of EEG rhythm changes following brain ischemia. Ann. Biomed. Eng. 31(2), 221–232 (2003)
3. Bruzzo, A.A., Gesierich, B., Santi, M., Tassinari, C.A., Birbrumer, N., Rubboli, G.: Permutation entropy to detect vigilance changes and preictal states from scalp EEG in epileptic patients. a preliminary study. Neurol Sci. 29(1), 3–9 (2008)
4. Cai, Y., Qiu, Y., Wei, L., Zhang, W., Hu, S., Smith, P.R., Crabtree, V.P., Tong, S., Thakor, N.V., Zhu, Y.: Complex character analysis of heart rate variability following brain asphyxia. Med Eng. Phys. 28(4), 297–303 (2006)
5. Cao, Y., Wen Tung, W., Gao, J.B., Protopopescu, V.A., Hively, L.M.: Detecting dynamical changes in time series using the permutation entropy. Phys. Rev. E 70(4) (October 2004)
6. Dutton, R.P., Stansbury, L.G., Leone, S., Kramer, E., Hess, J.R., Scalea, T.M.: Trauma mortality in mature trauma systems: are we doing better? an analysis of trauma mortality patterns, 1997-2008. J Trauma 69(3), 620–626 (2010)
7. Fawcelt, T.: Roc graphs: Notes and practical considerations for data mining researchers. In: Intelligent Enterprise Technologies Laboratory HP Laboratories Palo Alto, HPL-2003-4 (January 2003)
8. Gao, D., Hu, J., Buckley, T., White, K., Hass, C.: Shannon and Renyi entropy to classify effects of mild traumatic brain injury on postural sway. PLoS One 6(9) (2011)
9. Guo, D.: Local entropy map: A nonparametric approach to detecting spatially varying multivariate relationships. Int. J. Geogr. Inf. Sci. 24, 1367–1389 (2010)
10. Jennett, B., Snoek, J., Bond, M.R., Brooks, N.: Disability after severe head injury: observations on the use of the glasgow outcome scale. J. Neurol Neurosurg Psychiatry 44(4), 285–293 (1981)
11. Kahraman, S., Dutton, R.P., Hu, P., Stansbury, L., et al.: Heart rate and pulse pressure variability are associated with intractable intracranial hypertension after severe traumatic brain injury. Clinical investigation 22(4) (October 2010)
12. Kahraman, S., Hu, P., Stein, D., Stansbury, L., Dutton, R., Xiao, Y., Hess, J., Scalea, T.: Dynamic three-dimensional scoring of cerebral perfusion pressure and intracranial pressure provides a brain trauma index that predicts outcome in patients with severe traumatic brain injury. J. Trauma 70(3), 547–553 (2011)
13. Li, X., Cui, S., Voss, L.J.: Using permutation entropy to measure the electroencephalographic effects of sevoflurane. Anesthesiology 109(3), 448–456 (2008)
14. Li, X., Ouyang, G., Richards, D.A.: Predictability analysis of absence seizures with permutation entropy. Epilepsy Res. 77(1), 70–74 (2007)
15. Lopes, F.M., de Oliveira, E.A., Cesar, J.R.M.: Inference of gene regulatory networks from time series by Tsallis entropy. BMC Systems Biology 5(61) (2011)
16. Nicolaou, N., Georgeiou, J.: The use of permutation entropy to characterize sleep electroencephalograms. Clin. EEG Neurosci. 42(1), 24–28 (2011)
17. Olofsen, E., Sleigh, J.W., Dahan, A.: Permutation entropy of the electroencephalogram: a measure of anaesthetic drug effect. Br. J. Anaesth. 101(6), 810–821 (2008)

18. Ouyang, G., Dang, C., Richards, D.A., Li, X.: Ordinal pattern based similarity analysis for EGG reordering. Clin. Neurophysiol. 121(5), 694–703 (2010)
19. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers, pp. 61–74. MIT Press (1999)
20. Provost, F., Domingos, P.: Well-trained pets: Improving probability estimation trees (2000)
21. Stein, D., Hu, P.F., Brenner, M., Sheth, K., et al.: Brief episodes of intracranial hypertension and cerebral hypoperfusion are associated with poor functional outcome after severe traumatic brain injury. Journal of Trauma-Injury Infection & Critical Care 71(2), 364–374 (2011)
22. Sun, X., Zou, Y., Nikiforova, V., Kurths, J., Walther, D.: The complexity of gene expression dynamics revealed by permutation entropy. BMC Bioinformatics 11, 607 (2010)
23. Zanin, M.: Forbidden patterns in financial time series. Chaos 18(1), 013119 (2008)
24. Zhang, D., Jia, X., Ding, H., Ye, D., Thakor, N.V.: Application of Tsallis entropy to EEG: quantifying the presence of burst suppression after asphyxial cardiac arrest in rats. IEEE Trans. Biomed. Eng. 57(4), 867–874 (2010)

EEG Signals Classification Using a Hybrid Method Based on Negative Selection and Particle Swarm Optimization

Nasser Omer Ba-Karait[1], Siti Mariyam Shamsuddin[1], and Rubita Sudirman[2]

[1] Soft Computing Research Group, Faculty of Computer Science and Information Systems
[2] Faculty of Electrical Engineering
[1,2] Universiti Teknologi Malaysia, 81310 Skudai, Johor Bahru, Malaysia
bakarait@yahoo.com, mariyam@utm.my, rubita@fke.utm.my

Abstract. The diagnosis of epilepsy from EEG signals by a human scorer is a very time consuming and costly task considering the large number of epileptic patients admitted to the hospitals and the large amount of data needs to be scored. In this paper, a hybrid method called adaptive particle swarm negative selection (APSNS) was introduced to automate the process of epileptic seizures detection in EEG signals. In the proposed method, an adaptive negative selection creates a set of artificial lymphocytes (ALCs) that are tolerant to normal patterns. However, the particle swarm optimization (PSO) algorithm forces these ALCs to explore the space of epileptic signals and maintain diversity and generality among them. The EEG signals were analyzed using discrete wavelet transform (DWT) to extract the most important information needed for decision making. The features extracted have been used to investigate the performance of the proposed APSNS algorithm in classifying the EEG signals. The Experimental results confirm effectiveness and stability of the proposed method. Its classification accuracy outperforms many of the methods in the literature.

Keywords: Electroencephalogram, epileptic seizure, discrete wavelet transform, machine learning, particle swarm optimization, artificial immune system.

1 Introduction

Brain activity can be measured in a variety of ways such as Magneto Encephalogram (MEG), optical images, and Electroencephalogram (EEG) signals. The EEG signal is a highly complex signal represents the electrical activity of the brain. In the last decades, the EEG has been intensively studied due to it conveys valuable clinical information used to study brain function and neurological disorders. Thus, the EEG has long been an important clinical tool in diagnosing, monitoring and managing neurological disorders, especially those related to epilepsy [1-3]. Epileptic seizures are caused by temporary electrical disturbance of the brain. Seizures may go unnoticed, depending on their presentation, but sometimes may be confused with other events, such as a stroke, which can also cause falls or migraines. The occurrence of a seizure seems unpredictable and its course of action is still very poorly understood. Research is needed for better understanding of the mechanisms causing epileptic disorders.

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 427–438, 2012.

Careful analysis of the EEG records could provide valuable insight into this widespread brain disorder [4, 5].

When diagnosed properly, many cases of epilepsy can be controlled effectively by medications or surgical treatments. In the case of surgical treatments, patients undergo long presurgical evaluations. During this period, large numbers of multi-channel EEG recordings are acquired for locating the epileptic part of the brain to be removed during the surgery [6, 7]. Clearly, analysis of the recorded EEG based on visual inspection is a very time consuming and costly task. In some other cases, individuals with epilepsy have seizures that are uncontrollable. Recently, methods have started being developed for medically resistant epilepsy. In these methods, a local therapy such as direct electrical stimulation or chemical infusions is delivered to the affected regions of the brain in order to avoid the onset of a seizure. Detection of seizures automatically forms an integral part of such methods [7, 8].

With the above premises, there is a great need for development of automated systems to recognize EEG changes. Therefore, tremendous effort has long been devoted by researchers for solving this problem and various methods have been presented in the literature. Mostly, these approaches coming from the area of artificial intelligence (AI) such as artificial neural networks [4-7, 9-11], adaptive neuro-fuzzy inference system [12-14] , support vector machine [3, 15-17] , decision tree [18, 19], and artificial immune system [20]. As it can be seen in above mentioned studies, the features that characterize the behavior of the EEG signals are extracted using techniques such as fourier transform, autoregressive, wavelet transform, and eigenvector methods.

However, algorithms involving artificial immune systems (AIS) have not been widely explored in the field of EEG-based medical diagnosis. Only few studies exists in the literature such as Polat and Güneş [20] in which artificial immune recognition system (AIRS) algorithm was applied for EEG signals classification. Therefore, investigating the performance of other AIS algorithms such as the negative selection algorithm (NSA) is of great importance. In this study, an adaptive NSA was hybridized with the particle swarm optimization (PSO) algorithm to introduce a novel method named adaptive particle swarm negative selection (APSNS) algorithm. The performance of the proposed algorithm in classifying the EEG signals was evaluated using features extracted by discrete wavelet transform (DWT).

2 Artificial Immune Systems

In the 1990s, artificial immune systems (AIS) emerged as a new computational research field inspired by the simulated biological behavior of the natural immune system (NIS). The NIS is a very complex biological network with rapid and effective mechanisms for defending the body against a specific foreign body material or a pathogenic material called antigen [21]. During the reactions, the adaptive immune system memorizes the characteristic of the encountered antigen by producing plasma or memory cells. The obtained memory promotes a rapid response of the adaptive immune system to future exposure to the same antigen [22]. In order to respond only to antigen, the immune system distinguishes between what is normal (self) and foreign

(non-self or antigen) in the body. The NIS is made up of lymphocytes, which are white blood cells circulate throughout the body, mainly of two types, namely B-cells and T-cells. These cells play the main role in the process of recognizing and destroying antigens [23].

Both T-cell and B-cell are created in the bone marrow and they have receptor molecules on their surfaces (the B-cell receptor molecule is also known as antibody). The way B-cells and T-cells can identify a specific antigen is called a key and key hole relationship as explained in Fig.1 [21]. In this case, the antigen and the receptor molecule have complementary shapes and so they can bind together with a certain binding strength, measured as affinity. After a binding between an antibody's paratope and an antigen's epitope, an antigen-antibody-complex is formed which results in deactivation of the antigen. The B-cell is already mature after creation in the bone marrow, whereas the T-cell first becomes mature in the thymus. A T-cell becomes mature if and only if it does not have receptors that bind with molecules that represent self cells. Consequently, it is very important that the T-cell can differentiate between self and non-self cells [24].

The AIS as defined by de Castro and Timmis [25] are: "Adaptive systems inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving". The AIS are one of many types of algorithms inspired by biological systems, such as neural networks, evolutionary algorithms and swarm intelligence. There are many different types of algorithms within AIS and research to date has focused primarily on the theories of immune networks, clonal selection and negative selection. These theories have been abstracted into various algorithms and applied to a wide variety of application areas such as anomaly detection, pattern recognition, learning, and robotics [26].

The negative selection algorithm (NSA) introduced by Forrest *et al*. in 1994 [27] inspired by the mature T-cells of the natural immune system; which are self-tolerant, that is mature T-cells have the ability to distinguish between self cells and foreign/non-self cells. The NSA uses a set of self patterns to train a set of artificial lymphocytes (ALCs) to be self-tolerant. These ALCs are applied as detectors to classify new data as self or non-self [25]. In NSA, any generated ALC is added to the self-tolerant set if the calculated affinity between the ALC and all self patterns is higher than a specified affinity threshold. The algorithm is summarized as in Alg.1.

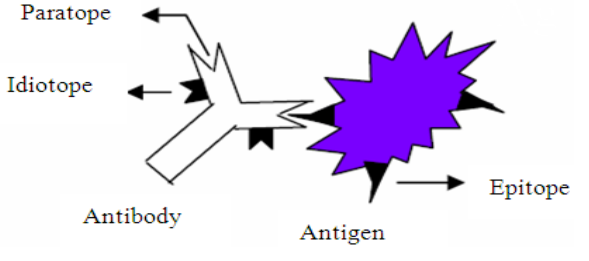


Fig. 1. Antibody-antigen complex

Alg.1. Negative selection algorithm

Create an empty set of self-tolerant ALCs as C;
Determine the training set of self patterns as Z_S;
Repeat
  Randomly generate an ALC, c_i;
  Calculate the affinity between c_i and each pattern in Z_S;
    If the calculated affinity with at least one pattern in Z_S is lower than the affinity threshold, then *reject* c_i;
    Otherwise *add* c_i to set C;
Until size of C equals some predefined number;

3 Particle Swarm optimization

The particle swarm optimization (PSO) algorithm was originally designed by Kennedy and Eberhart in 1995 [28]. The idea was inspired by the social behavior of flocking organisms. It belongs to the broad class of stochastic optimization algorithms that may be used to find optimal (or near optimal) solutions to numerical and qualitative problems. PSO uses a population (swarm) of individuals (particles) to probe promising regions of the search space. Each particle moves in the search space with a velocity that is dynamically adjusted according to its own flying experience and its companions flying experience and retains the best position it ever encountered in memory. The best position ever encountered by all particles of the swarm is also communicated to all particles [29].

The popular form of the PSO algorithm is defined as:

$$v_{id}(t+1) = w * v_{id}(t) + c_1 r_1 (pbest_{id}(t) - x_{id}(t)) + c_2 r_2 (gbest_d(t) - x_{id}(t)) \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (2)$$

where v_{id} is the velocity of particle i along dimension d, x_{id} is the position of particle i in d, c_1 is a weight applied to the cognitive learning portion, and c_2 is a similar weight applied to the influence of the social learning portion. r_1 and r_2 are separately generated random number in the range of zero and one. $pbest_{id}$ is the previous best location of particle i. $gbest_d$ is the best location found by the entire population. w is the inertia weight. Velocity values must be within a range defined by two parameters $-v_{max}$ and v_{max}. The PSO with the inertia weight in the range (0.9, 1.2) on average have a better performance. To get a better searching pattern between global exploration and local exploitation, researchers recommended decreasing w over time from a maximal value w_{max} to a minimal value w_{min} linearly [30, 31].

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{t_{\max}} * t \quad (3)$$

where, t_{max} is the maximum number of iterations allowed and t is the current iteration number.

4 Materials and Methods

4.1 EEG Data

The present work used the publicly available EEG data described by Andrzejak *et al.* [32]. In this dataset, all EEG signals were recorded with the same 128-channel amplifier system, using an average common reference. The analog data were digitized at 173.61 samples per second by a 12 bit A/D resolution with band-pass filter settings of 0.53-40 Hz (12 dB/oct). The complete dataset contains five different sets (denoted A-E), each containing 100 single channel EEG segments of 23.6 sec. duration. These signals were selected and cut out from continuous multi-channel EEG recordings after removing artifacts caused due to eye movements, scalp muscular activity and power line interference.

Signals in sets A and B have been recorded from five healthy volunteers through external surface electrodes using the international 10–20 electrode placement scheme. The volunteers were relaxed in an awake state with eyes open (set A) and closed (set B). The EEG archive of presurgical diagnosis was used to originate sets C, D and E. EEG recordings taken from five patients using intracranial electrodes were selected. All patients had achieved complete seizure control after resection of one of the hippocampal formations, which was therefore correctly diagnosed to be the epileptogenic zone. Segments in sets C and D were measured in seizure free intervals from within the epileptogenic zone and opposite the epileptogenic zone of the brain, respectively. Set E were obtained from within the epileptogenic zone during seizure activity. Fig.2 shows typical EEG segments, one from each category.

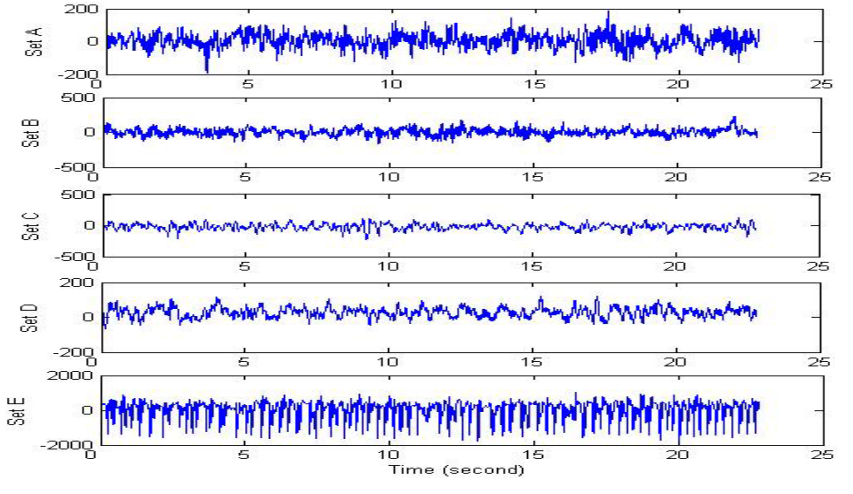


Fig. 2. Samples of five different sets of EEG data

4.2 Discrete Wavelet Transform: Feature Extraction

Discrete wavelet transform (DWT) has been particularly successful in the area of epileptic seizure detection due to its ability to capture transient features and localize them in both time and frequency domains accurately [9]. The DWT analyzes the signal $s(n)$ at different frequency bands by decomposing the signal into an approximation and detail information using two sets of functions known as scaling functions and wavelet functions, which are associated with low-pass $g(n)$ and high-pass $h(n)$ filters, respectively. Fig.3 shows the decomposition process of DWT.

When the DWT is used to analyze the signals, two important aspects should be considered: the number of decomposition levels and the type of wavelet. The decomposition levels number is selected based on the dominant frequency components of the signal. According to Subasi [10], the levels are selected such that those parts of the signal that correlate well with the frequencies required for the signal classification are retained in the wavelet coefficients. Therefore, level 4 wavelet decomposition was selected in the present study. Accordingly, the EEG signals have been decomposed into the details D1-D4 and one final approximation, A4. Table 1 shows the ranges of the various frequency bands of EEG data used. The smoothing feature of the Daubechies wavelet of order 2 (db2) made it more suitable to detect changes of EEG signals [12]. In this research, the db2 has been used to compute the wavelet coefficients of the EEG signals.

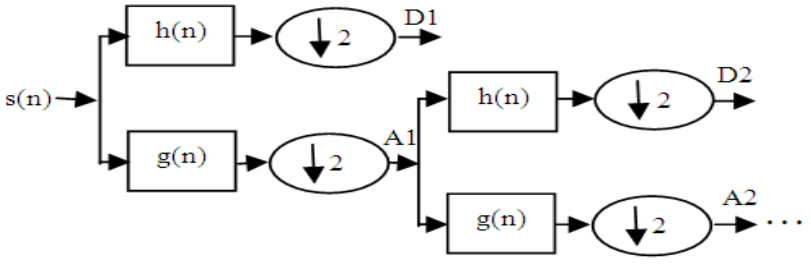


Fig. 3. Sub-band decomposition of DWT

Table 1. different decomposition levels Frequencies of db2 wavelet for the EEG dataset

Decomposed signal	Frequency range (Hz)
D1	43.4-86.8
D2	21.7-43.4
D3	10.8-21.7
D4	5.4-10.8
A4	0.0-05.4

The computed coefficients of discrete wavelet provide a compact representation that shows the energy distribution of the signal in time and frequency. In order to further decrease dimensionality of the extracted feature vectors, statistics over the set of the wavelet coefficients are used [12]. The following statistical features were used

to represent the time-frequency distribution of the EEG signals: Maximum, Minimum, Mean, and Standard deviation of the wavelet coefficients in each sub-band.

4.3 Adaptive Particle Swarm Negative Selection: EEG Classification

Adaptive particle swarm negative selection (APSNS) algorithm is a hybrid method based on PSO and negative selection algorithms. It has been introduced in this research to classify EEG signals for diagnosis purposes. In APSNS, all patterns are represented in space as real-valued vector and Euclidean distance is used as the affinity measure. Each ALC has its own affinity threshold, *r*, to determine the matching with a non-self pattern. The steps of the algorithm are summarized in Alg.2.

An adaptive negative selection algorithm is proposed to evolve a set of ALCs to be self-tolerant, meaning they have the ability not to match any self pattern. Therefore, self patterns are used as the training set. The algorithm determines for each ALC its affinity threshold, *r*. To guarantee no overlap with the self, the *r* of the ALC is set to the closest self pattern. However if the value of *r* is equal to zero, the ALC is replaced by a new one. Otherwise, the ALC is considered self-tolerant, and it will classify any pattern as non-self if the distance between them is less than *r*.

Generally, self-tolerant ALCs do not cover all non-self space. In fact, only some of the non-self will be detected and only some of these ALCs will detect non-self patterns. Therefore, the PSO algorithm is used to promote the ALCs in self-tolerant set to new status called memory have a high ability to separate the non-self patterns from the self. In each run, PSO produces one optimal ALC which is added to the set of memory ALCs only if it detects new patterns in non-self training set.

The objectives of the PSO are to take the ALCs away from self patterns towards non-self space and to maintain diversity and generality among the ALCs. Hence, PSO needs to maximize: (1) the value of *r* for the evolved ALC, and (2) the distance between the new ALC and the ALCs in memory set. This guarantees the lowest average overlap between the memory ALCs and forces greater coverage of non-self space. To evaluate the quality of an ALC, c_i, fitness of i^{th} particle is calculated using the following function:

$$FitF(M, c_i) = \frac{1}{2}(r + DivF(M, c_i)) \tag{4}$$

where M is the set of memory ALCs, c_i is the ALC which the fitness must be calculated, and

$$DivF(M, c_i) = \frac{\sum_{j=1}^{|M|} Ed(m_j, c_i)}{|M|} \tag{5}$$

where m_j is the j^{th} ALC in the memory set and *Ed* returns Euclidean distance.

Alg.2. Adaptive particle swarm negative selection algorithm

1. Create an empty set of memory ALCs, *M*
2. Repeat
  (a) Initialize *N* particles, *X*
  (b) For t= 1 to t_{max}
    (i) Send *X* to adaptive negative selection to create self-tolerant ALCs set, *C*
    (ii) For each particle c_i
      (1) Calculate the fitness using Eq. (4)
      (2) Find personal best solution, *pbest*
    (iii) Find the global best solution, *gbest*
    (iv) Update each particle using Eq. (1) and (2)
  (c) If *gbest* detects new patterns then add it to the set *M*
3. Until non-self is covered or a maximum number of iterations is reached

5 Experimental Results

5.1 Performance Measures

In medical diagnosis tasks, the common performance measures are sensitivity, specificity and classification accuracy. The sensitivity is defined by the percentage of correctly detected epileptic EEG patterns to the total number of patterns in the epileptic EEG. On the other hand, specificity is defined by percentage of correctly detected normal EEG patterns to the total number of patterns in the normal EEG. Finally, the percentage of all correctly classified patterns to the total number of patterns in both normal and seizure EEG dataset represent the accuracy. Formally, the performance of a diagnostic system is measured as:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{7}$$

where TP, TN, FP and FN denote true positives, true negatives, false positives and false negatives respectively.

$$Accuracy\,(Z_T) = \frac{\sum_{i=1}^{|Z_T|} Classify\,(z_{Ti})}{|Z_T|} \tag{8}$$

where, Z_T is the testing set, z_{Ti} is a pattern in Z_T to be classified, and *Classify* (z_{Ti}) returns one if z_{Ti} classified correctly by the algorithm and zero if not.

5.2 Results and Discussion

The APSNS algorithm was evaluated on the EEG data in order to investigate its performance in detecting the epileptic seizures. In the present study, the sets A and E were selected of the complete dataset to represent the normal and epileptic classes respectively. In each set of EEG data, there are 100 EEG signals of 4096 samples. Each signal is further divided by a rectangular window composed of 256 samples. Hence, the dataset was formed of 3200 patterns, i.e., each class has 1600 patterns. The DWT coefficients at the fourth level (D1-D3, D4 and A4) were computed for each pattern. The statistical features that were calculated over the set of wavelet coefficients reduced the dimensionality of feature vector to 20.

In machine learning field, it is common to partition the dataset into two separate sets: a training set and a testing set. Additionally, k-fold cross validation is often used by the researchers to evaluate the behavior of the algorithm in the bias associated with the random sampling of the training data. In this research, the EEG dataset (sets A and E) was randomly divided into training-testing as 50-50%, 60-40%, and with 10-fold cross validation. The class distribution of the patterns in the training and testing sets are summarized in Table 2.

Table 2. Class distribution of the patterns in the training and testing EEG datasets

Training-testing dataset partitions (%)		Class		Total
		Normal	Epileptic	
50-50	Training set	800	800	1600
	Testing set	800	800	1600
60-40	Training set	960	960	1920
	Testing set	640	640	1280
10-fold cross validation	Training set	1440	1440	2880
	Testing set	160	160	320

Ten particles (N=10) were trained for 200 iterations (t_{max}=200) to create the ALCs of the memory set. The values of other parameters of APSNS are the following: v_{max}=0.05, c_1=2.0, c_2=2.0, w_{max}=0.9, w_{min}=0.4. Consequentially, the ability of the generated memory ALCs is tested in order to assess effectiveness of the proposed method. Table 3 presents the results achieved by APSNS algorithm on the testing set with respect to sensitivity, specificity and accuracy in terms of average (Avg) and standard deviation (SD) of 10 runs. As it is seen from Table 3, the APSNS classified the EEG signals of training-test datasets partitions: 50-50%, 60-40%, and 10-fold cross validation with the accuracies of 99.44%, 99.60%, and 99.66% respectively. The results show good performance and stable behavior of the proposed method in recognizing epileptic and normal activities in the brain.

A comparison of the proposed algorithm with previous studies in the literature is shown in Table 4. Only the studies that used the same EEG dataset with the sets A and E are considered. Besides, all results illustrated in Table 4 are according to same training-test dataset partition and in terms of classification accuracy. The comparison proves the competitiveness of the APSNS algorithm for the epileptic seizures diagnosis in EEG signals.

Table 3. The values of Average, and SD for sensitivity, specificity, and accuracy of APSNS algorithm on EEG signals

Training-testing dataset partitions (%)	Performance measures (%)			
		Sensitivity	Specificity	Accuracy
50-50	Avg	99.69	99.19	99.44
	SD	0.21	0.55	0.23
60-40	Avg	99.73	99.47	99.60
	SD	0.28	0.45	0.26
10-fold cross validation	Avg	99.63	99.69	99.66
	SD	0.79	0.67	0.45

Table 4. Comparison of classification accuracy of the APSNS algorithm on EEG signals with methods in the literature

Study	Accuracy (%)	
	Previous study	This study
Kannathal *et al.* [13]	92.22	99.60
Polat and Güneş [18]	98.72	99.66
Subasi [4]	94.50	99.60
Polat and Güneş [20]	99.81	99.44
Chandaka *et al.* [17]	95.96	99.44
Übeyli [3]	99.56	99.44
Kumar *et al*. [11]	99.75	99.60
Wang *et al*. [33]	99.50	99.66
Nicolaou and Georgiou [16]	93.55	99.60

6 Conclusion

The present study introduced a hybrid detection system for automatic diagnosis of epileptic seizures in EEG signals. In this system, the diagnosis process is performed in two stages: feature extraction using discrete wavelet transform and decision maker using adaptive particle swarm negative selection. The ability of the proposed method has been evaluated on EEG dataset that have healthy and seizure signals. The results reveal that the APSNS shows promising performance for EEG signals discrimination compared to other methods in the literature. The proposed system could be an efficient tool to assist the experts by facilitating analysis of a patient's information and reducing the time and effort required to make accurate decisions on their patients.

Acknowledgment. This research is supported by the Ministry of Higher Education (MOHE) and Universiti Teknologi Malaysia (UTM) under Research University Grant (VOT Q.J130000.2528.00H71). The authors would like to thank Soft Computing Research Group, BioMedical & Insrumentation Electronics Research Group, and Hadhramout University of Science and Technology for the support in making this study a success.

References

1. Hapuarachchi, P.: Feature selection and artifact removal in sleep stage classification. Master Thesis, University of Waterloo. Canada (2006)
2. Adeli, H., Zhou, Z., Dadmehr, N.: Analysis of EEG records in an epileptic patient using wavelet transform. Journal of Neuroscience Methods 123, 69–87 (2003)
3. Übeyli, E.D.: Least squares support vector machine employing model-based methods coefficients for analysis of EEG signals. Expert Systems with Applications 37, 233–239 (2010)
4. Subasi, A.: EEG signal classification using wavelet feature extraction and a mixture of expert model. Expert Systems with Applications 32, 1084–1093 (2007)
5. Nigam, V.P., Graupe, D.: A neural-network-based detection of epilepsy. Neurological Research 26, 55–60 (2004)
6. Ocak, H.: Optimal classification of epileptic seizures in EEG using wavelet analysis and genetic algorithm. Signal Processing 88, 1858–1867 (2008)
7. Patnaik, L.M., Manyam, O.K.: Epileptic EEG detection using neural networks and post-classification. Computer Methods and Programs in Biomedicine 91, 100–109 (2008)
8. Gardner, A.B.: A novelty detection approach to seizure analysis from intracranial EEG. PhD Thesis, Georgia Institute of Technology. Georgia, United States (2004)
9. Subasi, A.: Automatic detection of epileptic seizure using dynamic fuzzy neural networks. Expert Systems with Applications 31, 320–328 (2006)
10. Subasi, A.: Epileptic seizure detection using dynamic wavelet network. Expert Systems with Applications 29, 343–355 (2005)
11. Kumar, S.P., Sriraam, N., Benakop, P.G., Jinaga, B.C.: Entropies based detection of epileptic seizures with artificial neural network classifiers. Expert Systems with Applications 37, 3284–3291 (2010)
12. Güler, İ., Übeyli, E.D.: Adaptive neuro-fuzzy inference system for classification of EEG signals using wavelet coefficients. Journal of Neuroscience Methods 148, 113–121 (2005)
13. Kannathal, N., Choo, M.L., Acharya, U.R., Sadasivan, P.K.: Entropies for detection of epilepsy in EEG. Computer Methods and Programs in Biomedicine 80, 187–194 (2005)
14. Übeyli, E.D.: Automatic detection of electroencephalographic changes using adaptive neuro-fuzzy inference system employing Lyapunov exponents. Expert Systems with Applications 36, 9031–9038 (2009)
15. Acır, N., Güzeliş, C.: Automatic spike detection in EEG by a two-stage procedure based on support vector machines. Computers in Biology and Medicine 34, 561–575 (2004)
16. Nicolaou, N., Georgiou, J.: Detection of epileptic electroencephalogram based on Permutation Entropy and Support Vector Machines. Expert Systems with Applications 39, 202–209 (2012)
17. Chandaka, S., Chatterjee, A., Munshi, S.: Cross-correlation aided support vector machine classifier for classification of EEG signals. Expert Systems with Applications 36, 1329–1336 (2009)
18. Polat, K., Güneş, S.: Classification of epileptiform EEG using a hybrid system based on decision tree classifier and fast Fourier transform. Applied Mathematics and Computation 187, 1017–1026 (2007)
19. Valenti, P., Cazamajou, E., Scarpettini, M., Aizemberg, A., Silva, W., Kochen, S.: Automatic detection of interictal spikes using data mining models. Journal of Neuroscience Methods 150, 105–110 (2006)

20. Polat, K., Güneş, S.: Artificial immune recognition system with fuzzy resource allocation mechanism classifier, principal component analysis and FFT method based new hybrid automated identification system for classification of EEG signals. Expert Systems with Applications 34, 2039–2048 (2008)
21. Hur, J.: Multi-robot system control using Artificial Immune System. PhD Thesis, The University of Texas at Austin. Texas, United States (2007)
22. Timmis, J., Hone, A., Stibor, T., Clark, E.: Theoretical advances in artificial immune systems. Theoretical Computer Science 403, 11–32 (2008)
23. Timmis, J., Neal, M., Hunt, J.: An artificial immune system for data analysis. Biosystems 55, 143–150 (2000)
24. Engelbrecht, A.P.: Computational intelligence: an introduction. John Wiley & Sons, England (2007)
25. de Castro, L.N., Timmis, J.: Artificial immune systems: a new computational intelligence approach. Springer, London (2002)
26. Smith, S.L., Timmis, J.: An immune network inspired evolutionary algorithm for the diagnosis of Parkinson's disease. Biosystems 94, 34–46 (2008)
27. Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-nonself discrimination in a computer. In: 1994 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, pp. 202–212 (1994)
28. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, Perth, pp. 1942–1948 (1995)
29. Ba-Karait, N.O.S., Shamsuddin, S.M.: Handwritten Digits Recognition using Particle Swarm Optimization. In: Second Asia International Conference on Modeling & Simulation, pp. 615–619. IEEE Xplore, Kuala Lumpur (2008)
30. Eberhart, R.C., Shi, Y.: Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 Congress on Evolutionary Computation, Seoul, pp. 81–86 (2001)
31. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of 1998 IEEE International Conference on Evolutionary Computation: IEEE World Congress on Computational Intelligence, Anchorage, pp. 69–73 (1998)
32. Andrzejak, R.G., Lehnertz, K., Mormann, F., Rieke, C., David, P., Elger, C.E.: Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. Physical Review E 64, 061907 (2001)
33. Wang, D., Miao, D., Xie, C.: Best basis-based wavelet packet entropy feature extraction and hierarchical EEG classification for epileptic detection. Expert Systems with Applications 38, 14314–14320 (2011)

DAGSVM vs. DAGKNN: An Experimental Case Study with Benthic Macroinvertebrate Dataset

Henry Joutsijoki and Martti Juhola

School of Information Sciences
University of Tampere, Kanslerinrinne 1, FI-33014 Tampere, Finland
henry.joutsijoki@uta.fi, csmajuh@sis.uta.fi

Abstract. In this paper we examined the suitability of the Directed Acyclic Graph Support Vector Machine (DAGSVM) and Directed Acyclic Graph k-Nearest Neighbour (DAGKNN) method in classification of the benthic macroinvertebrate samples. We divided our 50 species dataset into five ten species groups according to their group sizes. We performed extensive experimental tests with every group, where DAGSVM was tested with seven kernel functions and DAGKNN with four measures. Feature selection was made by the scatter method [8]. Results showed that the quadratic and RBF kernel functions were the best ones and in the case of DAGKNN all measures produced quite similar results. Generally, the DAGSVM gained higher accuracies than DAGKNN, but still DAGKNN is a respectable option in benthic macroinvertebrate classification.

Keywords: Directed acyclic graph support vector machine, directed acyclic graph k-nearest neighbour, machine learning, benthic macroinvertebrates, water quality, kernel function.

1 Introduction

Biological issues are an important part of the modern society. Different threats are constantly present in our everday life and we need to invent new methods for monitoring and predicting the state of the surrounding nature. Freshwater areas are a sensitive part of the environment and changes in it are quickly seen with the naked eye. Illegal dumping, oil emissions and other effluents can be some of the reasons for destruction of the sensitive fauna in the water systems. How can we investigate the exact consequences of the human induced actions? Benthic macroinvertebrates live on the bottom of the waterbodies and they quickly react to any changes in the state of the aquatic environment [18]. This is why the benthic macroinvertebrates are commonly used in biomonitoring.

Benthic macroinvertebrates consist of a large variety of species. One freshwater area can have dozens of species from many taxonomical groups. Wide diversity of the benthic macroinvertebrates makes their automatic taxa identification a challenging task. Differences between species can be very small making the automated identification process even harder from the pattern recognition point of view. Traditional approach to the identification is human-based when usually biological experts, taxonomists, perform the classification. A disadvantage of this approach is that it is time-consuming and, hence, the costs are high. The main idea is to automatize the classification procedure as

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 439–453, 2012.

far as it can be done. Thus, taxonomists and other biologists can focus their attention from often so routine identification process on more difficult and interesting problems. Moreover, biological experts can centralize their energy into solving the reasons behind the changes in the aquatic environments and to find out the solutions for these problems.

The classification of the benthic macroinvertebrates [5,6,7,10,11,12,13,14,17,18] is a difficult problem. Since the differences can be small between taxonomical groups, classification requires reliable and efficient methods. For the classification of benthic macroinvertebrates, the benthic animals are scanned and each scan was saved as an individual image. The identification of the benthic macroinvertebrates becomes even harder because they are not imaged in the same position. Moreover, the size and shape of the benthic macroinvertebrates vary in each image. Data is then heterogeneous and reflects the diversity of nature.

In this paper we have two aims to solve. Firstly, we want to investigate how Directed Acyclic Graph Support Vector Machine (DAGSVM) [5,15], a multi-class extension of SVM [1,2], succeeds in the classification of the benthic macroinvertebrate samples. Secondly, we present rarely in benthic macroinvertebrate classification used Directed Acyclic Graph k-Nearest Neighbour method (DAGKNN) and we examine how it works in this application. In feature selection we use a novel approach called scatter method [8]. In Section 2 we give a short overview of the SVM in a binary case [2,6,9] and we introduce DAGSVM [15] and DAGKNN. In Section 3 we describe data and test arrangements and, moreover, we analyse results. Section 4 is left for the discussion and further research questions.

2 Methods

2.1 Support Vector Machine

Suppose that we have a training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)$ where $\mathbf{x}_i \in \mathbb{R}^n$ are the training examples and $y_i \in \{-1, 1\}$ is the corresponding class label of $\mathbf{x}_i$. In the input space a separating hyperplane is $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$ where $\mathbf{w} \in \mathbb{R}^n$ is a weight vector and $b \in \mathbb{R}$ is a bias term. A decision function can now be stated as the sign of the $f(\mathbf{x})$. If we have a linearly separable training data, we can rescale weight vector and bias term such that the closest members of both classes lie in the canonical hyperplanes $|\langle \mathbf{w}, \mathbf{x} \rangle + b| = 1$. In other words the closest training points to the hyperplane are at the distance of $\frac{1}{\|\mathbf{w}\|}$ from the hyperplane. Hence, the distance between the canonical hyperplanes equals $\frac{2}{\|\mathbf{w}\|}$. We can maximize the margin by minimizing $\frac{1}{2}\|\mathbf{w}\|^2$ subject to $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1, i = 1, 2, \ldots, l$. This optimization problem can be solved by means of Lagrangian theory. Now, we want to minimize the primal Lagrangian:

$$\min_{\mathbf{w},b} L_P(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i [y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1]$$

where the Lagrange multipliers α_i's are non-negative. Moreover, L_P is maximized subject to $\boldsymbol{\alpha}$. By evaluating the derivatives respect to $\mathbf{w}$ and b and making a suitable

resubstitution, we obtain the dual form of the optimization problem. The dual form is more convenient to solve:

$$\max L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle, \tag{1}$$

subject to $\alpha_i \geq 0$ and $\sum_{i=1}^{l} \alpha_i y_i = 0$. Training examples having positive α_i are called as the support vectors. When we have linearly non-separable problems, we need more tools. An important factor in SVM theory was the invention to use the kernel trick where the training examples in the input space are mapped with a nonlinear transformation into a higher dimensional feature space where the optimal hyperplane can be constructed again. This method can be justified according to the Cover's theorem [3]. The difference between the equation (1) and the feature space version is that in the latter one $\mathbf{x}$ is replaced with $\phi(\mathbf{x})$. It is

$$\max L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle,$$

subject to $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^{l} \alpha_i y_i = 0$. An important fact is that actually we do not need to make mapping into a higher dimensional space and compute the inner products $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ there, because we can use kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$. Now, the decision function can be expressed as a sign of $f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b$ where α_i's are optimal.

2.2 DAGSVM and DAGKNN

Decision Directed Acyclic Graph (DDAG) is a learning structure introduced by Platt et al. [15]. DDAG is a graph where there are no cycles and the edges have directions. The main point for this structure is to combine the binary classifiers to a multi-class classifier. The classification of a test sample begins at the root node where the classification continues via the left or right edge depending on the result of a classifier in a node. In this way we get an evaluation path for the test sample from the root node to the leaf where the final class label for the test sample exists. In M class case we need only $M - 1$ comparisons in order to solve the final class for the test sample.

DDAG structure can also be represented on a list, where every node eliminates one class from the list. In this approach a test sample is evaluated against the node which is formed from the first and the last element on a list. A test sample obtains either the class label i or j from the node and the class label that the classifier gives for a test sample remains in the list and the other will be removed from the list. Hence, we get the same result that a test sample needs only $M - 1$ comparisons in order to solve the final class. DDAG has $\frac{M(M-1)}{2}$ nodes where in everyone there is an SVM (or k-NN) classifier. DAGSVM has some advantages. Firstly, the training phase is similar that of one-vs-one method and, hence, it is computationally lighter than one-vs-all method. Secondly, the evaluation phase is fast and we do not need to handle any tie situations, when the order

of the list (or DDAG) is fixed. DAGKNN uses the same learning structure as the DAGSVM but now in the node there is a k-NN classifier instead of an SVM classifier. An example of a four-class DAGSVM can be found from [5,9]. DAGSVM and DAGKNN also contain a disadvantage which exists in the graph construction itself. We can form the list (or DDAG) in different orders. For instance, if we have an M-class classification task, the list can be formed up to $M!$ different orders and each one of these can produce different results. One of the problems is to find the optimal order. Platt et al. [15] made some limited experimental tests with different orders and they did not notice any crucial differences between the results. Because the term $M!$ grows extremely fast when M increases, it is in practise impossible (or computationally very heavy) to go through every possible order. For example, in ten class cases list can be put to 3628800 different orders.

3 Experimental Tests

3.1 Data Description and Test Arrangements

Our dataset has altogether 50 species of benthic macroinvertebrates. Benthic macroinvertebrates were scanned three times and the scannings were identified with a label set1, set2 or set3 [18]. The following preprocessing steps were made according to data including all scannings. Firstly, we sorted the species in the data into decreasing order according to their group sizes. Secondly, we divided the species into five disjoint groups such that the number of samples of each species within a group would be as equal as possible. More specifically, ten largest species were chosen to group 1 and the next ten species were chosen to group 2 etc. until the ten smallest classes formed group 5. When the division was made, the final step was to take the samples from the first scanning for the classification. This procedure explains why in Tables 1 and 2 the sizes are not always in a decreasing order. From Tables 1 and 2 the species and their sizes can be seen. Moreover, from the [5,6,7,10,11,12,17,18] some example images of the benthic macroinvertebrates can be found.

Table 1. Species and their corresponding number of samples in groups 1, 2 and 3

Group 1		Group 2		Group 3	
Species	Size	Species	Size	Species	Size
Asellus aquaticus	328	*Leuctra*	176	*Micrasema gedium*	70
Baetis muticus	290	*Limnius volckmari*	167	*Ceratopogodinae*	61
Bithynia tentaculata	292	*Baetis rhodani*	136	*Caenis rivulorum*	70
Micrasema setiferum	291	*Cheumatopsyche lepida*	126	*Arctopsyche ladogensis*	65
Nemoura	246	*Hydropsyche pellucidulla*	117	*Ephemera mucronata*	55
Ephemara aurivillii	236	*Ephemera ignita*	116	*Sericostoma personatum*	60
Myxas glutinosa	228	*Hydraena*	113	*Caenis luctuosa*	66
Ceratopsyche silfvenii	222	*Ameletus inopinatus*	113	*Pisidium*	50
Elmis aenea	185	*Callicorixa wollastoni*	84	*Heptagenia sulphurea*	54
Baetis niger	181	*Habrophlebia*	81	*Chimarra marginata*	52

Table 2. Species and their corresponding number of samples in groups 4 and 5

Group 4		Group 5	
Species	Size	Species	Size
Tanypodinae	62	*Agapetus*	24
Leptophlebia	40	*Radix balthica*	24
Sigas semistriata	39	*Gyraulus*	20
Ceratopsyche nevae	36	*Heptagenia fuscogrissea*	19
Lepidostoma hirtum	37	*Baetis digitatus*	20
Atherix ibis	31	*Oulimnius tuberculatus larvae*	13
Oulimnius tuberculatus	31	*Athripsodes*	13
Gammarus lacustris	38	*Ophiogomphus cecilia*	11
Capnia	29	*Paraleptophlebia*	12
Dicranota	27	*Wormaldia subnigra*	11

Collected benthic macroinvertebrate samples were scanned by a flatbed scanner (HP Deskjet 4850) and saved in the JPG format. Features were extracted from the images and calculated with a public Java-based ImageJ program [4]. The data has 32 features altogether. Features can be divided into two categories: simple shape features and grey value features. Accurate definitions from every feature can be found from [4]. Features in the data consists of the same features as what are used in [5,6,7,10,11,12,18] and the features {FeretX, FeretY, FeretAngle, MinFeret, AR, Round, Solidity}. The final feature selection for the classification was made with the help of the scatter method [8]. The full description of the scatter method algorithm can be found from [8]. We ran the scatter method with ten iterations for every group case to ensure that the obtained features are appropriately chosen. Furthermore, a criteria for the selected features was that their separation power in the scatter method was atleast 0.1. With the scatter method we got the following feature sets:

- Group 1:{Min, Integrated Density, Area, Perimeter, Minor, Circularity, Feret, MinFeret, AR, Round, Solidity}
- Group 2:{Mean, Mode, Min, YM, Integrated Density, Median, Skewness, Area, Y, Perimeter, Major, Minor, Feret, FeretX, MinFeret, AR, Round}
- Group 3:{Mean, Standard Deviation, Mode, Min, Max, XM, YM, Integrated Density, Median, Skewness, Kurtosis, Area, X, Y, Perimeter, Width, Height, Major, Minor, Circularity, Feret, FeretX, MinFeret, AR, Round}
- Group 4:{Mean, Standard Deviation, Mode, Min, Max, YM, Integrated Density, Median, Skewness, Area, Y, Perimeter, Major, Minor, Circularity, Feret, MinFeret}
- Group 5:{Mean, Standard Deviation, Mode, Integrated Density, Median, Skewness, Area, Perimeter, BX, Major, Minor, Circularity, Feret, MinFeret, AR, Round, Solidity}.

Because the range of the sizes of the species alternate greatly, we applied different techniques. We used a crossvalidation technique in every classification such that in case of groups 1 and 2 we utilised 10 times 10-fold crossvalidation. For group 3 10 times 5-fold crossvalidation was used and in the case of groups 4 and 5 10 times 3-fold crossvalidation was applied. Hence, we got enough samples for each training and test sets. Before

representing the data for SVM and k-NN classifiers, we standardized the columns of the data matrix in every group case to have zero mean and unit variance. Other tranformations, such as normalization or principal component analysis or linear scalings, were not used because we wanted to perform the classification procedure as close as possible to the original input space. Thus, the classification is more truthfully. We used the same parameter spaces for different kernel parameters as in [6]. Thus, for box constraint, σ and κ parameter space was $\{0.5, 1.0, \ldots, 20.0\}$ and for δ in Sigmoid parameter space was $\{-20.0, -19.5, \ldots, -0.5\}$. Hence, the RBF and Sigmoid kernel functions were tested with 1600 parameter combinations and the linear and polynomial kernel functions (degrees of 2, 3, 4 and 5) were tested with 40 parameter values. Furthermore, we made an agreement of $\kappa = -\delta$ because, otherwise, the number of the parameter combinations in Sigmoid kernel function would have increased from 1600 to 64000.

Table 3. Classification times with DAGSVM

Kernel	Group 1	Group 2	Group 3	Group 4	Group 5
Linear	1h 54min 36s	29min 19s	4min 35s	1min 55s	1min 40s
Pol. $d = 2$	2h 27min 47s	35min 32s	5min 2s	1min 59s	1min 40s
Pol. $d = 3$	2h 45min 28s	38min 18s	5min 21s	2min 3s	1min 42s
Pol. $d = 4$	3h 3min 31s	41min 9s	5min 30s	2min 7s	1min 43s
Pol. $d = 5$	3h 19min 52s	42min 53s	6min 21s	2min 15s	1min 53s
RBF	191h 27min	41h 53min 20s	5h 41min 10s	1h 50min 37s	1h 26min 10s
Sigmoid	155h 30min 20s	35h 45min	4h 37min 35s	1h 40min 14s	1h 15min 11s

We performed all the experimental tests with Dell Latitude E6500 laptop having 4GB of memory and 2.8GHz Intel Core 2 Duo processor. From the Table 3 we can see how much time was spent to classifications with different kernel functions, when all parameter combinations were tested. For the DAGKNN we performed the classification with the odd k values, which were less or equal to smallest species size in the group. Furthermore, we repeated the classification procedure with the DAGKNN altogether with four different measures. These were standard Euclidean and cityblock metrics and correlation and cosine measures and the spent time for the DAGKNN classification can be seen from Table 4. Results show that DAGKNN is faster than DAGSVM, but we need to remember that in DAGKNN we tested less k values than the parameter combinations in DAGSVM. We used the binary SVM implementation of Bioinformatics Toolbox of Matlab as a basis for our tests and all tests were made with Matlab. We used the Least Square method [16] in finding the optimal hyperplane.

Optimal parameter values in DAGSVM were chosen the following way. We present it in a general way. We had $10 \times \mu$ disjoint training and test sets. Firstly, we trained each binary SVM using suitable subsets from the full training data. Secondly, we evaluated the accuracy of the training set by giving the full training set as a test set to trained SVMs. Thirdly, we evaluated the accuracy of the real test set with the trained SVMs. The final accuracy for the specific parameter combination was the average of the $10 \times \mu$ accuracies. Hence, for all parameter combinations we obtained a pair of values where

Table 4. Classification times with DAGKNN

Distance	Group 1	Group 2	Group 3	Group 4	Group 5
Euclidean	1h 24min 54s	11min 47s	2min 27s	36s	11s
Cityblock	1h 23min 26s	11min 45s	2min 26s	35s	11s
Correlation	1h 27min 29s	13min 10s	2min 43s	40s	12s
Cosine	1h 27min 9s	12min 48s	2min 40s	39s	12s

Table 5. Kernel parameter values

Kernel	Group 1	Group 2	Group 3	Group 4	Group 5
Linear	(20.0)	(19.5)	(1.5)	(18.5)	(5.5)
Pol. $d = 2$	(16.5)	(7.0)	(0.5)	(0.5)	(0.5)
Pol. $d = 3$	(0.5)	(0.5)	(0.5)	(0.5)	(0.5)
Pol. $d = 4$	(0.5)	(0.5)	(0.5)	(0.5)	(0.5)
Pol. $d = 5$	(0.5)	(0.5)	(0.5)	(1.0)	(0.5)
RBF	$(20.0, 1.0)$	$(20.0, 1.5)$	$(20.0, 1.5)$	$(20.0, 3.5)$	$(18.0, 3.5)$
Sigmoid	$(19.5, 20, -20)$	$(20, 19.5, -19.5)$	$(9.5, 19.5, -19.5)$	$(17.5, 10, -10)$	$(16, 4.5, -4.5)$

the first element was the mean accuracy of the training sets and the second element was the mean accuracy of the test sets.

Overfitting is always an existent problem when using SVM. If too large parameter values are given to SVMs, a model becomes too complex and its generalization ability weaken. Thus, the classification error in a training set tends to zero and in a test set it tends to one. Hence, the final parameters were chosen with an easy method. We calculated

$$\arg\min_i[(1 - ACC_{TRAIN,i}) + 2 \cdot (1 - ACC_{TEST,i})]$$

where i is the index for parameter combination and ACC is the accuracy. In other words we sought that parameter combination index which gained the minimum of the weighted sum of the training and test set classification errors. Weightening was made in order to prevent possible tie situations and to separate those parameters which caused overfitting. By this means we do not always get those parameters which give the best accuracy in the test set, but we get a compromise where the accuracy of a training set is determined from a full training set and we take also into account the accuracy of the test set. Other possible ways to determine the best parameter values are nested cross-validation which is time-consuming or to use a validation set technique. In both cases a disadvantage is that the optimal parameter values are determined from a smaller set than the actual training data. In Table 5 we see the obtained kernel parameter values from each group. In kernel functions from the linear to the 5th degree of polynomial kernel function, we had only one parameter, box constraint, and it is in the parenthesis. In RBF the first value is the box constraint and the second one is σ. In Sigmoid the first value in the parenthesis is the box constraint, the second value is κ and the last one is δ.

3.2 Results

In the following tables we have compressed the results such that when the results of DAGSVM are presented, every row in the result table indicates the classification rates with a specific kernel function. When DAGKNN is in question, a row of the result table indicates the classification rates with specific distance alternative and k value. Furthermore, the last column in the result tables depicts accuracy obtained from a specific kernel function or specific distance alternative with some k value for easyning the analysis of the results. Class labels in result tables are abbreviations from the latin-based names of the species in Tables 1 and 2. We boldfaced the best classification rate (or classification rates in the case of tie situations) from each column of the result tables for facilitating analysis. Sigmoid kernel function was the worst kernel function in each group so we do not take it into account in our analysis. Reasons behind the poor results with Sigmoid kernel funtions may lie in the preprocessing. Normalization of the data or linear scaling to interval $[-1, 1]$ or $[0, 1]$ after the stardardization could have increased the general level of Sigmoid, but every transformation what we make to the data draws the situation more away from the original input space. Throughout all classification results with DAGKNN small k values (odd integers from 1 to 9) gave the best results. From Tables 6 and 7 we find the results when DAGKNN and DAGSVM

Table 6. DAGKNN: Results (%) with different distance alternatives and k values in group 1

		ASE	MUT	BIT	SET	NEM	AUR	MYX	SIL	ELM	NIG	Mean accuracy
Euclidean	$k=5$	78.5	71.5	88.9	90.4	47.3	65.9	84.2	96.2	**89.4**	61.2	77.7
	$k=7$	78.9	72.9	89.4	89.7	47.8	65.8	83.1	**96.5**	87.9	60.7	77.7
	$k=9$	78.8	73.2	89.2	89.6	47.7	64.7	83.1	96.0	88.4	63.1	77.7
Cityblock	$k=5$	**80.4**	73.6	88.4	**91.3**	49.3	**66.2**	84.1	95.7	88.9	**66.5**	**78.7**
	$k=7$	78.1	**75.2**	89.6	91.1	49.1	**66.2**	84.5	96.1	88.1	66.2	**78.7**
	$k=9$	77.8	**75.2**	**90.6**	90.3	48.9	66.1	84.0	95.5	86.9	**66.5**	78.5
Correlation	$k=3$	79.4	64.4	87.1	89.7	**54.8**	62.3	79.4	94.8	74.9	56.2	75.1
	$k=5$	77.1	66.8	87.1	89.5	54.5	62.5	81.9	94.2	77.8	54.3	75.3
	$k=7$	76.1	68.7	88.2	89.4	53.8	62.3	81.3	95.4	76.1	55.0	75.4
Cosine	$k=3$	77.6	66.8	85.6	90.6	53.6	64.1	82.2	94.8	87.1	60.3	76.6
	$k=5$	75.9	67.0	85.6	91.1	53.7	63.6	83.3	95.4	87.9	59.6	76.5
	$k=7$	73.8	67.9	86.0	90.4	54.5	64.4	**84.6**	96.3	88.3	60.9	76.8

were used to classify group 1. The DAGKNN achieved very similar accuracies with all measures. Accuracies in Table 6 were within 4% interval, but the best accuracy, nearly 79%, was obtained by the cityblock metric when $k = 5$ and $k = 7$. Classes BIT, SET and SIL were identified above 90% classification rate and from these classes SIL was recognized with classification rate over 95% which is a very good result. Class ELM got nearly 90% classification rate and other classes which obtained above 80% classification rate were MUT and MYX. The poorest results were in the classes NEM, AUR and NIG. These classes had below 67% classification rates and especially NEM clearly separated from other classes having below 55% classification result.

Table 7. DAGSVM: Results (%) with different kernel functions in group 1

	ASE	MUT	BIT	SET	NEM	AUR	MYX	SIL	ELM	NIG	Mean accuracy
Linear	77.8	71.2	77.2	96.0	67.6	71.0	86.8	93.1	89.3	76.0	80.3
Pol. $d = 2$	**85.0**	**80.3**	92.4	97.2	**73.5**	**77.6**	**93.5**	**96.6**	93.5	76.5	**86.7**
Pol. $d = 3$	84.7	78.8	**94.1**	97.4	71.0	76.4	92.5	96.4	**94.5**	**76.8**	86.4
Pol. $d = 4$	78.3	75.8	91.5	**98.0**	70.3	69.5	89.2	83.4	93.6	68.1	82.1
Pol. $d = 5$	61.8	66.9	86.3	97.0	64.5	58.0	86.2	56.1	93.0	57.1	73.0
RBF	84.6	76.7	92.3	96.4	66.6	73.6	90.4	96.2	93.3	**76.8**	84.8
Sigmoid	0.7	21.0	34.8	67.5	32.3	33.6	77.4	57.5	38.7	13.5	36.8

DAGSVM succeeded in the classification of group 1 better than DAGKNN. Five from the seven kernel funtions obtained over 78.7% accuracy. Particularly the quadratic kernel function obtained a high accuracy being 86.7%. Almost the same accuracy was the cubic kernel function which was only 0.3% inferior to the quadratic kernel function. The third noteworthy kernel was RBF which obtained nearly 85% accuracy. The same classes, as in DAGKNN, NEM, AUR and NIG were the hardest classes to identify. Classes BIT, SET, MYX, SIL and ELM got very high classification rates, since the results of these classes were over 93%. Classes NEM, AUR and NIG were the three hardest classes to classify as in the corresponding DAGKNN case.

Table 8. Results (%) with different distance alternatives and k values in group 2

		LEU	LIM	BAE	CHE	PEL	IGN	HYD	AME	CAL	HAB	Mean accuracy
Euclidean	$k = 3$	74.7	87.9	66.8	78.4	68.9	**68.4**	99.9	61.4	91.5	**43.8**	75.1
	$k = 5$	75.5	**89.2**	76.8	78.6	68.5	68.0	99.6	**63.6**	94.2	40.2	**76.5**
	$k = 7$	74.5	88.7	**77.4**	**79.1**	67.2	63.1	**100.0**	62.7	93.9	35.7	75.5
Cityblock	$k = 5$	75.5	87.5	71.7	76.9	69.0	65.5	**100.0**	60.7	**97.6**	38.0	75.3
	$k = 7$	73.5	88.1	74.9	76.9	68.4	64.9	**100.0**	60.8	96.0	35.1	75.0
	$k = 9$	72.3	88.0	76.8	78.7	66.6	64.2	**100.0**	60.5	96.2	32.3	74.8
Correlation	$k = 3$	80.1	85.7	66.6	77.5	68.1	58.8	99.2	55.6	88.5	30.9	72.8
	$k = 5$	**81.6**	86.9	71.3	78.4	68.1	57.9	98.4	52.5	87.3	26.4	73.0
	$k = 7$	81.2	85.6	70.3	78.8	66.2	55.2	98.2	52.6	88.2	28.1	72.4
Cosine	$k = 3$	80.9	86.0	72.5	77.3	**73.3**	58.1	98.4	56.5	90.7	43.0	75.0
	$k = 5$	80.1	86.2	69.8	77.6	71.5	55.5	98.2	56.7	92.1	43.4	74.3
	$k = 7$	81.0	85.6	68.2	76.2	70.6	53.4	98.2	57.8	91.5	39.2	73.6

Tables 8 and 9 show the compelling results for group 2. Compared to Table 6 the general level of classification decreased a bit. Now, the best accuracy was gained by the Euclidean metric when $k = 5$, but all the accuracies still were within 5% range. An eye-catching detail was that the class HYD obtained a perfect 100% classification rate four times when using DAGKNN, but in Table 9 we do not see any kernel function which would have managed to do this. Another very well recognized class was CAL which had almost 98% classification rate. Furthermore, classes LEU and LIM achieved above 80%

Table 9. Results (%) with different kernel functions in group 2

	LEU	LIM	BAE	CHE	PEL	IGN	HYD	AME	CAL	HAB	Mean accuracy
Linear	80.5	92.6	65.5	79.5	71.8	67.1	99.1	61.4	92.3	74.8	78.7
Pol. $d = 2$	80.0	**94.7**	74.6	80.0	71.7	**73.8**	99.2	**69.4**	**96.3**	72.5	81.4
Pol. $d = 3$	75.7	91.0	73.7	**81.1**	62.3	70.2	99.2	68.0	85.3	70.5	78.1
Pol. $d = 4$	53.5	81.2	61.5	67.9	43.2	47.4	99.6	59.8	71.3	50.4	63.9
Pol. $d = 5$	35.2	73.0	48.3	60.3	25.5	50.3	98.4	50.6	70.9	35.2	54.5
RBF	**82.6**	92.4	**75.5**	80.9	**83.6**	71.3	**99.8**	65.8	91.9	**75.2**	**82.2**
Sigmoid	27.8	7.1	37.7	43.3	44.6	24.4	94.5	49.4	72.3	4.4	38.5

classification rates which is always a good result. Other classes obtained below 80% classification rates and especially HAB distinguished from all classes having clearly under 50% classification rate. Also, classes IGN and AME were quite poorly recognized since they obtained below 70% classification rates. Rest of the classes were identified above 70%, but still under 80% classification rates.

Table 10. Results (%) with different distance alternatives and k values in group 3

		GED	CER	CAE	ARC	MUC	SER	LUC	PIS	SUL	CHI	Mean accuracy
Euclidean	$k = 1$	**95.7**	**98.4**	77.4	97.0	55.5	74.2	75.9	90.6	67.5	87.5	82.4
	$k = 3$	95.1	**98.4**	77.7	97.1	51.7	78.7	79.6	87.9	70.0	90.6	83.1
	$k = 5$	93.5	**98.4**	76.0	98.5	51.1	77.7	**81.1**	88.1	71.2	89.3	82.9
Cityblock	$k = 1$	95.3	**98.4**	77.5	98.2	**56.4**	76.0	78.9	89.2	73.1	84.8	83.2
	$k = 3$	94.7	**98.4**	78.3	97.9	54.5	**79.3**	79.0	87.4	71.0	**91.4**	**83.6**
	$k = 5$	94.0	**98.4**	77.7	98.5	53.3	78.5	79.3	89.0	**73.3**	89.9	83.5
Correlation	$k = 1$	93.9	**98.4**	78.0	99.2	54.0	71.5	69.4	86.7	56.9	83.9	79.8
	$k = 3$	94.7	98.2	80.5	**100.0**	51.8	73.2	74.2	85.7	44.3	84.5	79.6
	$k = 5$	95.1	98.2	77.5	**100.0**	48.1	76.4	77.3	85.7	44.3	79.1	79.1
Cosine	$k = 1$	94.2	**98.4**	79.0	99.8	56.3	68.0	73.6	**90.7**	50.3	84.6	80.1
	$k = 3$	95.5	**98.4**	**80.7**	**100.0**	51.4	71.8	79.0	86.3	48.8	87.0	80.7
	$k = 5$	95.1	**98.4**	76.5	**100.0**	49.3	73.7	81.0	87.8	48.0	84.4	80.2

When in group 1 five kernel functions achieved better accuracies than the maximum accuracy with DAGKNN, now in group 2 four kernel functions had better accuracies than the maximum accuracy, 76.5%, with the DAGKNN. However, the best kernel functions were again the quadratic and RBF kernels which obtained classification rates over 80%. The RBF obtained the highest classification rate being 82.2%. Class HYD had a nearly perfect score and was clearly the most distinguished class together with CAL and LIM among all classes. Below 90% but above 80% classification rates were obtained for LEU, CHE and PEL. The lowest identifications were attained to the classes BAE, IGN, AME and HAB which got below 80% classification rates. When doing a search for equal classwise results between Tables 8 and 9, our eyes are focused on classes LEU, BAE, CHE, HYD and CAL. Other classes obtained larger differences. The general level

of DAGSVM results was lower than in Table 7 and this tendency was seen analogously in DAGKNN results. It needs to remember that the results between groups are not directly comparable since they consist of a totally different species and the group sizes vary greatly between groups.

Next we have classification results in group 3. In Table 10 we obtained interesting results. Firstly, the level of classification between different species was widely spread. Secondly, classes GED, CER, ARC, PIS and CHI were identified very well having above 90% classification rates and, specially CER was recognized with a nearly perfect score with all measures and ARC obtained a perfect classification when using correlation and cosine measures. Classes CAE and LUC got above 80% classification rates with cosine and Euclidean measures. Other classes were left below 80% classification rate and the most difficult class to recognize was MUC having the maximum classification rate below 60%. Accuracies were also separated since Euclidean and cityblock metrics achieved accuracies over 82%, but cosine and correlation measures were left below 81% accuracies.

Table 11. Results (%) with different kernel functions in group 3

	GED	CER	CAE	ARC	MUC	SER	LUC	PIS	SUL	CHI	Mean accuracy
Linear	92.7	**96.7**	**81.3**	**98.5**	**66.2**	79.9	73.4	89.6	**84.8**	88.9	85.3
Pol. $d = 2$	**95.0**	**96.7**	78.6	97.4	61.9	72.5	**79.7**	**93.2**	73.7	83.5	83.5
Pol. $d = 3$	92.3	95.6	79.9	78.7	58.3	66.3	71.6	86.1	66.2	72.3	77.2
Pol. $d = 4$	89.2	91.4	75.8	64.4	53.9	64.9	65.7	82.7	69.2	75.4	73.5
Pol. $d = 5$	77.8	87.6	72.9	54.2	52.4	62.4	63.7	78.0	65.9	69.0	68.5
RBF	93.1	**96.7**	80.9	98.3	61.3	**81.9**	**79.7**	91.7	81.2	**92.3**	**85.9**
Sigmoid	52.7	59.0	49.6	93.1	15.4	32.8	45.0	81.6	17.8	7.8	46.5

When compared DAGKNN accuracies with the DAGSVM accuracies, DAGKNN accuracies were left behind the DAGSVM accuracies. The linear and RBF kernel functions gained better accuracies than the maximum accuracy in the DAGKNN results. Three kernel functions of the seven possible were distinguished from the rest. These were the linear, quadratic and RBF kernel functions and these three kernel functions were also the best ones in Table 11. It seems that the quadratic and RBF kernels are the best choices for this classification task. In classwise examination, again, classes GED, CER, ARC, PIS and CHI were the best classifiable classes. This is consistent with the DAGKNN results. Furthermore, in the cases of CAE, SER and LUC differences between the best results of DAGKNN and DAGSVM were not large. The most significant differences came in classes MUC and SUL. Class MUC was identified almost 10% better with DAGSVM than with DAGKNN. Moreover, class SUL was recognized over 11% better in DAGSVM with the linear kernel function. With the linear and RBF kernel functions above 85% accuracy was achieved when the quadratic kernel obtained 83.5% result. The cubic, 4th and 5th degree of the polynomial kernel functions and Sigmoid got below 80% accuracies.

DAGKNN succeeded in the classification of the fourth group quite similarly to group 1 classification. The Euclidean and cityblock metrics, when $k = 3$ or $k = 5$, got 79%

Table 12. Results (%) with different distance alternatives and k values in group 4

		TAN	LEP	SIG	NEV	HIR	IBI	OUL	GAM	CAP	DIC	Mean accuracy
Euclidean	$k=1$	86.1	71.8	95.3	93.9	59.3	51.3	88.0	**89.6**	65.6	**70.4**	78.4
	$k=3$	90.6	80.9	95.3	93.0	54.6	60.0	94.3	88.3	60.3	63.7	79.8
	$k=5$	93.0	80.0	96.0	94.2	48.2	51.3	97.7	84.8	60.8	65.4	79.0
Cityblock	$k=3$	90.9	**83.1**	95.8	93.8	57.1	56.0	97.8	88.8	63.1	58.6	**80.4**
	$k=5$	92.7	80.7	96.3	95.0	48.6	47.0	98.4	84.5	63.4	63.1	78.9
	$k=7$	92.9	82.2	97.1	95.8	43.2	40.9	97.8	82.4	62.1	61.4	77.7
Correlation	$k=1$	89.0	63.1	89.9	92.2	56.2	**61.9**	93.3	89.3	**67.5**	46.0	76.6
	$k=3$	94.5	60.3	91.9	93.1	58.0	56.5	99.7	87.2	60.7	47.4	77.1
	$k=5$	**96.9**	58.9	94.5	94.2	55.7	47.4	**100.0**	85.3	61.6	55.2	77.2
Cosine	$k=1$	88.3	67.8	90.1	93.2	**63.2**	59.3	95.8	**89.6**	64.7	52.2	78.1
	$k=3$	91.0	69.6	**97.4**	**96.6**	58.0	55.2	99.7	88.8	58.1	48.4	78.4
	$k=5$	93.2	72.0	96.6	93.9	48.7	48.1	**100.0**	84.7	61.9	51.2	77.2

or higher accuracy, but the difference between the worst and the best accuracy was only less than 4%, so the same trend continued in group 4 classification than in the previous ones. Three from seven kernel functions achieved better accuracy than DAGKNN with the cityblock metric when $k = 3$. In this particular case DAGKNN obtained an accuracy over 80%. The linear, quadratic and RBF kernels performed above 83% accuracy which is a very good result and the highest accuracy 86.3% was reached by the RBF. When examining the DAGKNN results more closely, we noticed that a large part of the highest classification rates were among the correlation and cosine results.

Table 13. Results (%) with different kernel functions in group 4

	TAN	LEP	SIG	NEV	HIR	IBI	OUL	GAM	CAP	DIC	Mean accuracy
Linear	90.6	72.0	88.1	96.9	70.4	**77.3**	**94.7**	89.8	**81.4**	75.0	84.2
Pol. $d=2$	88.5	83.3	93.0	**97.5**	70.1	68.3	93.5	80.9	78.5	70.0	83.2
Pol. $d=3$	80.5	78.9	81.3	90.3	53.8	64.4	91.6	67.8	64.3	54.4	73.8
Pol. $d=4$	68.2	73.0	69.8	77.8	50.4	55.9	85.8	62.3	63.7	51.6	66.3
Pol. $d=5$	65.2	68.2	63.8	70.5	47.1	48.7	81.6	59.7	61.0	52.2	62.2
RBF	**93.7**	**83.6**	**94.5**	95.6	**73.7**	71.9	93.9	**91.7**	75.7	**78.3**	**86.3**
Sigmoid	72.4	44.5	41.5	84.5	33.8	36.6	58.8	42.8	34.2	25.3	49.7

The best classification results can be divided into three categories. Firstly, the classes TAN, SIG, NEV and OUL got excellent classification rates being above 96%. Especially, in the case of class OUL we obtained a perfect 100% classification rate with correlation and cosine measures. Compared to DAGSVM, where the classes also achieved above 90% classification rates, there are not any perfect scores. Secondly, in the DAGKNN classification rates of classes LEP and GAM were located into interval 83%-90%. The best classification rate of class LEP in the DAGSVM was nearly identical with the classification rate in DAGKNN. Thirdly, the rest of the classes in DAGKNN had below 71% classification rates and these classes were consistently classified better with DAGSVM. The RBF kernel function in Table 13 obtained six topmost classification rates

Table 14. Results (%) with different distance alternatives and k values in group 5

		AGA	RAD	GYR	FUS	DIG	TUB	ATH	OPH	PAR	WOR	Mean accuracy
Euclidean	$k=1$	90.7	84.7	62.8	81.2	71.5	65.8	46.3	72.7	56.0	**88.8**	73.8
	$k=3$	**93.7**	75.4	76.0	85.8	80.0	65.2	49.0	54.9	56.8	86.4	75.0
	$k=5$	91.2	80.3	82.1	86.4	84.1	62.1	44.7	46.9	53.0	82.0	75.0
Cityblock	$k=1$	90.3	**88.0**	78.3	85.7	81.5	66.5	**63.2**	67.4	56.0	83.8	78.4
	$k=3$	89.9	80.4	81.2	**88.0**	87.8	74.2	55.7	62.4	**58.5**	**88.8**	**79.0**
	$k=5$	89.4	82.5	**84.1**	85.3	86.6	69.2	46.8	47.2	52.2	**88.8**	76.5
Correlation	$k=1$	85.6	72.4	65.3	74.8	85.6	64.7	40.5	**73.0**	44.3	74.8	70.4
	$k=3$	83.5	69.0	73.4	79.5	**94.4**	76.5	38.8	70.5	53.7	84.5	73.9
	$k=5$	78.4	68.3	76.8	76.4	92.3	**80.0**	29.0	72.2	34.4	83.9	71.2
Cosine	$k=1$	89.0	71.6	61.7	76.7	85.2	61.5	47.3	71.3	56.0	82.1	72.0
	$k=3$	84.8	71.7	73.8	81.1	93.0	76.7	46.5	70.5	56.0	87.3	75.6
	$k=5$	82.6	69.6	78.0	77.5	91.8	77.7	33.0	71.5	38.2	86.7	72.7

from all classes and the rest of the classes were classified with the highest classification rates when the linear and quadratic kernel function were used. More details about the results of group 4 can be found from Tables 12 and 13.

Table 15. Results (%) with different kernel functions in group 5

	AGA	RAD	GYR	FUS	DIG	TUB	ATH	OPH	PAR	WOR	Mean accuracy
Linear	**91.8**	75.8	82.5	83.9	**86.9**	**84.5**	**86.5**	69.6	**74.4**	80.7	82.6
Pol. $d=2$	85.7	78.3	78.1	**89.3**	83.4	69.8	75.8	51.2	60.9	**87.7**	77.9
Pol. $d=3$	84.5	62.3	75.1	87.6	77.9	70.3	73.0	32.7	47.8	71.7	70.7
Pol. $d=4$	82.8	57.0	68.6	87.2	70.7	62.5	69.2	22.9	48.8	65.0	66.1
Pol. $d=5$	76.8	57.0	60.8	85.9	66.1	55.7	70.3	26.6	52.2	64.6	63.7
RBF	85.6	**89.1**	**87.1**	83.2	86.6	79.7	72.2	**70.7**	74.1	**87.7**	**82.9**
Sigmoid	27.4	60.3	64.0	68.9	46.7	51.4	21.8	54.4	16.2	31.6	46.4

In the last group the classification level of DAGKNN was spread out more wide interval than in the previous cases. Table 14 shows that the accuracies were spread from 70.4% to 79% which was achieved by the cityblock metric. The highest classification rates were obtained among the Euclidean, cityblock and correlation distances. Two of ten classes were identified above 90% classification rate and these two classes AGA and DIG gained 93.7% and 94.4% results. Classes RAD, FUS and WOR were recognized with classification rate 88% or 88.8%. Also, classes GYR and TUB gained classification rates of 80% or higher. Classes ATH, OPH and PAR were the hardest classes to classify and from these OPH gained 73% and the rest were left below 64% results. Compared to DAGKNN, DAGSVM (see Table 15) did not succeeded much better. Now, only two of seven kernel functions got higher accuracies than 79%. These were the linear and RBF kernel functions and their corresponding accuracies were 82.6% and 82.9%. The linear kernel obtained five times the topmost classification rate among all classes and the RBF got four times the topmost results. Only the class AGA was classified with a

classification rate over 90% in DAGSVM, in DAGKNN there were two classes which got above 90% classification rate. Classes ATH, OPH and PAR were also the hardest classes to identify as it was in DAGKNN case. The best classification rates of the rest of the classes were between 84.5% and 89.1%. Accuracies of the linear and RBF kernel function were very close to each other since they had only 0.3% difference. The linear and RBF kernel functions were the only ones that achieved above 80% accuracies when taking into account also the results of DAGKNN.

4 Discussion

We applied in this paper DDAG learning structure to SVM and k-NN classifiers. DAGSVM was applied to benthic macroinvertebrate identification in [5] with great success and it inspired us to examine how DAGKNN succeeds in this classification problem compared to DAGSVM. Generally, in all groups the linear, quadratic and RBF kernel functions and from these kernels especially the quadratic and RBF showed their power in this classification task. DAGKNN method did not manage to obtain higher accuracies than DAGSVM, but it is still a very comparable classification method since the simplicity of k-NN compared to SVM is from the pratical and computational point of view much more user-friendly. The DAGKNN method contains only two parameters: the choice of distance and k value. In SVM the choice of a kernel function and the tuning of the parameters are the key factors for successful classification. How to find the right kernel and the right parameter values can be computationally demanding problem as Table 6 showed. How to speed up the parameter tuning and how to choose the right kernel functions are problems to be researched more closely in the future. Also, we need to examine how other machine learning methods such as Linear Discriminant Analysis or Naïve Bayes manage in benthic macroinvertebrate classification when using the DDAG learning structure. Furthermore, other multi-class methods of SVMs and classification methods need to be considered in the further research. Feature selection is an important factor in classification problems. In this paper we solved this problem by using the scatter method [8] which is a novel approach. From the results we can conclude that the scatter method is a valid feature selection method for the classification of benthic macroinvertebrates. Results also showed that the fully automated benthic macroinvertebrate identification is possible when the classifiers are tuned up.

Acknowledgments. We thank Finnish Environment Institute, Jyväskylä, Finland, for the data. The first author is also thankful to the Tampere Graduate Program in Information Science and Engineering for the support. We also want to thank Markku Siermala, Ph.D., for the scatter method.

References

1. Christiani, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press, Cambridge (2000)
2. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20, 273–297 (1995)
3. Cover, T.M.: Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Transactions on Electronic Computers EC-14 (1965)

4. ImageJ: public domain Java-based image processing program, http://rsbweb.nih.gov/ij/docs/index.html
5. Joutsijoki, H., Juhola, M.: Automated benthic macroinvertebrate identification with decision acyclic graph support vector machines. In: Proceedings of the 2nd IASTED International Conference Computational Bioscience (CompBio 2011), Cambridge, United Kingdom, pp. 323–328 (2011)
6. Joutsijoki, H., Juhola, M.: Comparing the one-vs-one and one-vs-all methods in benthic macroinvertebrate image classification. In: Perner, P. (ed.) MLDM 2011. LNCS (LNAI), vol. 6871, pp. 399–413. Springer, Heidelberg (2011)
7. Joutsijoki, H., Juhola, M.: Kernel selection and its consequence to the number of ties in majority voting method. Artificial Intelligence Review (in press), doi:10.1007/s10462-011-9281-3
8. Juhola, M., Siermala, M.: A scatter method for data and variable importance evaluation. Integrated Computer-Aided Engineering 19(2), 137–149 (2012)
9. Kahsay, L., Schwenker, F., Palm, G.: Comparison of Multiclass SVM Decomposition Schemes for Visual Object Recognition. In: Kropatsch, W., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 334–341. Springer, Heidelberg (2005)
10. Kiranyaz, S., Gabbouj, M., Pulkkinen, J., Ince, T., Meissner, K.: Classification and Retrieval on Macroinvertebrate Image Databases using Evolutionary RBF Neural Networks. In: Proceedings of the International Workshop on Advanced Image Technology, Kuala Lumpur, Malaysia (2010)
11. Kiranyaz, S., Ince, T., Pulkkinen, J., Gabbouj, M., Ärje, J., Kärkkäinen, S., Tirronen, V., Juhola, M., Turpeinen, T., Meissner, K.: Classification and retrieval on macroinvertebrate image databases. Computers in Biology and Medicine 41(7), 463–472 (2011)
12. Kiranyaz, S., Gabbouj, M., Pulkkinen, J., Ince, T., Meissner, K.: Network of evolutionary binary classifiers for classification and retrieval in macroinvertebrate databases. In: Proceedings of 2010 IEEE 17th International Conference on Image Processing (ICIP), pp. 2257–2260 (2010)
13. Larios, N., Deng, H., Zhang, W., Sarpola, M., Yuen, J., Paasch, R., Moldenke, A., Lytle, D.A., Correa, S.R., Mortensen, E.N., Shapiro, L.G., Dietterich, T.G.: Automated insect identification through concatenated histograms of local appearance features: feature vector generation and region detection for deformable objects. Machine Vision and Applications 19(2), 105–123 (2008)
14. Lytle, D.A., Martinez-Muñoz, G., Zhang, W., Larios, N., Shapiro, L., Paasch, R., Moldenke, A., Mortensen, E.N., Todorovic, S., Dietterich, T.G.: Automated processing and identification of benthic invertebrate samples. Journal of North American Benthological Society 29(3), 867–874 (2010)
15. Platt, J.C., Christiani, N., Shawe-Taylor, J.: Large margin dags for multiclass classification. In: Advances in Neural Information Processing Systems (NIPS 1999) 12, Denver, USA, pp. 547–553 (2000)
16. Suykens, J.A.K., Vandewalle, J.: Least squares support vector machine classifiers. Neural Processing Letters 9, 293–300 (1999)
17. Tirronen, V., Caponio, A., Haanpää, T., Meissner, K.: Multiple Order Gradient Feature for Macro-Invertebrate Identification Using Support Vector Machines. In: Kolehmainen, M., Toivanen, P., Beliczynski, B. (eds.) ICANNGA 2009. LNCS, vol. 5495, pp. 489–497. Springer, Heidelberg (2009)
18. Ärje, J., Kärkkäinen, S., Meissner, K., Turpeinen, T.: Statistical classification and proportion estimation - an application to macroinvertebrate image database. In: 2010 IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2010), Kittilä, Finland, pp. 373–378 (2010)

Lung Nodules Classification in CT Images Using Shannon and Simpson Diversity Indices and SVM

Leonardo Barros Nascimento[1], Anselmo Cardoso de Paiva[2], and Aristófanes Corrêa Silva[1]

[1] Federal University of Maranhão - UFMA,
Post-graduate Program in Engineering of Electricity
Av. dos Portugueses, SN, Campus do Bacanga, Bacanga 65085-580, São Luis, MA, Brazil
lbarros.nascimento@gmail.com, paiva@deinf.ufma.br
[2] Federal University of Maranhao - UFMA, Department of Computer Science
Av. dos Portugueses, SN, Campus do Bacanga, Bacanga 65085-580, Sao Luís, MA, Brazil
ari@dee.ufma.br

Abstract. In this work, we present the use of Shannon and Simpson Diversity Indices as texture descriptors for lung nodules in Computerized Tomography (CT) images. These indices will be proposed to characterize the nodules into two classes: benign or malignant. The investigation is done using the Support Vector Machine (SVM) for classification in a dataset consisting of 73 nodules, 47 benign and 26 malignant; the results of the methodology were: sensitivity of 85.64%, specificity of 97.89% and accuracy of 92.78%.

Keywords: Lung Nodules, Computer-aided diagnosis (CADx), Shannon and Simpson Diversity Indices.

1 Introduction

The causes of cancer can be very diverse; they can be external or internal to the organism. The external causes are related to the environment and to the habits of a sociocultural environment. The internal causes are, in most cases, genetically predetermined, and are connected to the ability of the organism to fight against external aggressions. These factors can interact in a variety of ways, increasing the probability of malignant transformations in normal cells [1].

The appearance of cancer depends on the intensity and time of exposition of the cells to the carcinogenic agents. For example, the risk of a person developing lung cancer is directly proportional to the number of cigarettes smoked per day and to the number of years smoking [1].

Lung cancer is the commonest type of malignant tumors, presenting an increase of 2% per year in the worldwide incidence. 90% of the diagnosed cases are related to the use of tobacco and its derivatives [1].

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 454–466, 2012.

In Brazil, the estimative for new cases of lung cancer for 2011 were of 27,630, from which 17,800 occur among men and 9,830 among women [2].

The easiest way of diagnosing lung cancer is through an x-ray of the chest, associated to a computerized tomography. These technologies can help specialists in the early detection of lung cancer. Also, through the use of information from the images, it is possible to develop computer aided systems that help diagnosing the nodule and, act as a second opinion to the analysis of the exams. CAD systems are those that help specialists in the detection of regions of interest in the exam, but do not make the diagnosis. CADx systems do suggest a diagnosis (malignant or benign, for example). These systems used image processing techniques to make the auxiliary diagnosis.

In image processing many computational methodologies have been developed for the task of detection and diagnosing of lung cancer using CT exams. In [3], a set of three geometrical features was evaluated as a form of distinguishing between nodules and non-nodules using Support Vector Machine (SVM) as classifier. The results achieved by them were 100% of correct classification. In [4], a computer aided lung nodule detection scheme based on enhanced analysis of voxel in CT image was presented. The best results were obtained with sensitivity = 0.9375, accuracy = 0.8782 and specificity = 0.8766. In [5], it is proposed a new segmentation algorithm based on region growing for detection of lung nodules in CT images. The experimental results showed that the method is robust and promising, achieving sensitivity of 80.9%, with 0.23 false positives per slice. In [6], an alternative method of diagnosing malignant lung nodules through their shape is proposed. Preliminary experiments on 109 lung nodules (51 malignant and 58 benign) resulted in the 94.4% of correct classification (for the 95% confidence interval). In [7] is proposed method for diagnosis of malignant nodules using a proprietary base containing 109 nodules (51 malignant and 58 benign). Using the spatial distribution of intensities of images resulted in 96.3% accuracy in classification.

Diversity indices have been used in texture analysis to provide discriminant information among lung nodules. In Silva [8], Simpson's index, associated to the three geometrical features of Souza [3], was proposed for the classification with respect to the nature of the nodule (malignant or benign) and achieved accuracy of 100%, specificity of 100% and sensitivity of 100%.

In this paper, we propose a methodology for lung nodule diagnosis through the use of Shannon and Simpson's indices, which will be an aid in the evaluation of the diagnosis by the doctor.

This work is organized as follows: in Section 2, we describe how the images were obtained, the preprocessing of the images through histogram equalization, the segmentation of the nodules in the images, and present a discussion about the use of Diversity Indices (Shannon and Simpson's Index) as discriminant measurement in the diagnosis of lung nodules, the selection of features and the use of SVM as classification techniques. In Section 3, we present the experiments. In Section 4, the results are presented and discussed. Finally, we present the conclusions in Section 5.

2 Methodology

In this work the proposed methodology follows the steps seen in Fig. 1. The first step is the acquisition of the image, which was obtained from a patient's chest CT exam. Step 2 is the preprocessing of the image through histogram equalization. Step 3 is the segmentation of the tri-dimensional volume of the nodule. Step 4 is the extraction of representative features of the nodules, through the use of the Shannon and Simpson's indices. In step 5, the features are selected using LDA. Right after that, the classification of the nodules into benign or malignant is determined by the SVM. The last step is the validation of results.

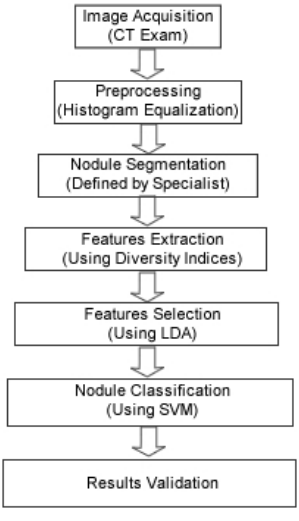


Fig. 1. Stages of the methodology

2.1 Image Acquisition

In this work, we used images made available by the US National Cancer Institute (NCI), created from a repository of images, which is the result of the formation of a consortium of institutions, known as LIDC (Lung Image Database Consortium) [9].

The Lung Image Database Consortium - LIDC is a group which aims to establish standard formats and management processes for pulmonary images, technical reports and clinic data necessary to the development, training and evolution of algorithms intended to detect and diagnose lung cancer.

In the LIDC base, all the images are in the DICOM format and have 16 bits per voxel. The base supplies an XML file with contour information for the slices, some features such as sphericity, texture, and malignancy with values ranging from 1 to 5 for lung nodules larger than 3 mm. And, only the information about the centroid for nodules smaller than 3 mm. So, from the 84 exams available in this base, only 58 exams present contour information. A contour is formed by pairs of coordinate points

(x,y) which bound, in each image of the exam, the region where the specialist found the nodule.

The process of annotating the nodules of the LIDC base was performed by four specialists in two stages. In the first stage, each radiologist analyzed the exams individually. In the second stage, the results of the four analysis of the first stage were presented together to the four radiologists. During this stage, each radiologist reanalyzed the exams and made again their annotations freely.

There is no imposition for consensus; all nodules indicated by the radiologists' revision are taken into account and recorded. So, it is possible to have different diagnosis for the same nodule. In this work, we consider only one instance per nodule, with the objective of minimizing the impact of subjectivity in the exams. The classification regarding malignancy or benignity is obtained first with the calculations presented in [10], which summarizes into one single value the nodular features made by up to four specialists through computing the mode or the median. According to the result of this summary, in this work we consider that malignant nodules are those cases which present malignancy semantic values of moderately suspicious or highly suspicious, and benign nodules are those cases which present characteristics of highly or moderately indicated benignity. As contour, we adopt the one that contains larger bounds. As a total, we obtained 73 nodules (47 benign and 26 malignant).

2.2 Preprocessing with Histogram Equalization

The histogram of a digital image with gray levels in the range $h(r_k) = n_k$, where r_k is the kth gray level and n_k is the number of pixels in the image which have gray level r_k. Histogram manipulation can be used effectively for image enhancement [11].

The equalized histogram using Cumulative Distribution Function (CDF) S_k can be obtained by:

$$S_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n} = \sum_{j=0}^{k} p_r(r_j) \tag{1}$$

where $0 < r_k < 1$ (normalized gray level) and $k = 0,1,2,...,L-1$ (L is the gray level number).

This work uses the technique of histogram equalization to enhance image features present in the nodules, improving the performance to the later stages of the methodology. Figure 1 illustrates this application.

2.3 Lung Nodule Segmentation

For segmenting the nodules, we get information about their contour from the exam database XML file. As previously mentioned, this file contains the coordinates of the nodule based on the analysis criterion of each specialist. The segmentation also

follows the summary presented in section 2.1, where only the larger bound is chosen to represent the instance of the nodules described by up to four specialists.

2.4 Features Extraction Using Diversity Indices

In Ecology, the term diversity is the variety and variability among living organisms and the ecological complexes in which they occur.

A diversity measure is an extremely reductionist parameter, which aims to express all the structural complexity of an ecological community through one single number. In the wide range of available methods for measuring diversity, we may highlight, due to the widespread use, non-parametric indices (or heterogeneity indices), such as Shannon's and Simpson's indices, which we will use in this work.

2.4.1 Shannon's Index

Originated from information theory [12], Shannon index is one of the most widely used indices. Probably its origin and its association with concepts such as entropy contribute to this [13]. The index is based on the principle that the diversity, or information, in a natural system can be measured in a similar way to the information contained in a code or a message. It assumes that individuals are randomly sampled from an infinitely large community, and that all species are represented in the sample [14]. The Shannon index is calculated from the equation 2:

$$H' = -\sum_{i=1}^{S} p_i \ln p_i \tag{2}$$

where p_i is the proportion of individuals found in species i, calculated as $p_i = n_i / N$, where n_i is the number of individuals in species i and N is the total number of individuals in the community. S is the total number of species.

The values obtained through Shannon's index vary between zero, when there is just one species, and the log of *S*, when all species are represented by the same number of individuals [12].

Shannon's Index were calculated considering the voxels of the volume under analysis as the individuals of the population and their intensities as the species.

2.4.2 Simpson's Index

The Simpson's Index is a second order statistical spatial feature that has been used by Ecology specialists to determine the biodiversity of species in a region [15]. Its main functionality is to summarize the representation of this diversity in a single value capable of qualifying this region as either very heterogeneous or uniform.

Simpson's Index takes into consideration the richness of the species, that is, the number of species present in an area, and still, the regularity of such species, what is a measurement of the relative abundance of each species [16] [17]. With these considerations

it is possible to analyze which community in a region is more diversified. The Simpson's Index is the measurement of the probability of two individuals, randomly selected from a sample, to belong to the same species i among the species j existing in the sample, as in Equation 3 [18].

$$D = \sum_{i=1}^{j} p_i^2 \tag{3}$$

where $p_i = \frac{n_i}{N}$. For each i it is found the probability (p_i) for the occurrence of the specie i; n represents the occurrence of individuals from species i and N is the total of individuals in the sample. The index is normally used according to Equation 1 when the sample is obtained by sampling process, not being possible to exactly determine the number of individuals in this sample. For a finite sample, where the total amount of individuals is known, the Simpson's Index can be obtained, still, through Equation 4 [19].

$$D = \frac{\sum_{i=1}^{j} n_i(n_i - 1)}{N(N-1)} \tag{4}$$

The values obtained for the Simpson's Index range from 0 to 1, where the value 0 represents infinite diversity in the sample and 1 means that there's no diversity.

Simpson's Index were calculated analogously to Shannon's Index, considering the voxels of the volume under analysis as the individuals of the population and their intensities as the species.

2.5 Support Vector Machine

Support Vector Machines (SVM), is a method to estimate the function classifying data into two classes [20]. The basic idea of SVM is to construct a hyperplane as the decision surface in such a way that the margin of separation between positive and negative examples is maximized.

The SVM term comes from the fact that the points in the training set which are closer to the decision surface are called support vectors. SVM achieves this surface by the structural risk minimization principle that is based on the fact that the error rate of a learning machine on the test data is bounded by the sum of the training-error rate and a term that depends on the Vapnik-Chervonenkis (VC) dimension [21].

The process starts with a training set of points $x_i \in \Re^n, i = 1,2,...,l$ where each point x_i belongs to one of two classes identified by the label $y_i \in \{-1,1\}$. The goal of maximum margin classification is to separate the two classes by a hyperplane such that the distance to the support vectors is maximized.

The construction can be thought as follows: each point x in the input space is mapped to a point $z = \Phi(x)$ of a higher dimensional space, called the feature space,

where data is linearly separated by a hyperplane. The nature of data determines how the method proceeds. Data can be linearly separable, nonlinearly separable and with impossible separation. The key property in this construction is that we can write our decision function using a kernel function $K(x, y)$ which is given by the function $\Phi(x)$ that maps the input space into the feature space. Such decision surface has the equation:

$$f(x) = \sum_{i=1}^{l} a_i y_i K(x, x_i) + b \tag{5}$$

where $K(x, x_i) = \Phi(x).\Phi(x_i)$ and the coefficients a_i and b are the solutions of a convex quadratic programming problem [22], namely

$$\begin{aligned} &\min_{w,b,\xi} \tfrac{1}{2} w^T \cdot w + C \sum_{i=1}^{l} \xi_i \\ &\text{subject to } y_i \left| w^T \cdot \Phi(\mathrm{x_i}) + \mathrm{b} \right| \geq 1 - \xi_i \\ &\xi_i \geq 0. \end{aligned} \tag{6}$$

where $C > 0$ is a parameter to be chosen by the user, which corresponds to the strength of the penalty errors and ξ_i is a slack variable that penalizes training errors.

2.6 Selection of Features Using *Stepwise* Linear Discriminant Analysis

The selection of variables is a stage intended to reduce the dimensionality of the features space by reducing or eliminating irrelevant attributes. This stage must occur in such a way that the loss of relevant information is the smallest possible and with no negative influence in the classification performance after it ends. For this purpose, we used the stepwise linear discriminant analysis (LDA).

LDA looks for linear combinations of the input variables which best determine the separation of the supplied classes [23].

Instead of looking for a particular form of distribution, LDA uses an empirical approach to define the linear decision planes in the attributes space. The discriminant functions used in LDA are constructed from linear combinations of the variables, in such a way that the distinction between classes is maximized.

$$y = \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n = \beta' x \tag{7}$$

The problem is then reduced to finding an adequate vector β. The basic idea behind discriminant analysis is to determine how different the classes are with respect to the mean of a variable, and then use this variable to make a group suitable for the new sample [24].

Two computational methods can be used to determine a discriminant function: the simultaneous (direct) method and the stepwise method.

The simultaneous estimation involves the computation of the discriminant function in such a way that all independent variables are taken into consideration together. So, the discriminant function is computed with basis on the whole set of independent variables, regardless of the discriminant power of each independent variable [25].

In the stepwise estimation, a variable is selected with basis on its significance and, after each stage, the most significant variables are extracted to form a set of data for investigation. The process is started by choosing the best discriminant variable. The starting variable then pairs one of the other independent variables, one by one, and the variable which is more capable of improving the discriminant power of the function, combined with the first variable, is chosen. After each stage of incorporation of variables, comes the stage where the previously chosen variables may be discarded. The procedure ends when no variable is included or discarded.

In this work, we apply the Linear Discriminant Analysis to determine the variables which best discriminate the nodules with respect to their nature (malignant or benign) using the stepwise method to select the independent variables that best discriminate the classes in order to reduce the dimensionality of the variables for the model. This is done to reduce the processing time in the classification step and trying to eliminate correlated features that may prejudice the training of the SVM.

2.7 Validation of the Classification Method

In order to evaluate the methodology with respect to its power of characterizing the proposed classes, we tried to obtain the sensitivity, specificity and accuracy measurements for all the analysis performed in the study. Sensitivity is given by TP /(TP + FN), specificity is obtained by TN/(TN + FP), and accuracy is given by (TP + TN)/(TP + TN + FP + FN), where TP is true-positive, TN is true-negative, FP is false-positive and FN is false-negative [26] [27]. This way, the malignant lung nodules correctly computed are reported as true positives.

3 Experiments

In order to extract the texture features, we applied two different forms of analysis. In the first one, the features were obtained considering the areas of interest formed by concentric circles as shown in Fig. 2. In the second analysis, the area of interest was represented by circular rings, determined by two concentric circles, as in Fig. 3. Our objective with these forms of extraction is to determine differences in the diversity patterns for benign and malignant nodules in each region of study from the nodule bounds to its center.

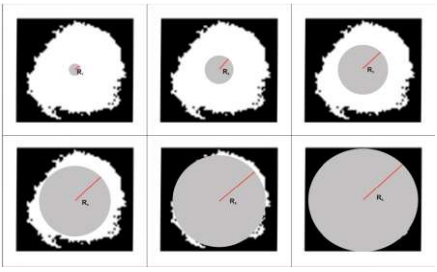


Fig. 2. Analysis applied to the nodule by means of concentric circles containing 6 radiuses

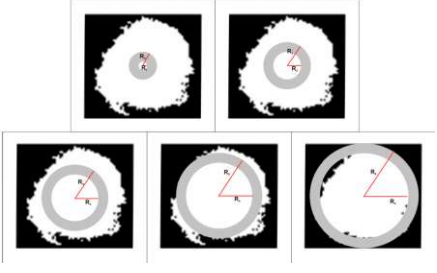


Fig. 3. Analysis applied to the nodule by means of circular rings containing 6 external radiuses

We determined the size of the circles by finding the central point, center of mass, of each nodule and then computing the distance of this central point to the most distant point of each nodule. This way, we get a radius R that represents a greater possible measurement for the construction of a circle which circumscribes the nodule or still, in the analysis by rings, the maximum allowed outer radius. From the radius R, we got the others values of radiuses as 1/6R, 1/3R, 1/2R, 2/3R and 5/6R. These are represented as r_1, r_2, r_3, r_4, r_5 e r_6 (value of R).

For the classification stage with the SVM classifier, we used the LIBSVM library. During this stage, we used a proportion percentage between training and test of 50/50 (Tr/Te) and a total of 73 nodules from the LIDC base, where 47 are benign and 26 are malignant. The cases used in each subgroup were randomly selected from the total.

The proposed methodology also intends to use different tonality scales in order to obtain a better description of texture. For this analysis, the sample is quantized in 8,12 and 16 bits. Each level received different labels, generating three quantization levels presented in the results as Q8, Q12 and Q16. The objective of this approach is to enhance the texture features present in different quantization levels, reinforcing the capacity of the feature extractors used.

4 Results

Table 1 shows the mean results for sensitivity (Se), specificity (Sp) and accuracy (A) in five classifications after the stage of selection of variables in the approaches by rings or by spheres in the three quantization levels described (Q8, Q12 and Q16)

using the two Diversity Indices (DIs): Shannon (Sh) or Simpson (Si). The best result of the experiment is in this table, with sensitivity of 85.64%, specificity of 97.89% and accuracy of 92.78%. The values in Table 2 are also the mean of the classification results, but for these results, all the sixty six variables generated by the experiment were submitted to the selection stage. Table 3 shows the current state of the art.

Table 1. Results using the selected features in each quantization, approach and diversity index

Q8							
Rings				**Spheres**			
DI	**Se (%)**	**Sp (%)**	**A (%)**	**DI**	**Se (%)**	**Sp (%)**	**A (%)**
Sh	54.94	87.65	76.67	Sh	69.94	79.10	74.44
Si	38.14	71.96	58.33	Si	18.53	92.35	65.56
Q12							
Rings				**Spheres**			
DI	**Se (%)**	**Sp (%)**	**A (%)**	**DI**	**Se (%)**	**Sp (%)**	**A (%)**
Sh	61.71	91.06	80.00	Sh	84.54	92.31	89.44
Si	81.90	73.39	76.11	Si	53.15	83.78	71.11
Q16							
Rings				**Spheres**			
DI	**Se (%)**	**Sp (%)**	**A (%)**	**DI**	**Se (%)**	**Sp (%)**	**A (%)**
Sh	81.64	99.17	93.33	Sh	85.64	97.89	92.78
Si	66.03	74.36	71.11	Si	62.85	78.48	72.78

Table 2. Result using the selected features from all quantizations, approaches and diversity indices

Q8, Q12 and Q16			
Rings and Spheres			
DIs	**Se (%)**	**Sp (%)**	**A (%)**
Sh and Si	84.64	94.36	90.56

Table 3. Comparison of related works for lung nodules classification

Methodology	**Database**	**Se(%)**	**Sp(%)**	**A(%)**
(SILVA, 2009) [3]	PRIVATED	100,00	100,00	100,00
(EL-BAZ, NITZKEN, *et al.*, 2010) [6]	PRIVATED	92,15	94,82	94,49
(EL-BAZ, GIMEL'FARB, *et al.*, 2011) [7]	PRIVATED	92,00	100,00	96,36
PROPOSED METHODOLOGY	LIDC	85,64	97,89	92,78

After forming some subsets with the variables generated by both diversity indices and repeating the selection stage in each approach (rings and spheres) and in each different quantization, the outputs were the same obtained with Shannon's index only, which denotes the non-linear distribution of these combinations of features. In the test where all variables were inserted as input for the process of selection of variables,

only three were selected, contemplating the quantizations Q8, Q16 and both diversity indices in the analysis in spheres with radius r_1. The values obtained from this test (Table 2) were close to those obtained by Shannon's Index in the regions of rings and spheres in Q16, also highlighting the similar stability during the five classifications with generated the mean presented in the table. This shows that there can be important features in differents quantizations and in Simpson's index, despite the individual use did not present a high discriminant power.

During the whole experiment, the most frequent approaches were a_1 and r_1, is known by studying the morphology of nodules that these regions (which correspond to the inner regions of the nodule) present relevant morphological aspects, such as presence of necrosis, calcification and fat, which are important for the classification of the nodules as malignant or benign. During almost the whole experiment, we also observed higher values in the specificity, compared to sensitivity. This may be a result of the unbalance of the database, where most nodules are benign.

Analyzing the works proposed in the literature, we can observe that the proposed methodology achieves results comparable to the best ever published, as shown in Table 3. Although, sometimes, some values are lower than for some measures, show that the experiments performed on the task of classification of pulmonary nodules in malignant and benign appear promising. This fact encourages further study, considering, even for use in conjunction with other existing methodologies.

It is important to stress that, for a fair comparison of the cited methodologies, it would be necessary to use the same images in all of the works. Besides, there should be some standard parameters, such as resolution, bits per voxel, protocol, etc. Another factor which should be common to the works is the used sample, because the methodologies should use the same data for the training and test stages

5 Conclusions

This work presented the Shannon and Simpson Diversity Indices as texture descriptors for lung nodules. The results of using the SVM classifier are promising, achieving sensibility of 85.64%, specificity of 97.89% and accuracy of 92.78%. Although the best results have been obtained with larger number of bits per voxels and with the use of Shannon index in both approaches, the selection of variables from the whole set returned values generated by Simpson's index in images with low number of gray levels, and even if the individual results of this configuration have not been superior in the experiment, they show that the quantization technique and Simpson's index may have important elements to distinguish nodules. All these aspects encourage deeper studies in the use if these diversity indices in the classification of lung nodules through SVM.

According to the results, we intend to develop future works with the use of other methods for selecting variables (such as Principal Component Analysis, Genetic Algorithms) and the use of other databases to check whether the behavior is similar to the present experiment.

Acknowledgements. We would like to acknowledge FAPEMA, CNPQ and CAPES for the financial support to this research.

References

1. INCA: Falando sobre câncer e seus fatores de risco (2011) (in Portuguese), http://www.inca.gov.br/conteudo_view.asp?id=81
2. INCA: Tipos de Câncer: Pulmão (2011) (in Portuguese), http://www2.inca.gov.br/wps/wcm/connect/tiposdecancer/site/home/pulmao
3. da Silva Sousa, J.R.F., Silva, A.C., de Paiva, A.C.: Lung Structure Classification Using 3D Geometric Measurements and SVM. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 783–792. Springer, Heidelberg (2007)
4. Liu, Y., Yang, J., Zhao, D., Liu, J.: Computer aided detection of lung nodules based on voxel analysis utilizing support vector machines (2010)
5. Vanessa de Oliveira, C., Raul Queiroz, F. (Advisor): Aristófanes Corrêa, S.(Advisor): Multicriterion Segmentation for Lung Nodule Detection in Computed Tomography. Rio de Janeiro (2009)
6. El-Baz, A., Nitzken, M., Vanbogaert, E., Gimel'Farb, G., Falk, R., El-Ghar, A.M.: A novel shape-based diagnostic approach for early diagnosis of lung nodules. In: Proc. 2011 IEEE Int. Symp. on Biomedical Imaging: From Nano to Macro (ISBI 2011), March 30-April 2, pp. 137–140. IEEE Press, Chicago (2011)
7. El-Baz, A., Gimel'Farb, G., Falk, R., El-Ghar, M.: Appearance analysis for diagnosing malignant lung nodules. In: Proc. 2010 IEEE Int. Symp. on Biomedical Imaging: From Nano to Macro, April 14-17, pp. 193–196. IEEE Press, Rotterdam (2010)
8. Silva, C.A.D., Silva, A.C., Netto, S.M.B., Paiva, A.C.D., Junior, G.B., Nunes, R.A.: Lung Nodules Classification in CT Images Using Simpson's Index, Geometrical Measures and One-Class SVM. In: Perner, P. (ed.) MLDM 2009. LNCS, vol. 5632, pp. 810–822. Springer, Heidelberg (2009)
9. Mcnitt-Gray, M.F., Armato 3rd, S.G., Mclennan, G., Meyer, C.R., Yankelevitz, D., Aberle, D.R., Hensche, C.I., Hoffman, E.A., Kazerooni, E.A., Macmahon, H., Reeves, A.P., Croft, B.Y., Clarke, L.P.: Lung image database consortium research group. Lung Image Database Consortium: Developing a Resource for the Medical Imaging Research Community. Radiology 232(3), 739–748 (2004)
10. Jabon, S.A., Raicu, D.S., Furst, J.D.: Content-based versus semantic-based similarity retrieval: a LIDC case study. In: SPIE Medical Imaging Conference, Lake Buena Vista, FL (2009)
11. Gonzales, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Prentice Hall (2002)
12. Shannon, C.E., Weaver, W.: The mathematical theory of communication. University of Illinois Press, Urbana (1949)
13. Magurran, A.E.: Ecological diversity and its measurements. Princeton University Press, Princeton (1988)
14. Pielou, E.C.: Ecological diversity. Wiley Interscience, New York (1975)
15. Simpson, E.H.: Measurement of diversity. Nature 163, 688 (1949)
16. Hill, M.O.: Diversity and evenness: a unifying notation and its consequences. Ecology 54, 427–432 (1973)

17. Ahumada, L.J., Ayres, D.: Curva de ranqueamento de espíecies, índice de diversidade - simpson (2007), http://wiki.teamnetwork.org/wiki/pages/viewpage.action?pageId=9638
18. Ricklefs, R.E.: 22. In: Estrutura da Comunidade, 3rd edn., pp. 344–346. Guanabara Koogan, Rio de Janeiro (1997)
19. Lyons, D.J., Dunworth, P.M.: Tilbury, D.W.: Simpson's diversity index (2008), http://www.countrysideinfo.co.uk/simpsons.htm
20. Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. Kluwer Academic Publishers (1998)
21. Zhuang, L., Dai, H.: Parameter Optimization of Kernel-based One-class Classifier on Imbalance Learning. Journal of Computers 1 (2006)
22. Scholkopf, B., Smola, A.: Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press (2002)
23. Haykin, S.: Redes Neurais: Princípios e Príatica, 2nd edn. Bookman, Porto Alegre (2001)
24. Hair Jr., J.F., Anderson, R.E., Tathan, R.L., Black, W.C.: Análise Multivariada de Dados. Bookman (2005)
25. Lachenbruch, O.A., Goldstein, M.: Discriminant Analysis. Biometrics 35(1), 69–85 (1979)
26. da Silva Braga, A.C.: Curvas ROC: Aspectos funcionais e aplicações. PhD thesis, Universidade do Minho (2000)
27. Metz, C.E.: Receiver operating characteristic analysis: a tool for the quantitative evaluation of observer performance and imaging systems. American College of Radiology 3, 413–422 (2006)

Comparative Analysis of Feature Selection Methods for Blood Cell Recognition in Leukemia

Tomasz Staroszczyk[1], Stanislaw Osowski[1,2], and Tomasz Markiewicz[1,3]

[1] Warsaw University of Technology, Warsaw, Poland
[2] Military University of Technology, Warsaw, Poland
[3] Military Institute of Medicine, Warsaw, Poland
sto@iem.pw.edu.pl

Abstract. This study analyses different methods of diagnostic feature selection in the problem of classification of the blood cells in leukemia. The analyzed methods belong to the wrapper and filter methods and cover wide range of approaches to feature selection problem. In particular they cover 7 methods, each of them working on different principle. As a results of this preprocessing stage we define the best (according to the applied method) set of features which is next used as the input for the Gaussian kernel SVM classifier. The last step of blood cell recognition is the integration of the results of application of all methods. The numerical results of experiments will be presented and analyzed.

Keywords: diagnostic features, selection methods, classifier, ensemble of classifiers.

1 Introduction

The problem of optimal selection of the feature set used in classification problems belongs to the most crucial steps in the classification process. In any classification problem after the automatic feature extraction we are in disposition of many features of different classification power. Some of them may be meaningless or redundant and present no recognition ability. In the process of feature selection we tend to delete such features to obtain higher generalization performance of the classifier and also to accelerate the classification process.

In practice two approaches to feature selection are applied. One of them is the wrapper and the second filter method [5], [14]. In the wrapper approach the process of feature selection is associated with the classification of data and checking the actually acquired generalization ability of the trained classifier. Usually in this selection process we operate with the set of many features at the same time. In filter approach we apply the selection methods which are not associated with any classification tool. We simply investigate the measure of contribution of each feature for the characterization of the problem.

The selection methods represent the local optimization techniques and the global optimality of resulting feature set is not guaranteed. Moreover each selection method

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 467–481, 2012.

usually results in different set of features and application of them as the input information for the classifier may lead to different results of classification. In typical approach we choose the set which guarantees the best results on the validation data. However better results of classification may be obtained if we integrate the results of application of each set of features into the ensemble of classifiers responsible for the final decision.

Observe that each classifier generates the classification decision which is independent from each other. All of them may be burdened by some errors, which vary from classifier to classifier and usually are also independent. Combining all individual results together allows to compensate some errors and in this way to reduce their average level.

The paper will present and compare 7 different methods of feature selection. They include the Fisher method, correlation of the feature with a class, application of linear and nonlinear Support Vector Machine (SVM), principal component analysis (PCA), independent component analysis (ICA) as well as application of genetic algorithm. These methods will be compared on the difficult example of recognition of two neighboring blood cell types of the bone marrow. The experiments are performed by using support vector machine of the Gaussian kernel working as the classifier.

In the final step of classification we apply the ensemble of classifiers, each applying different sets of features. It will be shown that integrating the results of individual classifiers into final outcome of classification allows to improve the generalization ability of the classification system and reduce the total error.

2 Problem Statement

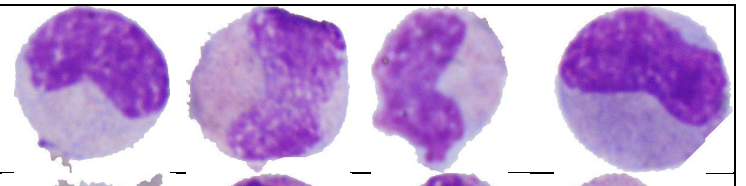
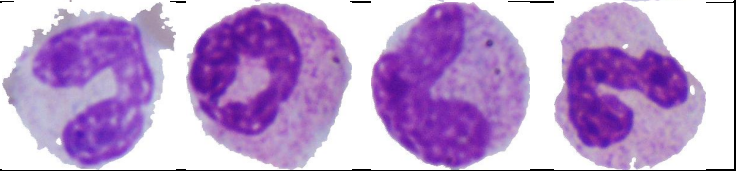
The presentation and comparison of the feature selection techniques will be done on the example of recognition of blood cells of the bone marrow. This is an important problem in leukemia at the recognition of the development stage of the illness and proper treatment of the patients [11]. The specialist recognize different cell lines development in the bone marrow: the erythrocyte, monocyte, lymphocyte, plasma and granulocytic series. A lot of different blast cell types belonging to these lines have been defined up to now by the specialists. They differ by the size, texture, shape, density, color, size of nucleus and cytoplasm [11].

The difficulty of cell recognition follows from the fact that there are large variations among the cells belonging to the same family and the fact that there is a great similarity of the cells belonging to different classes, especially the neighbours. Observe that the transition from one cell type to the neighbouring one is continuous and even expert is in trouble for recognizing the exact moment of transition from one class to the next one.

In automatic recognition of many classes, especially at application of the most efficient SVM classifiers, the most often used approach is one-against-one in which we recognize between only two classes. For each pair of classes we have to select the optimal set of features providing the best generalization ability of the automatic classification system. The most difficult is the recognition of two neighboring cell types,

and all our considerations will be devoted to this problem. From the family of blood cells we have selected two neighboring cell types: the neutrophilic metamyelocyte (class 1) and the neutrophilic band (class 2). They represent the succeeding stages of development of the granulocytic line of the cells. These cell types represent the most difficult classes for recognition, since their shape is very similar and at the same time there is a great variety of the their representatives in one class. Table 1 presents some possible cases of these two cell families.

Table 1. The typical representatives of the cells belonging to neutrophilic metamyelocyte and neutrophilic band

neutrophilic metamyelo-cyte (class 1)	
neutrophilic band (class 2)	

In our investigations we have used 99 metamyelocytes and 146 representatives of neutrophilic bands. The data base of these blood cells has been created in cooperation with Hematology Hospital in Warsaw. The bone marrow smear samples have been collected from more than 40 patients suffering from myelogenous leukemia. The image was digitized using Olympus microscope with the magnification of 1000x and digital camera of resolution 1712x1368 pixels. The picture was saved in RGB format. The smears have been processed by applying standard method of May-Grunwald-Giemsa (MGG).

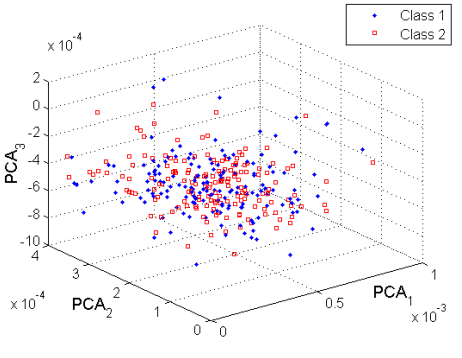


Fig. 1. The 3-dimensional PCA plot presenting the distribution of 87-dimensional data belonging to two neighbouring cells families: neutrophilic metamyelocyte and neutrophilic band

Each cell of the family has been described by the numerical descriptors related to such details as the shape, size of the cells, granulation, texture, distribution of intensity, colour, histogram, gradient of the image, etc. In general they refer to the description of texture of image, geometry of cells, statistical parameters as well as colour histograms determined for the whole cell and nucleus [11]. As a results of this process we have defined 87 descriptors forming the potential diagnostic features to be used by the classifiers. However due to high similarity of the neighbouring cells the numerical descriptors characterizing both families occupy similar positions in the high dimensional space. Good illustration of the difficulty is the principal component analysis (PCA) map of the 87-dimensional vectors of these descriptors to the 3-dimensional space presented in Fig. 1.

It is evident that both classes of cells occupy similar region in a 3-dimenional representation. The representations of both classes interlace each other and are difficult to recognize. We have to look for better numerical representation of the cells, by eliminating the least important descriptors and limiting their number to the most significant values.

3 Feature Selection

The main task of the selection procedure of the numerical descriptors is to choose these features which are best correlated with the class under recognition [2], [5]. Good feature should be stable for samples belonging to the same class (the smallest possible variance) and at the same time it should differ significantly for different classes. The feature assuming similar values for different classes has no discriminative power and may be treated as the noise from the classification point of view. On the other side the individual feature should cooperate well with the other members of the selected set to provide the highest possible generalization ability of the applied classifier. There are many different selection methods belonging in general to wrapper and filter approaches [4-6], [13]. Each of these approaches measures the importance of the feature according to its own strategy, not necessary globally optimal. Therefore their application to feature selection brings usually different results. In our work we apply and then combine together 7 methods. They include the Fisher method, correlation of the feature with a class, application of linear Support Vector Machine (SVM), nonlinear kernel, principal component analysis (PCA), independent component analysis (ICA) as well as application of the genetic algorithm. All of them have been implemented on the Matlab platform [9].

To assess properly the discrimination ability of the selected set of features we have used them as the input signals to the SVM classifier at application of the cross validation approach. In this approach we split the data into 10 approximately equal groups covering the same proportion of both classes. Nine groups are used in learning and the last one is left for testing the learned classifier. The procedure is repeated 10 times, every time at different choice of testing group of data. As the measure of accuracy of the classifier we treat the average of all these 10 classification trials. To determine the optimal number of features we have repeated the cross validation procedures at

application of different number of features, changing them successively from one up to 87. The minimum value of the cross validation error will indicate the optimal number of features.

3.1 Fisher Method

The basic criterion in this method is relied on the values of variance and means of the data samples belonging to each class. The variance of the features corresponding to cells belonging to one class should be as small as possible. Moreover the positions of means of feature values for the data of different classes should be separated as much as possible. The feature of the standard deviation value higher than the distance between two neighboring class centres is useless for these two particular classes recognition, since it does not distinguish between them. In Fisher method the assessment of the feature f for distinction between class A and B is done on the basis of the discrimination coefficient $S_{AB}(f)$, defined in the way [5]

$$S_{AB}(f) = \frac{|c_A(f) - c_B(f)|}{\sigma_A(f) + \sigma_B(f)} \tag{1}$$

In this definition c_A and c_B are the mean values of the feature f in the class A and B, respectively. The variables σ_A and σ_B represent the standard deviations determined for both classes. The large value of $S_{AB}(f)$ indicates good potential separation ability of the feature f for these two classes. On the other side small value of it means that this particular feature is not good for the recognition between classes A and B.

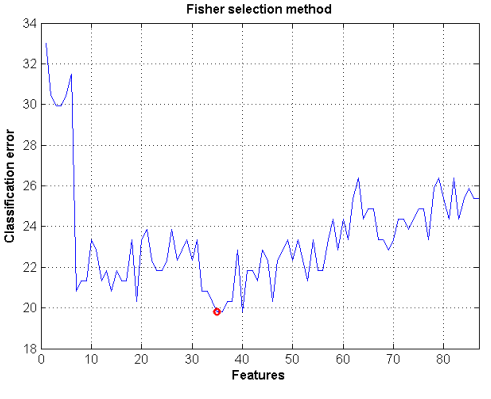


Fig. 2. The distribution of the cross validation errors for the recognition of two classes at application of different number of features according to Fisher method (N_f=36, minE=19.6%)

Application of Fisher method includes few stages. In the first one each feature from the 87 members of potential feature set is analyzed using (1). Then the features are ordered according to their discrimination values (from the highest to the lowest). In the last step we train Gaussian kernel SVM for recognition of classes at application of different number of selected features, starting from one (best) feature and then

adding successively the next ones. In the applied cross validation approach the classification process is repeated 10 times, each time using 9 groups of data for learning and one group for testing.

Fig. 2 shows the results of this cross validation approach in the form of the mean of classification errors at different population of the best features forming the input to the SVM classifier (N_f is the optimum number of features and minE represents the minimum error). We can see quite important diversity of results. Inclusion or exclusion of the particular feature increases or decreases the recognition ability of the classifier. The results of classification depend also on the composition of the data used in experiments, due to their random way of selection. The best result of recognition of two classes has been obtained at selection of 36 best features. The mean error of the recognition of testing set corresponding to this choice of features was equal 19.6%.

3.2 Correlation of the Feature with the Class

The discriminative power of the candidate feature f for the recognition of the particular class can be also measured by the correlation of this feature with the class [13]. Let us assume that the target class k is one between the classes forming target vector of the recognized classes, denoted by d. Let us assume that the feature f is described by its unconditional and conditional means $m_c = E\{f\}$ and $m_{ck} = E\{f|k\}$. Assume that the variance of f is known, $\mathrm{var}(f) = E\{(f - m_c)^2\}$. The correlation between f and **d** is derived from the covariance vector cov(f,**d**), related by the respective variance. The discriminative power of the feature f is measured as the squared magnitude of the correlation vector corr(f,**d**), i.e.,

$$S(f) = \left|\mathrm{corr}(f,\mathbf{d})\right|^2 = \frac{\left|\mathrm{cov}(f,\mathbf{d})\right|^2}{\mathrm{var}(f)\,\mathrm{var}(\mathbf{d})} \tag{2}$$

Denoting by P_k the probability of *k*th class and taking into account that $\mathrm{var}(\mathbf{d}) = \sum_{k=1}^{K} P_k(1-P_k)$, we get $\mathrm{cov}(f,\mathbf{d}) = \left[P_1(m_{c1} - m_c),...,P_K(m_{cK} - m_c)\right]^T$. The discriminative power of the feature f is then given in the form [13]

$$S(f) = \frac{\sum_{k=1}^{K} P_k (m_{ck} - m_c)^2}{\mathrm{var}(f)\sum_{k=1}^{K} P_k(1-P_k)} \tag{3}$$

where K denotes the number of recognized classes (K=2 in our case). The results of numerical experiments show that the importance of the succeeding features selected in this way was similar to the results of Fisher method. The Pearson correlation coefficient between the measures of the discriminative powers of the features calculated by applying Fisher and class correlation methods was equal 0.94. The cross validation experiments at application of different number of the best features have shown that the optimum number of features was this time equal 6. At this number the mean value of the cross correlation error of classification of the testing set was equal 18.27%.

Fig. 3. The distribution of the cross validation errors at different number of features at application of correlation of the feature with a class (N_f=6, minE=18.27%)

3.3 The Feature Selection Based on the Application of the Multiple Input Linear SVM

This is the well known and widely used selection method proposed originally by Guyon and Vapnik [6]. The ranking of the features is done here at application of all features as the input to the linear kernel SVM working as a classifier. The linear kernel SVM is used as the classifier, because this kernel does not deform the original impact of the features on the result of the classification. The decision function of the N-dimensional input vector $\mathbf{x}$ is a linear function defined as $D(\mathbf{x})=\mathbf{w}^T\mathbf{x}+b$ with the weight vector $\mathbf{w}$ and bias b dependent of the linear combination of the training patterns ($\mathbf{x}_k$, d_k) belonging to the support vectors [12].

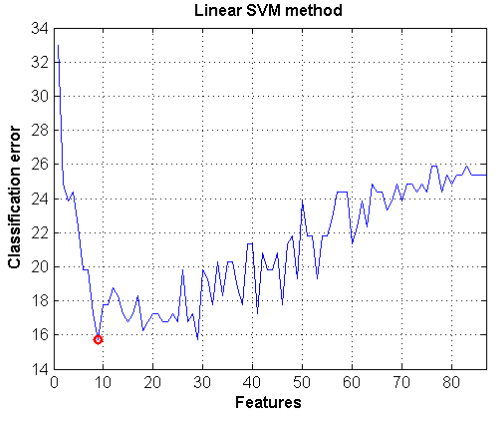


Fig. 4. The distribution of the cross validation errors for the recognition of two classes at application of different number of features according to the linear SVM (N_f=9, minE=15.74%)

The method is based on the idea, that the absolute values of weights of a linear classifier produce a feature ranking. The feature associated with the larger weight is more important than that associated with the small one. All values of weights are arranged in decreasing order and only the most important should be selected. To find the optimal number of the features we have performed many experiments of learning the Gaussian kernel SVM at application of different number of the features chosen successively according to their position after ranking. This was done according to the same cross validation procedure as in the previous experiments.

Fig. 4 presents the results of these cross validation trials in the form of mean values of errors at different populations of feature set. At 9 feature set we observe the significant decrease of the classification error and this number of the best features was assumed as the optimal one. The mean error of the recognition of testing set corresponding to this choice of features was equal 15.74%.

3.4 The Feature Selection Based on the Application of Nonlinear Kernel

In this approach we use the Gaussian kernel $K(\mathbf{x},\mathbf{x}_i)=\exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_i\|}{2\sigma^2}\right)$ treating it as the measure of similarity of vectors $\mathbf{x}$ and $\mathbf{x}_j$, that is $D(\mathbf{x}, \mathbf{x}_j)= K(\mathbf{x}, \mathbf{x}_j)$. Let us assume that class 1 is represented by n_1 and class 2 by n_2 patterns. Then the average similarity of data belonging to one (*k*th) class is described by

$$m_k=\frac{1}{n_k(n_k-1)}\sum_{\substack{i,j=1\\ i\neq j}}^{n_k} D(\mathbf{x}_i^{(k)},\mathbf{x}_j^{(k)}) \tag{4}$$

Good candidate for feature should be characterized by high value of m_k for k=1, 2. At the same time the similarity of the features characterizing the patterns belonging to opposite classes should be as low as possible. The average between class similarity is described by

$$d_{12}=\frac{1}{n_1 n_2}\sum_{i,j=1}^{n_1,n_2} D(\mathbf{x}_i^{(1)},\mathbf{x}_j^{(2)}) \tag{5}$$

On the basis of these measures we can define the measure of separability of two classes by the feature f as follows

$$S_{12}(f)=\frac{m_1+m_2}{d_{12}} \tag{6}$$

High value of $S_{12}(f)$ means good discrimination ability of the feature f . We have implemented this approach in the following way. Starting from full set of features in each iteration of the algorithm we search for the feature which removal maximizes the measure of separability of the remaining set. At any search after removing *i*th feature we define the modified value of separability measure

$$S_{12}^{(-i)}(f) = \frac{m_1^{(-i)} + m_2^{(-i)}}{d_{12}^{(-i)}} \tag{7}$$

The highest value of this measure points to the feature that should be eliminated from the set. The next iterations start from the already obtained reduced set. The process is continued up to the minimal number of features guaranteeing the best performance of the classifier. In each step we have applied the width of the Gaussian function $\sigma = \sqrt{N/2}$, where N is the size of the actual feature set.

To find the optimal number of features we have followed this step by the already presented cross validation approach implemented by Gaussian kernel SVM. The results in the form of the mean classification error at different number of optimally selected feature sets are presented in Fig. 5. The optimum number of features corresponds to the set containing 27 selected features. The mean value of the classification error of the testing set at this point was equal 19.80%.

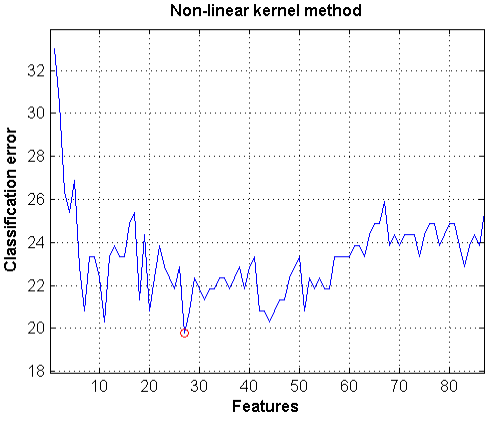


Fig. 5. The distribution of the cross validation errors for the recognition of two classes at application of different number of features selected according to the nonlinear SVM (N_f=27, minE=19.8%)

3.5 Principal Component Analysis for Feature Extraction and Selection

Principal Component Analysis represents a classical statistical technique for analyzing the covariance structure of multivariate statistical observations, enhancing the most important elements of information. It reveals the structure behind the correlation of many variables and is described as the linear transformation **y=Wx**, mapping the N-dimensional original feature vector **x** into K-dimensional output vector **y**, where K<N. The vector **y** preserves the most important elements of original information and **W** is the PCA transformation matrix composed of the eigenvectors of the correlation matrix $\mathbf{R}_{xx}$ associated with the set of feature vectors $\mathbf{x}_i$.

In distinction to the above presented techniques this approach operates not on the original features but on the linear combination of them. Similarly to the other methods there is still a problem of determining the optimal number of principal components

that will be used as the features applied in the classification process. This problem was solved by learning many different SVM classifiers of Gaussian kernel in the cross validation mode. The results of this approach are presented in Fig. 6.

The best results of classification correspond to the application of 19 most important (corresponding to the highest eigenvalues of the covariance $\mathbf{R}_{xx}$ matrix) principal components. The mean classification error of the recognition of testing set at this feature set was equal 24.30%.

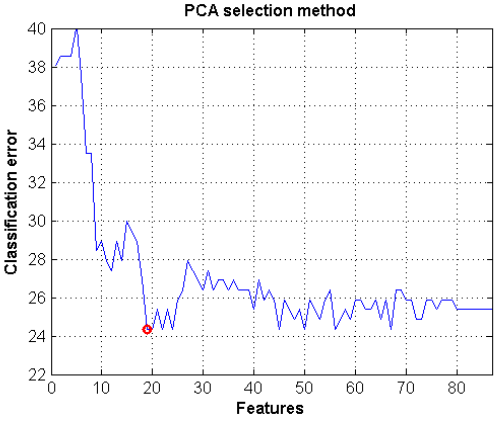


Fig. 6. The distribution of the cross validation errors for the recognition of two classes at application of different number of PCA created features (N_f=19, minE=24.3%)

3.6 Independent Component Analysis for Feature Extraction and Selection

Independent component analysis (ICA) is a computational method for separating a multivariate signals into additive subcomponents assuming the mutual statistical independence of the non-Gaussian source signals [1]. These subcomponents represent the independent source signals extracted from the measured set of signals. As these signals we will treat the numerical descriptors characterizing the succeeding blood cells. At N descriptors and M representatives of them we have N streams of data forming the "time" series of M entries described by the matrix **X**. The ICA decomposes this matrix into another matrix **Y** of the independent rows using linear relation

$$\mathbf{Y} = \mathbf{W}\mathbf{X} \tag{8}$$

where **W** is the ICA $N \times N$ transformation matrix. The mechanism of decomposition is relied on such choice of matrix **W**, which guarantees the vectors forming matrix **Y** to be far from normal as much as possible. Different measures of non-Gaussianity may be applied in practice: kurtosis, negentropy, mutual independence, etc. [1], [4]. We have applied here the kurtosis. In practical implementation of ICA we have exploited the library FastICA [1]. In the first step the whitening procedure by using PCA of data is performed. The additional advantage of this step is ordering the components according to their energetic values measured by the proper eigenvalue of

the covariance matrix. The second step of ICA is performed on this whitened set of vectors, rearranged according to their energetic impact (from the highest to smallest values).

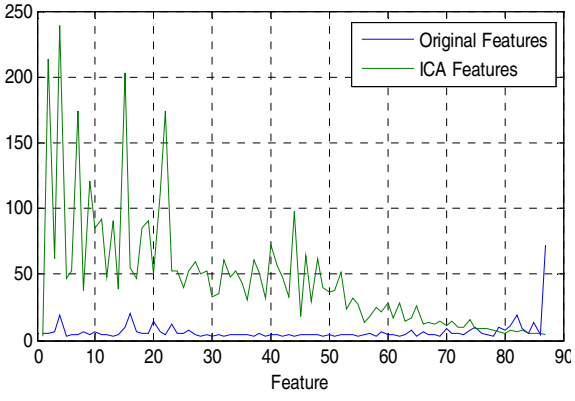


Fig. 7. The distribution of kurtosis of the original features (the solid line) and the features following from ICA transformed data (the dashed line)

Fig. 7 depicts the distribution of the values of kurtosis for the original set of 87 features (solid line) and the kurtosis corresponding to the independent set of the same population (the dotted line). The original set as the mixture of the independent components is close to the normal and the kurtosis is rather low (the average value was equal 6.36). After ICA decomposition the kurtosis corresponding to independent components is very high, especially for the most important components. The average value of kurtosis of the whole data set after ICA transformation was now 46.72 and for the first 24 most important components its value exceeds 100.

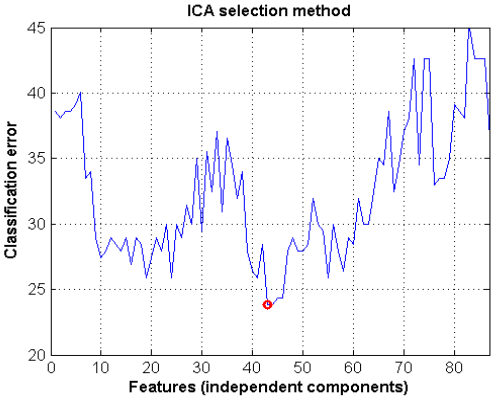


Fig. 8. The distribution of the cross validation errors for the recognition of two classes at application of different number of ICA created features (N_f=43, minE=23.8%)

Fig. 8 presents the results of the cross validation procedure of classification of data using different number of the most important independent components as the features (from one to all of them). This time the lowest value of classification error rate is obtained at application of 43 ICA features. The percentage error of classification of the testing set in this case was equal 23.80%.

3.7 Selection of the Features Using Genetic Algorithm

Genetic algorithm (GA) represents another approach to feature selection. It belongs to the stochastic family inspired by the evolutionary biology such as inheritance, mutation, selection, and crossover and is able to find the global minimum of the optimized function [7], [10]. It consists of selecting parents for reproduction, performing crossover with the parents, and applying the operation of mutation to the bits representing children. The evolution starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness function values), and modified (recombined and possibly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. It was proved that the genetically inspired selection process guides the evolutionary algorithm towards ever-better solutions [7].

In the feature selection problem we have used the binary code representation of the individual feature. The value one in the chromosome means the inclusion of the

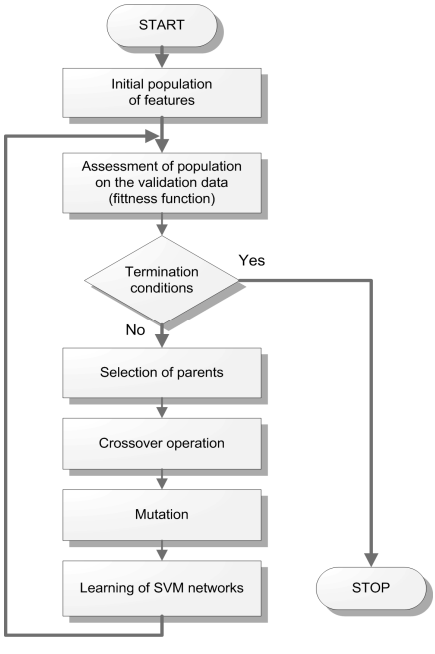


Fig. 9. The diagram of the genetic algorithm used for the feature selection

particular feature, zero – lack of this feature in the vector **x**. In the experiments we have applied an elitist strategy of passing two fittest population members to the next generation. This guarantees that the fitness is never declined from one generation to the next, which is the desirable property in our application.

Each chromosome is associated with the input vector **x** applied to the SVM classifier of the Gaussian kernel (the value 1 means real inclusion of the feature and zero – no such feature in a vector). The classifier is trained on the learning data and then tested on the validation data set. The testing error function on the validation data forms the basis for the definition of the fitness function (the error function taken with minus sign). The genetic algorithm maximizes the value of the fitness function by performing the subsequent operations of selection of parents, the crossover among the parents and finally the mutation. The fitness takes into account the classification error (the difference between the real class denoted by d_k and the actual class y_k indicated by the classifier) and act in the direction of minimizing the number of selected features N_f. To get such effect we have defined the fitness in the following form

$$fitness = -\left[\sum_{k=1}^{p}(y_k - d_k)^2 + \alpha N_f\right] \tag{9}$$

where α is the weighting coefficient adjusted by the user. The roulette wheel has been applied for selection. The general diagram illustrating the genetic algorithm applied for the best feature selection is presented in Fig. 9. The terminating conditions of the genetic algorithm apply the fixed number of generations reached, allocated computation time reached, the highest ranking solution's fitness reached or reached a plateau such that successive iterations no longer produce better results. In application of this algorithm we have applied the crossover probability P_c=0.8, mutation probability P_m=0.01, population equal 20 and the elitist strategy with two most fit individuals.

The algorithm is automatically ended, when in the last 25 generations no improvement of fitness function of the best individual was observed. As a result of such processing we have selected in this way 27 features resulting in 15.62% of average classification error in cross validation mode on the testing data.

As we see each selection method has chosen different numbers of features treated as optimal. We can observe quite significant variety of the selected features. When we compared them together we have found that only 4 of them have been selected by all investigated methods. These features include: mean value of pixels of nucleus gradient of green color, skewness of histogram of gradient of the green color of the whole cell, ratio of the nucleus area to its convex area and the mean distance of pixels of the nucleus to its central pixel. This is due to the fact that each method relied on different principle of operation and none of them is globally optimal.

4 Ensemble of Classifiers and Final Results of Classification

Analysing the results of single selection it is evident that the best results of classification correspond to the application of genetic algorithm. However we should be aware that the results of other methods may carry also significant portion of information that should be not ignored. Therefore we propose to apply the additional

stage of classification in the form of ensemble of classifiers, voting together for elaboration of final decision [8]. The general form of the ensemble system applied here is presented in Fig. 10.

The features selected by using different methods are put to the input of SVM classifiers of Gaussian kernel, responsible for recognition of the class of the cell [12]. Since each classifier undertakes its decision on the basis of different sets of features, their votes may be different. The integrating system considers all partial decisions of SVM classifiers and develops the final classification result.

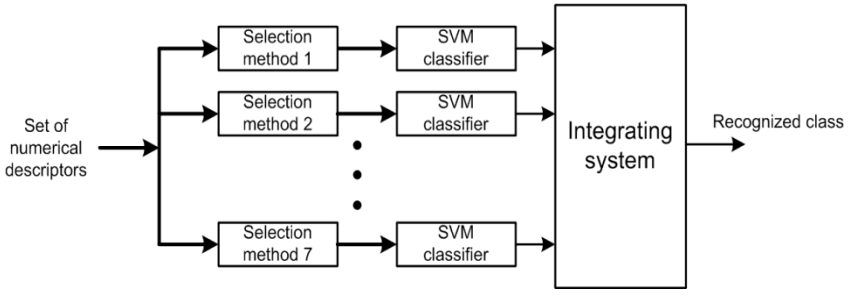


Fig. 10. The general scheme of the ensemble of classifiers

The important point is to choose the appropriate method of integrating these partial results together, providing the highest efficiency of the whole recognizing system. The simplest one is the majority voting [3], [8]. In this method the winning class corresponds to the this one, which was pointed by the highest number of classifiers. More advanced method is the application of an additional SVM classifier as an integrator. In this case we have to teach the additional SVM network to undertake proper decision by using the learning data set. The learning data for this classifier are formed by the decisions of 7 classifiers of the preceding stage, corresponding to the learning set and known destinations associated with them. The SVM integrator is learned on the learning data and tested in cross validation mode. In the same way as for individual classifiers we have split the whole data into 10 parts. Nine of them were used for learning and the last one tor testing. The whole process was repeated 10 times by exchanging the part of data left for testing.

Table 2. The confusion matrix of the classification results of the testing data by an ensemble of classifiers at application of SVM as an integrator

	Ratio of samples recognized as class 1	Ratio of samples recognized as class 2
Class 1	83.90%	16.10%
Class 2	12.46%	87.54%

Table 2 presents the average results in the form of the confusion matrix at application of SVM as an integrator. The diagonal entries of this matrix represent right recognition rate of cells and the off diagonal – the misclassification rate. Each row

presents how the cells of particular type have been classified. The column indicates how many cells have been recognized as the type mentioned in this column. The average misclassification ratio of the system defined as the ratio of all misclassified cases to all cases under recognition was equal 14.28%. The best results of the individual classifier relied on genetic algorithm was improved from 15.62% to 14.28%.

5 Conclusions

The paper has presented and compared different methods of diagnostic feature selection for the recognition of two classes of blood cells in leukemia. The features selected in each method have been applied as the input signals to SVM classifiers. Since each method produced different number of features the results of classification differ a lot. We have combined them into an ensemble and integrated into one final recognition system by using the additional stage of the SVM classifier. As a result of such integration we have got the improvement of the accuracy of final recognition results.

References

1. Cichocki, A., Amari, S.I.: Adaptive blind signal and image processing. Wiley, New York (2003)
2. Duda, R.O., Hart, P.E., Stork, P.: Pattern classification and scene analysis. Wiley, New York (2003)
3. Freund, Y.: Boosting a Weak Learning Algorithm by Majority. Information and Computation 121(2), 256–285 (1995)
4. Genc, H., Cataltepe, Z., Pearson, T.: A New PCA / ICA Based Feature Selection Method. In: IEEE 15th In Signal Processing and Communications Applications, pp. 1–4 (2007)
5. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. J. Mach. Learn. Res. 3(3), 1157–1182 (2003)
6. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using Support Vector Machines. Machine Learning 46, 389–422 (2002)
7. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Amsterdam (1989)
8. Kuncheva, L.: Combining pattern classifiers: methods and algorithms. Wiley, New York (2004)
9. Matlab user manual MathWorks, Natick, USA (2009)
10. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Berlin (1996)
11. Osowski, S., Markiewicz, T.: Support vector machine for recognition of white blood cells in leukemia. In: Camps-Valls, G., Rojo-Alvarez, J.L., Martinez-Ramon, M. (eds.) Kernel Methods in Bioengineering, Signal and Image Processing, pp. 93–123. Idea Group Publishing, London (2007)
12. Schölkopf, B., Smola, A.: Learning with Kernels. MIT Press, Cambridge (2002)
13. Schurmann, J.: Pattern classification, a unified view of statistical and neural approaches. Wiley, New York (1996)
14. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to data mining. Pearson Education, Inc., Boston (2006)

Classification of Breast Tissues in Mammographic Images in Mass and Non-mass Using McIntosh's Diversity Index and SVM

Péterson Moraes de Sousa Carvalho, Anselmo Cardoso de Paiva, and Aristófanes Correa Silva

Federal University of Maranhão – UFMA, Applied Computing Group NCA/UFMA, Av. dos Portugueses, S/N, Campus do Bacanga, Bacanga, 65085-580, São Luís, MA, Brazil
groo.peter@gmail.com, paiva@deinf.ufma.br, ari@dee.ufma.br

Abstract. This paper introduces three approaches, which use McIntosh's Diversity index to extract breast tissue features from mammographic images, for later classification, through Support Vector Machine (SVM), into mass and non-mass. In order to implement the diversity index, it is necessary to define the element that will represent the species in the image. So, in the first approach, the intensities of the pixels of the image are treated as species, and the texture statistic used is the histogram. Considering the spatial relations of direction and distance between pixels, we adopted a second approach, using GLCM as texture statistic, where the species are represented by pairs of pixels, and the third approach, using GLRLM as texture statistic, where the species are represented by gray level run lengths. We achieved an accuracy of 60.25% with the first approach, 99.00% with the second one and 99.75% with the third one.

Keywords: Mammography, McIntosh's Diversity Index, GLCM, GLRLM, Breast Tissues, SVM.

1 Introduction

Breast cancer is a serious public health problem, in both developed and emergent countries. It occurs more often among women, corresponding to 22% of the new cases every year. The mean five-year survival rate throughout the world is about 61%. This kind of cancer is relatively rare before 35 years of age, but its incidence increases rapidly after this age. In 2010, in Brazil, the estimations for breast cancer, also valid for 2011, point to 49,240 new cases (INCA, 2010). Following the guidelines of the American Cancer Society (ACS, 2010) for early detection of breast cancer increases the possibility of diagnosing it in a starting stage, possibly resulting in a successful treatment.

Mammography is presently the best technique for early detection of impalpable lesions of the breasts which have high chances of being a curable cancer. It is a radiography of the breast, usually taken from two views (x-ray images taken from different angles) of each breast. This procedure produces a black-and-white image of the breast

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 482–494, 2012.

tissue, either as a large film or as a digital computer image, which is read and interpreted by a radiologist (ACS, 2010).

Recent evidences prove that mammography offer a significant benefit for women aging around 40 years old. Nevertheless, mammography also has limitations: it may miss some kinds of cancer, and this leads the monitoring of cases which are not cancer, including biopsies, but is still a very effective and valuable tool to reduce the suffering and deaths by breast cancer (ACS, 2010). With the beginning of the use of mammograms, it was noticed a reduction of the mortality rates associated to this pathology.

Computer-based support tools for radiologists, CAD/CADx (Computer Aided Detection/Computer Aided Diagnosis), have been developed in the last years to improve the performance of the analysis of mammograms, through the identification of lesions, classification of regions or objects of interest, becoming, rapidly, a well accepted clinic practice to help radiologists in the interpretation of mammograms (MELLO-THOMS et al. 2007).

The commonest approach for the development of CAD/CADx systems involve the procedures of features extraction, performed either by a computer system or manually by the radiologists (PAPADOPOULOS, 2005). Many techniques employed for the extraction of features use texture attributes, since they approximate the evaluation made by the human vision. In mammographic images, these attributes can supply a description of the breast tissue

We propose, in this study, the use of the McIntosh's diversity index (MCINSOTSH, 1967) for the extraction of features from tissues in mammographic images, taking into consideration the capability of this index to compare the *species diversity* between samples with different sizes. The index is computed through the Euclidean distance, which is the measurement used to dimension the similarity between datasets, making the comprehension easier. Although this index is not much used in the literature, in (STANDDON et al., 1997), we can see that it managed to show differences between *burned* and *intact* mineral soils, which were not detected by Shannon's diversity index, largely used in Ecology. Due to these aspects, we consider this index as a strong candidate for the discrimination of breast tissues as *mass* and *non-mass*. So, in this work, we explore the McIntosh diversity index pooled with image processing and pattern recognition techniques.

This work is organized as follows: related studies are presented in Section 1.1. Section 2 gives a description of the McIntosh diversity index, the Histogram, the Gray Level Co-Occurrence Matrix (GLCM) and Gray Level Run Lengths Matrix (GLRLM). Section 3 presents a detailed description of the methodology and of the evaluation used in this work. Section 4 presents and discusses the achieved results. Finally, Section 5 brings the conclusion.

1.1 Related Work

Many studies have been developed, supplying efficient methodologies to help in the detection and diagnosing of breast cancer. In (MOHANY et al., 2011), it is proposed a system for classification of regions of interest of mammographic images into benign

or malignant. They used a set of 19 features, computed from the GLCM and GLRLM, achieving accuracy of 94.9% and, in an extra analysis, with 12 of the 19 original features, 92.3%. In (JUNIOR et al., 2009), it is introduced a methodology for discrimination of regions extracted from mammograms into mass and non-mass, using Moran's index and Geary's coefficient as texture attribute. They used SVM as classifier, achieving an accuracy of 99.39%, sensitivity of 100% and specificity of 98.94%. In that same study, the classification of masses as benign or malignant achieved an accuracy of 88.31%, sensitivity of 84.78% and specificity of 93.55%. In (SOUSA, 2011), the use of Shannon's diversity index is proposed, in two approaches, as texture measurement for classification, through SVM, of breast tissues into mass and non-mass, with accuracy of 99.88%, sensitivity of 99.94% and specificity of 99.78%. In (MERT et al., 2011), they propose the classification of breast masses into benign and malignant, where 30 features are initially extracted and only two are selected, through the *ICA dimensionality reduction algorithm*, achieving, through the SVM classifier with quadratic kernel, an accuracy of 94.41%. With basis on the above studies, we can see that the methodologies based on texture features and pattern recognition present promising results in the detection of cancer through mammograms.

2 Materials and Methods

This section contains the description of the materials and methods used to describe and discriminate the tissues of the mammographic image samples in the classes *mass* and *non-mass*, through the application of the McIntosh diversity index to the histogram of the image, in a first approach, to the GLCM matrix, in a second approach, and to the GLRLM matrix, in a third approach.

2.1 McIntosh's Diversity Index

The study of diversity is used in Ecology to determine the variety of species present in a community or area. The use of indexes, despite they do not represent the total composition of a community, allows us to dimension the richness, the equality and the diversity of the species in the different environment studied. These indexes are useful to monitor and predict the environmental changes, and were initially developed for Macroecology (KENNEDY & SMITH, 1995). The concept of diversity involves two parameters: *richness*, which corresponds to the number of species, and *relative abundance*, which is the number of individuals that determines the species occurring in a location or sample (PIANKA, 1994). Communities with the same richness may differ in diversity depending on the distribution of individuals among the species (MCINTOSH, 1967). The computation of the diversity index results in a single number. According to (MAHAFEE & KLOEPPER, 1997), the diversity index using a single number to represent a given situation is advantageous, since it makes comparisons easier in experiments, and enables the elucidation of changes occurring in the related communities. In the McIntosh diversity index, a community can be thought of as a point in an S-dimensional hypervolume and the Euclidean distance from the

community to the origin can be used as a diversity measurement (MAGURRAN, 2004). The distance is known as U and is computed as:

$$U = \sqrt{\sum_{i=1}^{s} {n_i}^2} \tag{1}$$

where s is the number of species (*richness*), and n_i is the number of individuals (*relative abundance*) of the species i. The diversity of any sample is formally given by:

$$N - U \tag{2}$$

where $N = \sum_{i=1}^{s} n_i$ represents the total number of individuals in the sample. The value of diversity increases when the size of the sample (N) increases and is useful only when samples with the same size are compared (MCINTOSH, 1967). Another McIntosh diversity index, which does not depend on N, is given by:

$$\frac{N - U}{N - \sqrt{N}} \tag{3}$$

This index has the advantage of expressing the observed diversity as a proportion of the absolute maximum diversity $(N - \sqrt{N})$ in a given N and varies from 0, if there is just one species, to 1, if the diversity is maximum (MCINTOSH, 1967). This index is useful when samples with different sizes are compared.

2.2 Histogram

A histogram (GONZALES & WOODS, 2002) is a first order statistic which represents the frequency of the gray levels of the pixels in the image. The histogram of a digital image, with gray levels in the interval $[0, L-1]$, is defined by a vector, where the value of each cell i, denoted by $H(i)$, represents the *number of pixels* in the image with gray level i.

2.3 GLCM – Gray Level Co-occurrence Matrix

Given a spatial relation among pixels that form a texture, the elements of the *Gray Level Co-Occurrence Matrix* (GLCM) describe the frequency at which occur the transitions of gray levels between pairs of pixels. Causing variations in the spatial relation by means of alterations in the orientation and distance between the coordinates of the

pixels, we can obtain several co-occurrence matrixes, from which we extract the measurements used to analyze the textures (HARALICK, 1973).

More specifically, each cell (i, j) of the co-occurrence matrix works as a counter and stores the frequency, denoted by $P(i, j, d, \theta)$, with which two pixels occur in the image, separated by a distance d, in a direction θ, one with color i and the other with color j. The computation of the element of the co-occurrence matrix, for the directions 0°, 45°, 90° and 135°, is described by four equations (HARALICK, 1973):

$$P(i, j, d, 0) = \#\{((k,l),(m,n)) \mid k-m=0, |l-n|=d, f(k,l)=i, f(m,n)=j\} \quad (4)$$

$$P(i, j, d, 45^o) = \#\{((k,l),(m,n)) \mid k-m=d, l-n=-d, f(k,l)=i, f(m,n)=j\} \quad (5)$$

$$P(i, j, d, 90^o) = \#\{((k,l),(m,n)) \mid |k-m|=d, l-n=0, f(k,l)=i, f(m,n)=j\} \quad (6)$$

$$P(i, j, d, 135^o) = \#\{((k,l),(m,n)) \mid k-m=d, l-n=d, f(k,l)=i, f(m,n)=j\} \quad (7)$$

where "#" denotes the number of pairs $((k,l),(m,n))$ of the set and $f(x, y)$ denotes the *gray level function* in the pixel (x, y).

Figure 1b illustrates the structure of the GLCM, built from the image of Figure 1a. The size of the GLCM is $L \; x \; L$, where L is the maximum number of gray levels that the image may have. In the image (Figure 1a), for example, there is three pairs of pixels with distance 2 and horizontal alignment, where the first pixel has gray level i and the second one has gray level j. So, the input (i, j) of the GLCM registers the frequency $P(i, j, 2, 0^o) = 3$.

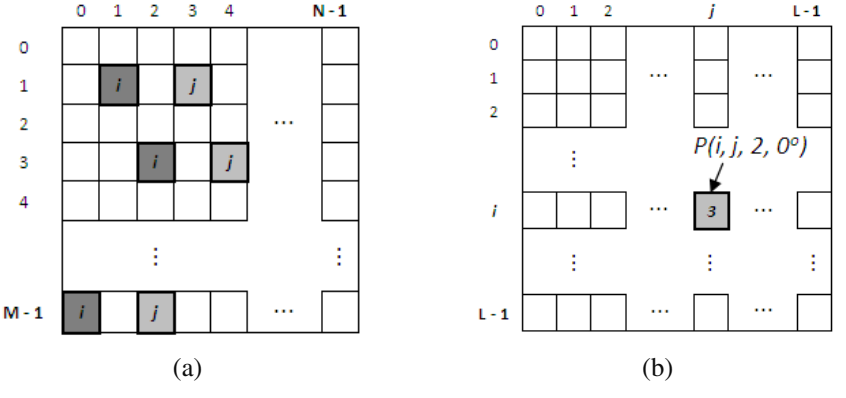


Fig. 1. (a) Image with $M \; x \; N$ pixels. (b) Gray Level Co-occurrence Matrix ($d = 2, \theta = 0^o$)

2.4 GLRLM – Gray Level Run Length Matrix

Given an image, we may define a *gray level run* as a set comprised of consecutive pixels, with the same gray level and collinear in a given direction. The number of pixels in this set denotes the *length* of the run. In order to synthesize the information obtained from these runs, we may compute *Gray Level Run Length Matrixes* (GLRLM), where each element, represented by $P(i, j, \theta)$, contains the number of *runs* with size j (length), having i as *gray level* of its pixel, and the parameter θ as the *orientation* of the line segment formed by the pixels (GALLOWAY, 1975). The computation of the element of the GLRLM (BEBIS et al., 2006) is defined as follows:

$$P(i, j, \theta) = CARD[\{(m, n) \mid f(m, n) = i, \tau(i, \theta) = j\}] \quad (8)$$

where $f(m, n)$ denotes the *gray level function* in the pixel (m, n), and $\tau(i, \theta)$ is the length of the gray level run i and direction θ, e $CARD$ stands for the cardinality of the set (number of elements). The values adopted for θ are 0°, 45°, 90° and 135°. It is necessary to compute the GLRLM for each direction.

Figure 2b illustrates the structure of the GLRLM, built from the image of Figure 2a. The size of the GLRLM is $L \; x \; K$, where L is the maximum number of gray levels that the image may have, and K is the longest gray level run length in the image, with respect to the direction θ. In the image (Figure 2a), for example, there are four gray level runs i, with length 3 and horizontal direction. So, the input $(i, 3)$ of the GLRLM registers the frequency $P(i, 3, 0^o) = 4$.

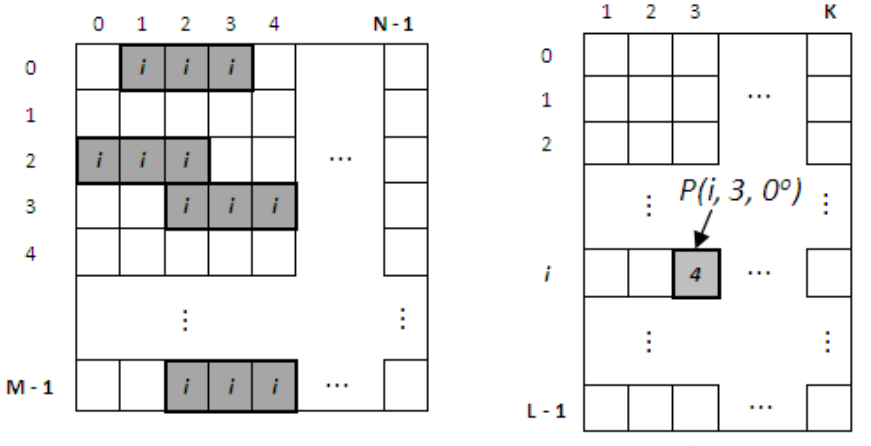


Fig. 2. (a) Image with $M \; x \; N$ pixels. (b) Gray Level Run Length Matrix $(\theta = 0^o)$

3 Proposed Methodology

In this section, we introduce the proposed methodology to discriminate breast tissues, in mammographic images, into the *mass* and *non-mass* classes.

3.1 Database

In this study, we used samples of digitized mammograms from the DDSM - *Digital Database for Screening Mammography* (HEATH et al. 2000), which is freely available over the Web. The images containing suspect areas (masses) are followed by a file containing the description of the lesion (*overlay*), where the *number*, *location*, *type*, *contour* and *diagnosis* of the lesions are informed. We used 800 samples, from which 400 correspond to normal tissues (*non-mass*) and 400 correspond to *masses*. The mass tissue samples were extracted from the contour of the lesions, through the application of a *bounding box* over the contour, despising the pixels between the contour and the *bounding box* and considering in the features extraction stage only the pixels inside the contour, like in (JÚNIOR et al., 2009). The samples of *non-masses* were taken from mammograms without suspect of anomalies. All the samples extracted in this stage had different sizes, since we tried to keep as much information as possible about the texture present in the mass tissues. We opted to use them this way in the next stages, considering the advantage of some diversity indexes, such as McIntosh's, of being relatively independent of the sampling effort. With relatively small sample, we are able to obtain a diversity value that will not change much as we increase the sampling effort, allowing us to directly compare communities studied with different sampling efforts (LLOYD et al., 1968; MAGURRAN, 2004).

3.2 Features Extraction and Classification

For the extraction of features, we performed some experiments, with the samples undergoing a pre-processing by means of the *global histogram equalization* (GONZALES et al., 2002), and other samples without pre-processing. The key function of the pre-processing is the improvement of the image in order to increase the chances of success of the next steps. So, we could observe the possible improvements achieved by the equalization. Later, in each sample, we perform uniform quantizations with 8, 16, 32, 64, 128 and 256 gray levels. So, we intend to aggregate the texture features present in the different quantizations, intending to increase the capability of discriminating tissues.

From each quantization, we compute a McIntosh diversity index (equation 3) to describe the texture of the sample. This computation is proposed through three independent approaches.

In the first approach, the idea is to compute the diversity of gray levels in the image and use it as texture attribute. So, the species are represented by the gray levels. Since the histogram (Section 2.2) registers the frequency of each gray level (species) of the image, we are able to extract the *species richness* (s) from it, represented by the number of non-null inputs (bins) of the histogram, and the *relative abundance* of

each species, represented by the value of each bin. So, the parameters s, N and U, needed for the computation of the McIntosh diversity index (equation 3), are obtained in the following manner:

$$s = \#\{H(i) \mid H(i) \neq 0, 0 \leq i < L\} \tag{9}$$

$$N = \sum_{k=1}^{s} H(i_k) \quad \text{e} \quad U = \sqrt{\sum_{k=1}^{s} (H(i_k))^2} \tag{10}$$

where "#" is the number of elements of the set and $H(i)$ denotes the value of the input of the histogram (frequency of the gray level i). Once we had computed a diversity index for each quantization, the resulting features vector presented 6 variables.

In the second approach, the idea is that if a tissue presents, in general, the texture more homogeneous than another one, it is probable that it has a higher concentration of co-occurrences of homogenous pixels (pairs of pixels with the same gray level), what suggests the diversity index of these pairs of homogeneous pixels as a texture attribute. In this case, the *species* are represented by the *pairs of homogeneous pixels* of gray level i, separated by a distance d, and aligned to a direction θ. So, the GLCM turns out to represent the *distribution of the species* of the region of interest. This way, from the GLCM we obtain the *species richness* (s), represented by the number of non-null inputs of the matrix, and the *relative abundance* of each species, represented by the value contained in each of these non-null inputs. So, considering $P(i, j, d, \theta)$ the value of the input (i, j) of the GLCM, then the values of the parameters s, N and U, necessary for the computation of the McIntosh diversity index (equation 3) are obtained in the following manner:

$$s = \#\{P(i, j, d, \theta) \mid i = j, P(i, j, d, \theta) \neq 0\} \tag{11}$$

$$N = \sum_{k=1}^{s} P(i_k, j_k, d, \theta) \quad \text{e} \quad U = \sqrt{\sum_{k=1}^{s} (P(i_k, j_k, d, \theta))^2} \tag{12}$$

that is, is the number of inputs $P(i, j, d, \theta)$, different from zero of the main diagonal ($i = j$) of the GLCM. If, for example, a GLCM of a sample of tissue, with 256 gray levels, presents the co-occurrence frequencies of homogenous pixels, of gray levels 120, 125, 170 and 182, with distance 2 and direction 0°, like $P(120, 120, 2, 0^o) = 30$, $P(125, 125, 2, 0^o) = 45$, $P(170, 170, 2, 0^o) = 35$ and $P(182, 182, 2, 0^o) = 70$, then the computation of the parameters N and U of the McIntosh diversity index from this GLCM will be:

$$N = \sum_{k=1}^{4} P(i_k, j_k, d, \theta) = 30 + 45 + 35 + 70 = 180$$

$$U = \sqrt{\sum_{k=1}^{4} (P(i_k, j_k, d, \theta))^2} = \sqrt{30^2 + 45^2 + 35^2 + 70^2} = 95.13148$$

So, the value of the McIntosh diversity index is computed as:

$$\frac{N - U}{N - \sqrt{N}} = \frac{180 - 95.13148}{180 - 13.4164} = 0.509465$$

In this approach, the values adopted for the *direction* θ were 0º, 45º, 90º and 135º, and for the *distance* d the values were 1, 2, 3, 4 and 5. This way, since we needed a GLCM for each θ and d, and we considered six quantizations, the resulting features vector had 120 texture attributes (5 distances x 4 directions x 6 quantizations).

In the third approach, we start from the assumption that if a tissue has, in general, many long gray level runs and few short runs (rough texture) and another tissue has many short gray level runs and few long runs (thin texture), then the variety of gray level runs of the first tissue tends to be smaller than that of the second tissue because, in order to fulfill an area, the long runs arrange themselves in a smaller number than the short ones, which causes a smaller distribution of runs. So, assuming that, in general, the mass tissue has thin texture and the non-mass tissue has a rough texture, or vice versa, then we propose the computation of the *diversity of the gray level run lengths* as a texture attribute, in order to enable the discrimination of these tissues. In order to adapt the concept of ecologic diversity, in this case, we adopt the representation of the entity *species* as a gray level run i, length j and direction θ. So, the GLRLM (Section 2.4) turns out to represent the *distribution of the species* in the region of interest. So, we extract from the GLRLM the *species richness* (s), represented by the number of non-null inputs of the matrix (equation 13), and the *relative abundance* of each species, represented by the value contained in each of these non-null inputs. Considering $P(i, j, \theta)$ as the value of the input (i, j) of the GLRLM for the direction θ, then the values of the parameters s, N and U, needed for the computation of the McIntosh diversity index (equation 3), are obtained as follows:

$$s = \#\{P(i, j, \theta) \mid P(i, j, \theta) \neq 0\} \tag{13}$$

$$N = \sum_{k=1}^{s} P(i_k, j_k, \theta) \quad \text{e} \quad U = \sqrt{\sum_{k=1}^{s} (P(i_k, j_k, \theta))^2} \tag{14}$$

Since we need a GLRLM for each direction, we used four of them, making θ equal to 0º, 45º, 90º and 135º. So, for the six quantizations considered, it was generated a features vector with 24 variables.

In all the approaches, each *features vector* was classified by means of *Support Vector Machine* (SVM)(VAPNIK, 1998; NOBLE, 2006), with the RBF (radial base function) kernel, more commonly used in pattern recognition problems.

4 Results and Discussion

In the classification stage, the set of features vectors was divided into two groups: *training* and *test*. Both groups are formed by vectors of mass and non-mass features. For each one of the three approaches, we ran 5 *repetitions* of training and test, through *random selection*. From the total of samples, 50% were used for training and 50% for test. In the training stage, using the RBF kernel, it is necessary to estimate, for each repetition, the value of two parameters, C and γ (table 1), which allow the SVM to optimize the classification model and obtain the best accuracy for each problem. In order to estimate these parameters, from training samples, we used the script *grip.py*, which is part of the package LIBSVM (CHANG et al., 2010). To validate the classification results, we used the measurements of *sensitivity*, *specificity* and *accuracy* (BLAND, 2000).

Table 1 presents the *minimum*, *maximum* and *mean accuracies* found in each approach, for both *equalized* and *non-equalized* samples. In the first approach (histogram), the non-equalized samples presented maximum accuracy of 60.25%, against 59.00% in the equalized samples. The equalization, in this case, brought no improvement. The second approach (GLCM), on the other hand, was considerably influenced

Table 1. Classification accuracy of the three approaches

Approach	Equalized		C	γ	Accuracy (%)
with Histogram	*No*	*Minimum*	*819.0*	*8.0*	*53.50*
		Maximum	*0.03125*	*8.0*	***60.25***
		Mean			*58.25*
	Yes	*Minimum*	*0.125*	*8.0*	*55.25*
		Maximum	*2048.0*	*0.007812*	***59.00***
		Mean			*57.65*
with GLCM	*No*	*Minimum*	*32768.0*	*0.125*	*85.75*
		Maximum	*32768.0*	*0.125*	***90.00***
		Mean			*88.05*
	Yes	*Minimum*	*32768.0*	*0.5*	*96.75*
		Maximum	*32768.0*	*0.5*	***99.00***
		Mean			*98.00*
with GLRLM	*No*	*Minimum*	*512.0*	*0.5*	*99.25*
		Maximum	*512.0*	*0.5*	***99.75***
		Mean			*99.55*
	Yes	*Minimum*	*2048.0*	*2.0*	*99.00*
		Maximum	*2048.0*	*0.5*	***99.50***
		Mean			*99.15*

by the equalization, since there was a mean increase of 9.95% in the accuracy. The explanation for this, in a first situation, is that the equalization of an image in 256 gray levels causes, besides the increase in contrast between the intensities of the pixels, a reduction in the number of intensities, which means a reduction in the species richness. In a second situation, the equalization extends the interval of gray levels in the image to [0, 255] in such a way that, in each quantization, there is an increase in the species richness and redistribution of the relative abundances, when compared to the species richness and relative abundances with basis on the respective quantizations of the non-equalized image. It is probable that both situations increase the differences between the respective diversity indexes of some samples of mass and non-mass which, so far, without equalization, would be very close, and so improving the discrimination of these samples. We can see in table 1 that the third approach (GLRLM), with non-equalized samples, presented the best result, with accuracy of 99.75%.

Table 2 presents the best results for each approach. The first approach presented a high number of false positives, which affected negatively the specificity and accuracy. In the approaches with GLCM and GLRLM, the rates of false positives and false negatives were considerably low, leading to good accuracies, above 99% in both. We can notice the high influence of the spatial relations of *distance* and *direction* in the discrimination of tissues, since there was an accuracy improvement, from 60.25%, in the first approach, to something around 99.00%, in the approaches with GLCM and GLRLM.

Table 2. Best performance results for each approach

Approach	TP	FP	TN	FN	Sensitivity	Specificity	Accuracy
with Histogram	146	106	95	53	73.37%	47.26%	60.25%
with GLCM	210	4	186	0	100.00%	97.89%	99.00%
with GLRLM	189	0	210	1	99.47%	100.00%	99.75%

5 Conclusion

In this work, we achieved significant results for the classification of breast tissues as mass and non-mass, adopting concepts of *species richness* and *relative abundance*, used in the study of ecologic diversity, through the McIntosh's diversity index, and texture information statistics, such as GLCM and GLRLM, generated from mammograms. Among the three approaches, we noticed the problem of the inability of the McIntosh diversity index to condensate texture information from the histogram of the sample (first approach), since the maximum accuracy in this approach was of 60.25%. With maximum accuracies of 99.00% for the second approach (with GLCM), and 99.75% for the third one (with GLRLM), we conclude that the spatial relations of *distance* and *direction* between pixels were decisive for the success of the methodology. Due to the generation of fewer variables and the achievement of the best accuracy, the approach with GLRLM is more efficient that the approach with GLCM, and represents a smaller computational cost for the SVM classifier. As future studies, we

intend to extend the approach with GLCM, starting from the computation of the diversity of gray level transitions between pairs of pixels, and employ the new approach, with GLGLM (Gray Level Gap Length Matrix), which, according to (XINLI, 1994), is complementary to the GLRLM and provides more texture information, with may be useful in the extension of the methodology to the classification of masses as *benign* and *malignant*.

Acknowledgements. We acknowledge FAPEMA, CNPQ and CAPES for the financial support.

References

1. A.C.S. (ACS), Breast Cancer (2010), http://www.cancer.org
2. Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Nefian, A., Meenakshisundaram, G., Pascucci, V., Zara, J., Molineros, J., Theisel, H., Malzbender, T.: Advances in Visual Computing, p. 901. Springer (2006)
3. Bland, M.: An introduction to medical statistics, 3rd edn. Oxford University Press, Oxford (2000)
4. Chang, C.C., Lin, C.J.: LIBSVM-A Library for Support Vector Machines (2010), http://www.csie.ntu.edu.tw/~cjlin/libsvm/
5. Galloway, M.M.: Texture Analysis using Gray Level Run Lengths. In: Computer Graphics and Image Processing, vol. 4, pp. 172–179 (1975)
6. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Prentice Hall, Upper Sadde River (2002)
7. Guliato, D., de Oliveira, W.A.A., Traina, C.: A new feature descriptor derived from Hilbert space-filling curve to assist breast cancer classification. In: IEEE 23rd International Symposium on Computer-Based Medical Systems (CBMS), pp. 303–308 (2010)
8. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural Features for Image Classification. IEEE Transactions on Systems, Man, and Cybernetics 3, 610–621 (1973)
9. Heath, M., Bowyer, K., Kopans, D., Moore, R., Kegelmeyer, W.P.: The Digital Database for Screening Mammography. In: Proceedings of the Fifth International Workshop on Digital Mammography. Medical Physics Publishing (2000)
10. INCA. Instituto Nacional do Câncer. Estimativas 2010: Incidência de Câncer no Brasil (2010), http://www1.inca.gov.br/estimativa/2010
11. Junior, G.B., Paiva, A.C., Silva, A.C., Oliveira, A.C.M.: Classification of breast tissues using Moran's index and Geary's coefficient as texture signutes and SVM. Computers in Biology and Medicine 12, 1063–1072 (2009)
12. Kennedy, A.C., Smith, K.L.: Soil microbial diversity and the sustaunability of agricultural soils. Plant and Soil 170, 75–86 (1995)
13. Lloyd, M., Inger, R.F., King, F.W.: On the diversity of reptile and amphibian species in a bornean rain forest. The American Naturalist 102, 497–515 (1968)
14. Magurran, A.E.: Measuring Biological Diversity, p. 248. Blackwell Science Ltd. (2004)
15. Mahafee, W.F., Kloepper, J.W.: Temporal changes in the bacterial communities of soil, rhizosphere, and endorhiza associated with field-grown cucumber (Cucumis sativus L.). Microbial Ecology 34, 210–223 (1997)
16. Mcintosh, R.P.: An Index of Diversity and the Relation of Certain Concepts to Diversity. Ecological Society of America 48, 392–404 (1967)

17. Mello-Thoms, C., et al.: Interactive Computer-Aided Diagnosis of Breast Masses: Computerized Selection of Visually Similar Image Sets From a Reference Library. Academic Radiology 14, 917–927 (2007)
18. Mert, A., Kilic, N., Akan, A.: Breast cancer classification by using support vector machines with reduced dimension. In: Proceedings Elmar, pp. 37–40 (2011)
19. Moayedi, F., Azimifar, Z., Boostani, R., Katebi, S.: Contourlet-based mammography mass classification using the SVM family. Computers in Biology and Medicine 4, 373–383 (2010)
20. Mohanty, A.K., Beberta, S., Lenka, S.K.: Classifying Benign and Malignant Mass using GLCM and GLRLM based Texture Features from Mammogram. International Journal of Engineering Research and Applications 1, 687–693 (2011)
21. Noble, W.S.: What is a Support Vector Machine? Nature Biotechnology 24, 1565–1567 (2006)
22. Nunes, A.P., Silva, A.C., Paiva, A.C.: Detection of masses in mammographic images using geometry, Simpson's Diversity Index and SVM. International Journal of Signal and Imaging Systems Engineering 1, 40–51 (2010)
23. Papadopoulos, A., Fotiadis, D.I., Likas, A.: Characterization of clustered microcalcifications in digitized mammograms using neural networks and support vector machines. Artificial Intelligence in Medicine 34, 141–150 (2005)
24. Pianka, E.R.: Evolutionary Ecology. HarperCollins, New York (1994)
25. Sousa, U.S.: Classificação de Massas na Mama a partir de Imagens Mamográficas usando o Índice de Diversidade de Shannon-Wiener. Tese de Mestrado em Engenharia Elétrica, UFMA, São Luís (2011)
26. Standdon, W.J., Duchesne, L.C., Trevors, J.T.: Microbial Diversity and Community Structure of Postdisturbance Forest Soils as determined by Sole-carbon-source utilization Patterns. Microbial Ecology 34, 125–130 (1997)
27. Vapnik, V.N.: Statistical Learning Theory, p. 736. Wiley, New York
28. Xinli, W., Albregtsen, F., Foyn, B.: Texture Features from Gray Level Gap Length Matrix. In: IAPR Workshop on Machine Vision Applications, pp.375–378 (1994)

A Semi-Automated Approach to Building Text Summarisation Classifiers

Matias Garcia-Constantino[1], Frans Coenen[1], P.-J. Noble[2], Alan Radford[2], and Christian Setzkorn[2]

[1] Department of Computer Science, The University of Liverpool, Liverpool, L69 3BX, UK
[2] School of Veterinary Science, University of Liverpool, Leahurst, Neston, CH64 7TE, UK
{mattgc,coenen,rtnorle,alanrad,c.setzkorn}@liverpool.ac.uk

Abstract. An investigation into the extraction of useful information from the free text element of questionnaires, using a semi-automated summarisation extraction technique to generate text summarisation classifiers, is described. A realisation of the proposed technique, SARSET (Semi-Automated Rule Summarisation Extraction Tool), is presented and evaluated using real questionnaire data. The results of this approach are compared against the results obtained using two alternative techniques to build text summarisation classifiers. The first of these uses standard rule-based classifier generators, and the second is founded on the concept of building classifiers using secondary data. The results demonstrate that the proposed semi-automated approach outperforms the other two approaches considered.

Keywords: Questionnaire Data Mining, Text Summarisation, Text Classification.

1 Introduction

Questionnaires are a useful and common research tool used for collecting information from groups of respondents in many problem domains. Questionnaires are typically comprised of closed-ended and open-ended questions, the answers are stored in tabular and free text formats respectively. While extracting useful information from the tabular part of questionnaires is straightforward (for example using well established data mining or statistical techniques), extracting useful information from the free text part is more challenging because, in most cases, the texts are short, unstructured and contain misspelled words, poor grammar, and abbreviations and acronyms related to a specific domain. A number of approaches aimed at the extraction of useful information from questionnaires have been reported in the literature, either directed at the tabular element of questionnaire data [4], or the free text element [1,7,9,12,21,23] or both [10,11,19,20].

There are a number of ways in which we can attempt to extract useful information from text. For example, we can attempt to use Natural Language Processing

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 495–509, 2012.

(NLP) [14] or Information Retrieval (IR) [3] techniques. However, given the unstructured nature of questionnaire free text data these approaches are unlikely to produce an appropriate result. The approach proposed here is to use classification techniques to extract meaning from free text whereby a sequence of previously generated classifiers are applied to the texts and the resulting class labels interpreted as a summarisation. The advantages of using this approach is that it avoids the use of Natural Language Processing (NLP) techniques which are not readily applicable to unstructured text. More specifically a rule based approach to text classification is promoted as this offers the additional advantage that the reasons for classifications can be easily provided. Note that if desired the tabular data element of the questionnaires can be processed in conjunction with the free text element so as to enhance the final classification result.

This paper presents a semi-automated classification technique called SARSET (Semi-Automated Rule Summarisation Extraction Tool) which aims to support document summarisation classification. The motivation for SARSET is as follows. Previous work conducted by the authors experimented with the use of standard classification techniques. Using these techniques it was discovered that it was not possible to build effective text summarization classifiers. The main reason for this was the inadequacy of the training data. There were two reasons for this: (i) the free text element typically consisted of very few words (less than 20) and (ii) the small number of examples associated with each of the large number of class labels that would typically be required. The research team therefore investigated the use of secondary data (and proposed the CGUSD system reported in [7]). The theory here was that given a sufficiently substantial repository of secondary data it would be possible to build classifiers that could then be applied to the questionnaire data. Although this was an interesting idea, the results were not as good as expected. This lead the authors to conclude that the only way that the desired classifiers could be built was via the intervention of domain experts. Hence SARSET.

SARSET allows domain experts (users) to select phrases from questionnaire returns in a training set that may be appropriate for inclusion in the antecedent of classification rules. SARSET then automatically generates variations of the suggested phrases, using a synonym database and "wild card" characters, and produces a rule set based on this collection of phrases. SARSET then identifies and displays examples from the training sets that are "covered" by the rules. The user can then select appropriate rules to be included in the final classifier and specify exceptions associated with particular rules. Exceptions are phrases (that includes one or more wild cards) that may be covered by a rule antecedent but which should not be used to classify a particular free text example; the concept of exceptions will be made clearer later in the text.

SARSET was evaluated using a corpus of questionnaire returns from veterinary practices (where each questionnaire was concerned with a single consultation) collected as part of the SAVSNET (Small Animal Veterinary Surveillance Network) project [18]. The results obtained using SARSET were compared with

the results obtained using standard rule based classification techniques and the previously proposed classification using secondary data technique.

The rest of this paper is organised as follows. A short review of related work is presented in Section 2, and a formal definition of the problem domain is presented in Section 3. SARSET is then described in detail in Section 4. A brief overview of the SAVSNET project is given in Section 5 together with a description of the SAVSNET veterinary questionnaire corpus. Section 6 presents a comprehensive evaluation of SARSET. A summary of the main findings and some discussion and conclusions are presented in Section 7.

2 Related Work

A number of approaches aimed at the extraction of useful information from questionnaires have been proposed in the literature. Some of this work focuses on the tabular element of questionnaire data [4], some on the free text part [1,7,9,12,21,23] and some on both [10,11,19,20]. It is the work on the free text element that is of interest with respect to this paper.

In recent years the amount of research related to questionnaire data mining has been growing under the influence of the following factors:

1. The accessibility and use of computers and the internet; which, in turn, has facilitated the use of on-line questionnaires in order to automatically collect opinions or commentaries concerning particular topics.
2. The desire of public and private institutions to speed up the process of gathering and analysing information from people (e.g. opinion about politicians, satisfaction with certain products, prevalence of medical conditions in a specific population group, etc.) through the digitalisation and automation of questionnaires and surveys to improve decision making (facilitated by 1).

An interesting trend in questionnaire data mining research is that most of the techniques proposed have been developed in Japan. A possible reason is the popularity of the Kansei Engineering method [17] in Japan. This method aims to design and produce products based on the feelings and impressions of consumers. Feedback from consumers is obtained through questionnaires and surveys, either paper based or electronically based; to which statistical, and more recently data mining techniques, have been applied to extract useful information.

A number of approaches aimed at the extraction of useful information from the free text element of questionnaires have been proposed in the literature. Some examples can be found in [1,9,21,23]. In [23], two statistical learning techniques (rule analysis and correspondence analysis) were combined and used on a balanced set of questionnaires. Here the free text was split into phrases and words so as to extract characteristics regarding individual analysis targets (objects from questionnaires, e.g. cars) and relationships between the characteristics of the targets. The structure of the free text answers within the questionnaires in [23] does not seem to be an issue even though the answers are short (one sentence); however it should be noted that the answers are in Japanese, a language

whose word representation and text structure is different to that of English, thus requiring a different analysis style.

In [21], based on co-occurrence analysis, a semi-automated system for extracting keywords from the free text element of questionnaires and visualising the relationship among sentences is described. A text mining technique called Hierarchical Keyword Graph (HK Graph) was used to extract the keywords and to represent them in a hierarchical structure. In the HK Graph technique, the free text was first divided into words and the keywords selected by the user, then the co-occurrences between the selected keywords and other words in the text was calculated and the words with the highest co-occurrence values extracted and represented as a hierarchical graph structure. A set of statistical techniques, known as Multi Dimensional Scaling (MSD), was used to interactively cluster the respondents in a visual space based on the similarity between extracted keywords. Finally, the HK Graph was used to visualise the relationship among the extracted keywords from each cluster. In [1] a domain expert's interpretation of free text was compared to the automated performance of the keyword co-occurrence text mining algorithm included in the Wordstat software.

Hiramatsu *et al.* [9] presented a system to support the analysis of open-ended questions by extracting only atypical or unexpected opinions present in the answers. The system classified opinions as typical or atypical. Three methods to extract atypical opinions were presented: (i) based on the ratio of typical word combinations in the sentences of an answer (the basic method), (ii) based on the keyword distance obtained after identifying keywords in the opinions and comparing them with words contained in a typical word database and (iii) based on the use of delimiters to split sentences in the opinions into phrases.

From the literature we can also identify approaches that combine questionnaire tabular data and free text, and apply data mining techniques to the combined data. Examples include [11,19,20]. In [11] an automated method was presented based on Probabilistic Latent Semantic Indexing (PLSI) to extract useful information from documents with both fixed (tabular data) and free (free text) formats, such as questionnaires, by representing both the tabular data and the free text as matrices, merging them, weighting their contents and clustering them according to similarity measures. The clusters are then analysed using statistical techniques. The semi-automated approach presented in [19] was aimed at the generation of hypothesis from questionnaire data by applying text clustering to the free text element and classification to the tabular element. While the text clustering process was semi-automated, the evaluation of the clusters generated against the classified tabular data was automated. The method was comprised of four steps: (i) clustering the free text, (ii) identifying interesting clusters, (iii) exploring the content in the clusters and (iv) formulation of hypotheses. In [20] it was suggested that the tabular and free text part of questionnaires can be used as ontology components (e.g. classes, relations and instances). In this context, the ontologies were used as a mechanism for: (i) semi-automatically focusing the mining process, (ii) for assisting in the interpretation of the discovered knowledge and (iii) for exposing the results on the semantic web.

As already noted the authors have previously proposed an automated questionnaire summarisation classification technique called Classifier Generation Using Secondary Data (CGUSD) [7]. The technique was founded on the idea of generating a classifier using an alternative source of free text data and then applying this to the "primary" questionnaire data. The motivation here was similar to that given for this paper; experiments conducted using standard classification techniques had indicated that the desired summarisation could frequently not be produced, because sufficient training data was unavailable with respect to the text summarisation classification task which typically featured a large number of classes and few example records for each. This approach was also applicable with respect to applications where we wish to build and apply a classifier to questionnaire data, but have no labelled data with which to "train" the classifier. The results obtained, although not as good as expected, indicated that CGUSD, despite requiring further refinement, did present a potential solution in cases where no training data is available to train a classifier. The results also prompted the view, promoted in this paper, that manual intervention by domain experts is essential if we wish to extract meaning from the free text element of questionnaires.

3 Problem Formalisation

SARSET is directed at the summarisation of questionnaire returns. The input is a collection of n questionnaires, $Q = \{q_1, q_2, \ldots, q_n\}$, where each questionnaire comprises a tabular component and a free text component, $q_i = \{T_i, S_i\}$ (where i is a numeric questionnaire identifier). The tabular component, in turn, comprises a subset of a global set of m attribute-value pairs, $A = \{a_1, a_2, \ldots, a_m\}$; thus $T_i \subseteq A$. The text element comprises sequences of words, numbers, punctuation and other printable characters. We indicate the set of tabular components as $T = \{T_1, T_2, \ldots, T_m\}$ and the set of free text components as $S = \{S_1, S_2, \ldots, S_m\}$. The objective is then to summarise the free text element of the questionnaires by searching for patterns in the document set that lead to particular classifications according to a number of classes $C = \{C_1, C_2, \ldots, C_n\}$. Note that we indicate the complete set of labels using the identifier C. Each class in C has a set of class values associated with it, $C_i = \{c_{i_1}, c_{i_2}, \ldots, c_{i_k}\}$ (where k is the number of values). Thus we have a multi-class problem. Given a pattern (phrase) s, that might indicate a class value c_{i_j}, this can be expressed in the form of a classification rule $s \Rightarrow c_{i_j}$. The idea is that we have a separate rule base, comprising rules of the from $s \Rightarrow c_{i_j}$ associated with each class, $R = \{R_1, R_2, \ldots, R_n\}$, and that these rule bases can then be applied to summarise (classify) a questionnaire collection. The overall objective is thus to translate the input $Q = \{q_1, q_2, \ldots, q_n\}$ to a sequence of sets of labels $\{\{c_{1_1}, c_{1_2}, \ldots, c_{1_n}\}, \{c_{2_1}, c_{2_2}, \ldots, c_{2_n}\}, \ldots, \{c_{n_1}, c_{n_2}, \ldots, c_{n_n}\}\}$ such that one set of labels $\{c_{i_1}, c_{i_2}, \ldots, c_{i_n}\}$ is associated with each questionnaire q_i and which serves as a summary for that questionnaire.

4 Classifier Generation Using SARSET (Semi-Automated Rule Summarisation Extraction Tool)

In this section the SARSET methodology is described in more detail. SARSET comprises 5 steps as shown in Figure 1 (each is considered in the following subsections). Broadly the SARSET process can be described as follows:

1. The user identifies a relevant phrase and the system automatically identifies variations of this phrase to give a set of phrases P.
2. The system extracts the subset of questionnaires in D that feature (are "covered" by) the phrases in P .
3. If a suitable phrase p_i can be identified in P (one that serves to identify a class value c_i): (i) generate a classification rule with p_i as the antecedent and c_i as the consequent, and add to R, (ii) if necessary add exceptions to the *exceptions base*, (iii) remove p_i from P. Otherwise go to 5.
4. Repeat 3
5. Exit if a suitably effective classifier has been generated. Otherwise go to 1.

Note that the process requires a training set D comprising the free text elements of a set of questionnaires. The "documents" in D are pre-processed so that numbers and symbols are removed, but keeping phrase delimiters such as commas, semicolons and full stops in order to have a clean but coherent free text from which the domain experts can identify relevant phrases.

4.1 Phrase Identification and Generation of Phrase Variations (Step 1)

The first step in the SARSET process, as indicated above, involves the participation of a user (this might be a domain expert). In the graphical user interface (GUI) of SARSET, the first document $d_i \in D$ is presented to the user, who then identifies a phrase relevant to the application domain which describes the document in terms of some summary class type. For example in the case of the SAVSNET veterinary questionnaires we typically have three main types of classes: *Symptoms*, *Diagnosis* and *Treatments*. The user identified phrase is conceptualised as an ordered sequence of k ($1 \leq k \leq 5$) words that includes at least one keyword as determined by the user, and one or more non-keywords (punctuation is ignored). Phrase variations are then generated for the identified phrase, whereby non-keywords are replaced with "wild card markers" which can be matched to any word using a one-to-one matching. The idea here is that given a likely phrase that may become part of a classification rule, similar phrases sharing the same pattern may also be useful. For the phrase variation construction the synonyms of the keyword(s) selected are also considered in order to broaden the coverage of the phrase pattern and its related phrases. The synonyms used are identified automatically using a Lucene[1] index that contains the synonyms defined in the WordNet[2] database.

[1] http://lucene.apache.org/core/
[2] http://wordnet.princeton.edu/

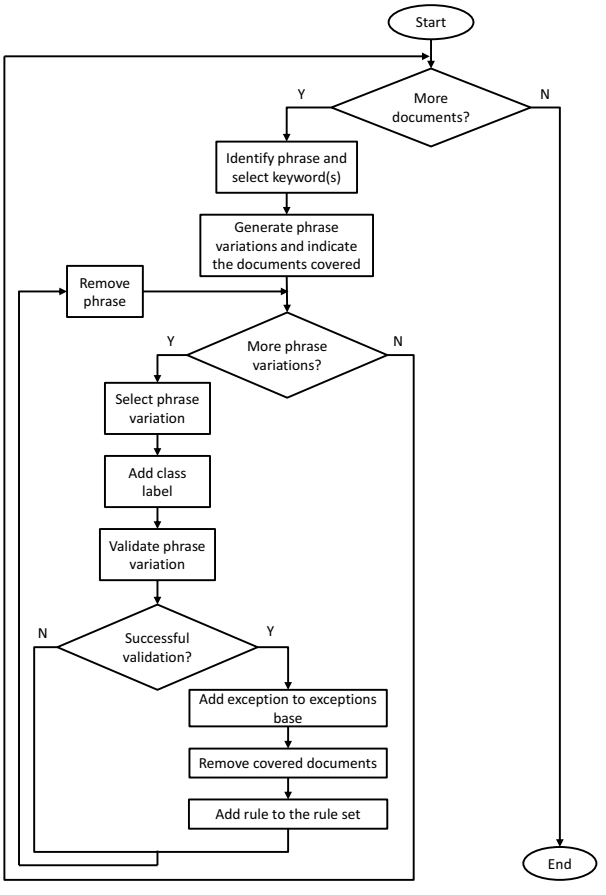


Fig. 1. The SARSET methodology

For example, suppose the phrase "continue with bland diet" has been identified and suppose the set of keywords is $K = \{diet\}$. Consequently the set of non-keywords is $W = \{continue, with, bland\}$. SARSET automatically builds all the variations of this phrase. Including synonyms we get:

$$P = \{\{continue, with, bland, diet\}, \{?, with, bland, diet\}, \{continue, ?, bland, diet\}, \{continue, with, ?, diet\}, \{?, ?, bland, diet\}, \{?, with, ?, diet\}, \{continue, ?, ?, diet\}, \{?, ?, ?, diet\}, \{?, bland, diet\}, \{continue, ?, diet\}, \{?, diet\}, \{continue, with, bland, dieting\}, \{?, with, bland, dieting\}, \{continue, ?, bland, dieting\}, \{continue, with, ?, dieting\}, \{?, ?, bland, dieting\}, \{?, with, ?, dieting\}, \{continue, ?, ?, dieting\}, \{?, ?, ?, dieting\}, \{?, bland, dieting\}, \{continue, ?, dieting\}, \{?, dieting\}\}$$

($|P| = 22$). In the above the first phrase is the phrase identified by the user, the following 10 are variations identified by the system, and the following 11 are the original phrase and its variations using "dieting" as a synonym for "diet".

4.2 Identification of Questionnaires Covered by Identified Phrases (Step 2)

The second step is the automatic retrieval of documents in D which are covered by the identified phrase in the set P. SARSET presents a list of the phrases ordered according to the frequency with which they appear in D, and gives the probabilities with which each phrase is associated with each class. The frequency and the probabilities associated with each phrase allows the user to determine their relevance with respect to the classification task. If desired, the user can inspect the documents covered by each phrase to see the context in which the phrase appears (for example so as to identify potential exceptions).

4.3 Rule Generation (Steps 3 and 4)

In steps 3 and 4 the user attempts to identify suitable phrases to be included in classification rules. This process continues until no more phrases can be identified. For each identified phrase a rule is constructed and added to the rule set R so far. An example of a generated rule is: ? ? *bland diet* $\Rightarrow$ *Diarrhoea*, in which the antecedent of the rule is comprised by a phrase variation and the consequent by the name of a class. The appropriateness of phrases is judged by their associated frequency count and probability values. If, once a rule is generated, the user can identify exceptions these are included in an exceptions base. As already noted exceptions are phrase patterns that should not be covered by particular identified rules. Documents that are covered by generated classification rules are not removed from the document set. The argument for not removing documents in our approach is that the free text element of questionnaires may contain more than one phrase that can be considered relevant. The generated classification rules are ranked according to their sensitivity; high ranked rules are "fired" before other rules. They are also sorted according to the size of their antecedent (number of words) to facilitate efficient matching.

4.4 Continuation of the Process or Exit (Steps 5)

The overall process continues until a suitably effective classifier is arrived at. Due to its semi-automated nature, typically this will be when all documents in the training set are covered by at least one rule in R or no more rules can be generated.

4.5 Applying Classification Rules to Unseen Documents

Once the classifier has been generated it may be applied to summarise unseen questionnaires. In practice several classifiers will be produced to cover different

aspects of the questionnaire set. As already noted with respect to the SAVSNET data three distinct classifiers are anticipated: (i) Symptoms, (ii) Diagnosis and (iii) Treatments. To apply classifiers generated using the SARSET methodology, the document collection to which the classifiers are to be applied must first be pre-processed in a similar way as the document collection that was used to generate the rules, that is, by removing numbers and symbols and by keeping phrase delimiters (commas, semicolons and full stops). A collection of feature vectors, as used with respect to some text classification systems, was not produced because it was not necessary and because it would have been computationally expensive to generate. Phrases of k ($1 \leq k \leq 5$) words size are then identified in the documents and the classification rules applied according to their ranking and antecedent size (i.e. rules whose antecedent comprises 2-words are applied to 2-words phrases). Phrases from an unclassified document that match the antecedent (a phrase pattern) of a classification rule are classified according to the rule that is "fired" first. In the case where a document cannot be classified because there is no phrase pattern that matches any of the phrases in the document a default class is selected (the class that appears most frequently in the training set).

5 The SAVSNET Application

The focus of the work described in this paper is the collection of questionnaire returns obtained as part of the SAVSNET (Small Animal Veterinary Surveillance NETwork) project [18]. SAVSNET is an initiative that is currently in progress within the Small Animal Teaching Hospital at the University of Liverpool in the UK. The objective of SAVSNET is to provide information on the frequency of occurrence of small animal diseases (mainly in dogs and cats). The project is partly supported by Vet Solutions, a software company whose software is used by some 20% of the veterinary practices located across the UK. Some 30 veterinary practices, all of whom use Vet Solutions' software, have "signed up" to the SAVSNET initiative.

The SAVSNET veterinary questionnaires comprise a tabular (tick box) and a free text section. Each questionnaire describes a consultation and is completed by the vet conducting the consultation. The tabular section of the questionnaires includes attributes that are associated with general details concerning the consultation (e.g. date, consultation ID, practice ID), while others are concerned with the "patient" (e.g. species, breed, sex) and its owner (e.g. postcode). The free text section of the questionnaires usually comprises notes made by the vet, which typically describe the symptoms presented, the possible diagnosis and the treatment to be prescribed. It is the free text section that we are interested in summarising, although in some cases the free text element of the questionnaires is left blank.

6 Evaluation

For experimental purposes we used two subsets of the SAVSNET questionnaire corpus, concentrating on symptoms; we refer to these two subsets as SAVSNET-840-4

and SAVSNET-984-3. These subsets were selected in consultation with domain experts who took into account their interestingness (in terms of Veterinary Science) and the amount of data available. SAVSNET-840-4 is comprised of 840 records and 4 class values: (i) Aggression, (ii) Diarrhoea, (iii) Pruritus and (iv) Vomiting (thus $C = \{Aggression, Diarrhoea, Pruritus, Vomiting\}$). SAVSNET-984-3 is comprised of 984 records and 3 classes: (i) Diarrhoea, (ii) Vomiting and (iii) Vomiting and diarrhoea (thus $C = \{Diarrhoea, Vomiting, Vom\&Dia\}$). Some statistical information concerning the two data sets is presented in Tables 1, 2 and 3. Note that both data sets are extremely unbalanced. Note also that the class $Vom\&Dia$ in SAVSNET-984-3 is comprised of records in which diarrhoea and vomiting were presented together, such records are not included under the $Vomiting$ or $Diarrhoea$ classes.

Table 1. Information of the data sets

Data set	Number of classes	Number of records	Min - Max size of documents (in words)
SAVSNET-840-4	4	840	3 - 223
SAVSNET-984-3	3	984	1 - 322

Table 2. Number of records per class in SAVSNET-840-4

	Free Text	
Class	Num.	%
Aggression	34	4.05
Diarrhoea	315	37.50
Pruritus	352	41.90
Vomiting	139	16.55
Total	840	100.00

Table 3. Number of records per class in SAVSNET-984-3

	Free Text	
Class	Num.	%
Diarrhoea	591	60.06
Vomiting	273	27.74
Vom&Dia	120	12.20
Total	984	100.00

The evaluation was conducted by comparing the operation of: (i) standard classifiers generated from the data, (ii) standard classifiers generated from secondary data and applied to the primary data (CGUSD) and (iii) classifiers generated using SARSET. In the first two cases the preprocessing of the free text was similar: stop words (common words that are not significant for the text summarisation process) were removed, stemming was applied using an implementation of the Porter Stemming algorithm [22] and keywords were identified with the well established TF-IDF (Term Frequency - Inverse Document Frequency) measure [13]. TF-IDF weights were calculated for each term and the most significant terms, according to their weight, were selected.

For the Classifier Generation Using Secondary Data (CGUSD) approach, the required secondary data set consisted of medical abstracts obtained from the

MEDLINE database, which comprises around 21 million citations for biomedical literature, including journals and books[3]. The abstracts were extracted using PubMed[4]; PubMed includes many options for searching the MEDLINE database. In our case we used one search query for each of the identified class labels (see Tables 2 and 3) and the "English Language" and "animals" options available in PubMed. In each case r (the maximum number of documents to be retrieved) was set to 500, however for some classes there were less than 500 documents. Thus the final secondary data set for SAVSNET-840-4 comprised 1,974 documents; whilst the SAVSNET-984-3 secondary data set comprised 1,128 documents. A detailed description of CGUSD can be found in [7].

For the two first automated approaches the standard classification technique applied was the TFPC (Total From Partial Classification) algorithm [6], which is a Classification Association Rule Mining (CARM) approach based on the Apriori-TFP (Total From Partial) Association Rule Mining (ARM) algorithm [5]. Apriori-TFP, in turn, is founded on the classic Apriori algorithm [2]. For the evaluation, comparisons were conducted using a range of support threshold (σ) values from 0.5 to 2.5 incremented in steps of 0.5, and a range of confidence threshold (γ) values from 50% to 80% incremented in steps of 10%. The evaluation metrics used were overall accuracy expressed as a percentage and the Area Under the receiver operating Curve (AUC) [8]. The later was deemed to be appropriate because of the unbalanced nature of the input data. For the first approach the reported results were obtained using stratified Ten-fold Cross Validation (TCV); the results are presented in Tables 4 and 5. For the CGUSD approach the accuracy and AUC values were obtained as a result of applying the generated classifier to the entire primary data set; the results are presented in Tables 6 and 7. With reference to Tables 4, 5, 6 and 7, as might be expected, the best results were obtained using a low support threshold coupled with a high confidence threshold. With respect to the results presented in Tables 4 and 6 it should be noted that similar experiments were reported in [7]. However the experiments reported in this paper used a more refined preprocessing of the data than in the case of the previous reported experiments. The results reported in this paper, using the first two approaches applied to the SAVSNET-840-4 data set, are therefore an improvement over the previously reported results.

For the SARSET approach a 50:50 training-test set split was adopted. The classifiers were generated according to the best knowledge of veterinary science possessed by the lead author (who is not a Veterinary Scientist). The classifiers were then evaluated using the test sets. Evaluation was also conducted by applying the classifiers to the training sets and the entire data (training and test set). TCV was not used because of the resource intensive nature of the SARSET approach. The results obtained with SARSET are presented in Table 8.

Inspection of the results presented in Tables 4, 5, 6, 7 and 8 indicate that, with respect to the AUC values, SARSET produces a better performance than the other two techniques. Considering accuracy, SARSET also outperformed the

[3] http://www.nlm.nih.gov/databases/databases_medline.html
[4] http://www.ncbi.nlm.nih.gov/pubmed

Table 4. Standard classification applied to SAVSNET-840-4

σ	$\gamma = 50\%$		$\gamma = 60\%$		$\gamma = 70\%$		$\gamma = 80\%$	
	Acc (%)	AUC	Acc (%)	AUC	Acc (%)	AUC	Acc (%)	AUC
0.5	**76.31**	**0.5466**	**76.67**	**0.5526**	**76.55**	**0.5511**	**76.19**	**0.5672**
1.0	75.00	0.4991	75.36	0.4978	74.17	0.4812	70.83	0.4556
1.5	74.05	0.4933	74.17	0.4909	72.74	0.4683	68.69	0.4353
2.0	70.12	0.4567	72.14	0.4637	67.86	0.4286	71.19	0.4455
2.5	71.19	0.4657	71.55	0.4676	68.69	0.4321	67.14	0.4209

Table 5. Standard classification applied to SAVSNET-984-3

σ	$\gamma = 50\%$		$\gamma = 60\%$		$\gamma = 70\%$		$\gamma = 80\%$	
	Acc (%)	AUC	Acc (%)	AUC	Acc (%)	AUC	Acc (%)	AUC
0.5	**66.53**	**0.4280**	**67.35**	**0.4488**	**66.32**	**0.4456**	**64.47**	**0.4260**
1.0	63.96	0.3935	64.37	0.4067	64.37	0.3924	62.31	0.3706
1.5	63.13	0.3827	63.54	0.3730	62.20	0.3575	62.31	0.3570
2.0	63.44	0.3902	63.44	0.3731	62.82	0.3719	62.51	0.3592
2.5	60.97	0.3685	61.79	0.3535	60.76	0.3477	60.35	0.3333

Table 6. CGUSD applied to SAVSNET-840-4

σ	$\gamma = 50\%$		$\gamma = 60\%$		$\gamma = 70\%$		$\gamma = 80\%$	
	Acc (%)	AUC	Acc (%)	AUC	Acc (%)	AUC	Acc (%)	AUC
0.5	56.55	**0.4929**	56.90	0.5024	53.81	**0.4898**	52.74	**0.4827**
1.0	49.40	0.3966	44.52	0.4544	42.50	0.4439	51.43	0.3983
1.5	59.05	0.4684	40.48	0.4335	**59.76**	0.4665	**59.52**	0.4629
2.0	**60.48**	0.4769	**60.95**	**0.4735**	**59.76**	0.4665	**59.52**	0.4629
2.5	60.36	0.4745	60.24	0.4672	28.45	0.4268	58.93	0.4594

Table 7. CGUSD applied to SAVSNET-984-3

σ	$\gamma = 50\%$		$\gamma = 60\%$		$\gamma = 70\%$		$\gamma = 80\%$	
	Acc (%)	AUC	Acc (%)	AUC	Acc (%)	AUC	Acc (%)	AUC
0.5	**52.73**	**0.4138**	**50.88**	**0.4252**	**47.79**	**0.4181**	**48.20**	**0.4066**
1.0	46.14	0.3628	45.21	0.3624	43.56	0.3621	40.58	0.3746
1.5	42.53	0.3767	41.19	0.3733	39.75	0.3708	37.59	0.3656
2.0	34.09	0.3374	32.13	0.3314	30.38	0.3264	30.07	0.3294
2.5	33.88	0.3417	31.82	0.3317	30.18	0.3273	29.97	0.3309

Table 8. SARSET applied to SAVSNET-840-4 and SAVSNET-984-3

	SAVSNET-840-4		SAVSNET-984-3	
	Acc (%)	AUC	Acc (%)	AUC
Entire data set	78.21	0.7429	66.36	0.7861
First half	**79.57**	**0.7530**	**67.35**	0.7880
Second half	77.32	0.7364	66.39	**0.7900**

other techniques except in the case where standard techniques were applied to the SAVSNET-984-3 data set where results were competitive. However, given the unbalanced nature of the SAVSNET-840-4 data set AUC is a much better measure of the quality of the different techniques. It should also be emphasised again that the classifiers produced using the SARSET methodology were not generated by a domain expert but by the lead author. If a domain expert had been used it is suggested that even better classification results could have been obtained.

7 Conclusion

This paper has reported on SARSET (Semi-Automated Rule Summarisation Extraction Tool), which supports a semi-automated approach to the generation of text summarisation classifiers. SARSET was tested using the free text element of two questionnaire data sets (SAVSNET-840-4 and SAVSNET-984-3). The results of the experiments were compared against the results obtained using two alternative techniques to build text summarisation classifiers, the first using standard rule-based classifier generators and the second (CGUSD) using secondary data related to the domain of the questionnaires. The motivation for a semi-automated classification technique was that most established techniques do not perform well with respect to small unstructured texts, such as those found in the free text element of questionnaires. The generation of rule-based classifiers using SARSET relies to some extent on the knowledge and criteria of domain experts, who are required to search for and identify relevant phrases within the document collection that can be used for the construction of classification rules. The SARSET methodology ends when the domain experts considers the rule base to be sufficiently comprehensive. The results obtained indicate that SARSET is a useful tool for generating rule-based classifiers from unstructured free text. In the context of the AUC measure it outperformed the standard classification and the CGUSD techniques to which it was compared. Although the results were of a satisfactory nature, it is suggested that classifiers generated by experienced domain experts will be even more comprehensive. It is also worth noting that SARSET can be easily adapted and used in other domains, making it a technique with general applicability.

References

1. Abd-Elrahman, A., Andreu, M., Abbott, T.: Using text data mining techniques for understanding free-style question answers in course evaluation forms. Research in Higher Education Journal 9, 11–21 (2010)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Data Bases, pp. 487–499 (1994)
3. Baeza-Yates, R., Ribeiro-Neto, B.: Modern information retrieval, vol. 463. ACM press, New York (1999)

4. Chen, Y.L., Weng, C.H.: Mining fuzzy association rules from questionnaire data. Knowledge-Based Systems 22, 46–56 (2009)
5. Coenen, F.: The LUCS-KDD TFP Association Rule Mining Algorithm. Department of Computer Science, The University of Liverpool, UK (2004), http://www.csc.liv.ac.uk/ frans/KDD/Software/Apriori_TFP/aprioriTFP.html
6. Coenen, F.: The LUCS-KDD TFPC Classification Association Rule Mining Algorithm. Department of Computer Science, The University of Liverpool, UK (2004), http://www.csc.liv.ac.uk/ frans/KDD/Software/Apriori_TFPC/aprioriTFPC.html
7. Garcia-Constantino, M., Coenen, F., Noble, P.-J., Radford, A., Setzkorn, C., Tierney, A.: An investigation concerning the generation of text summarisation classifiers using secondary data. In: Perner, P. (ed.) MLDM 2011. LNCS, vol. 6871, pp. 387–398. Springer, Heidelberg (2011)
8. Hand, D.J., Till, R.J.: A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. Machine Learning 45, 171–186 (2001)
9. Hiramatsu, A., Oiso, H., Tamura, S., Komoda, N.: Support system for analyzing open-ended questionnaires data by culling typical opinions. In: 2004 IEEE International Conference on Systems, Man and Cybernetics, vol. 2, pp. 1377–1382 (2004)
10. Hirasawa, S.: Analyses of Student Questionnaires for Faculty Developments. A Short Course at Tamkang University Taipei, Taiwan, R.O.C., March 7-9 (2006)
11. Hirasawa, S., Chu, W.W.: Knowledge acquisition from documents with both fixed and free formats. In: 2003 IEEE International Conference on Systems, Man and Cybernetics, vol. 5, pp. 4694–4699 (2003)
12. Hiroko, I., Masao, U., Hitoshi, I.: Criterion for judging request intention in response texts of open-ended questionnaires. In: Proceedings of the Second International Workshop on Paraphrasing, pp. 49–56. Association for Computational Linguistics (2003)
13. Jing, L.P., Huang, H.K., Shi, H.B.: Improved feature selection approach TFIDF in text mining. In: Proceedings of the First International Conference on Machine Learning and Cybernetics, pp. 944–946 (2002)
14. Joshi, A.K.: Natural language processing. Science 253, 1242 (1991)
15. McCallum, A.: Information extraction: Distilling structured data from unstructured text. ACM Queue 3, 48–57 (2005)
16. Morinaga, S., Yamanishi, K., Tateishi, K., Fukushima, T.: Mining product reputations on the web. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 341–349 (2002)
17. Nagamachi, M.: Kansei engineering: a new ergonomic consumer-oriented technology for product development. International Journal of Industrial Ergonomics 15, 3–11 (1995)
18. Radford, A., Noble, P.J., Coyne, K.P., Gaskell, R.M., Jones, P.H., Bryan, J.G.E., Setzkorn, C., Tierney, Á., Dawson, S.: Antibacterial prescribing patterns in small animal veterinary practice identified via SAVSNET: the small animal veterinary surveillance network. Veterinary Record 169, 310–318 (2011)
19. Rosell, M., Velupillai, S.: Revealing relations between open and closed answers in questionnaires through text clustering evaluation. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008), pp. 1716–1722 (2008)
20. Svátek, V.: Ontologies, Questionnaires and (Mining) Tabular Data. In: the 3rd European Semantic Web Conference (ESWC 2006) (2006)

21. Uchida, Y., Yoshikawa, T., Furuhashi, T., Hirao, E., Iguchi, H.: Extraction of important keywords in free text of questionnaire data and visualization of relationship among sentences. In: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2009), pp. 1604–1608 (2009)
22. Willett, P.: The Porter stemming algorithm: then and now. Program: Electronic Library and Information Systems 40, 219–223 (2006)
23. Yamanishi, K., Li, H.: Mining open answers in questionnaire data. IEEE Intelligent Systems, pp. 58–63 (2002)

A Pattern Recognition System for Malicious PDF Files Detection

Davide Maiorca, Giorgio Giacinto, and Igino Corona

Department of Electrical and Electronic Engineering (DIEE), University of Cagliari, Piazza d'Armi 09123, Cagliari, Italy
tnemesis@tin.it, {giacinto,igino.corona}@diee.unica.it

Abstract. Malicious PDF files have been used to harm computer security during the past two-three years, and modern antivirus are proving to be not completely effective against this kind of threat. In this paper an innovative technique, which combines a feature extractor module strongly related to the structure of PDF files and an effective classifier, is presented. This system has proven to be more effective than other state-of-the-art research tools for malicious PDF detection, as well as than most of antivirus in commerce. Moreover, its flexibility allows adopting it either as a stand-alone tool or as plug-in to improve the performance of an already installed antivirus.

1 Introduction

The ways in which hackers try to violate the security of computer systems have been constantly evolving. Operating systems have become more secure, as security fixes are constantly released, and the possibility of finding Zero-Day Vulnerabilities[1] is reduced. Therefore, third parties applications (such as Adobe Reader, Microsoft Outlook, etc.) and the file formats they read have become the most targeted ones by the attackers.

The PDF file format is nowadays widely used to read documents, and it is common to think that it is safe. However, its security has been harmed during the past years. The applications that are commonly used to open them have been targeted by cyber-criminals, who have been trying to discover bugs or vulnerabilities that might allow them to gain control of the computer systems used to read a PDF file. Moreover, the spectrum of the attacks is widened by the presence of third-party plugins connected to such applications, which often suffer from bugs that, although discovered, are not patched on time. Once these systems have been exploited, they might also be used by cyber-criminal organizations as part of botnets. This problem has been clearly reported by Symantec and IBM in their 2009 and 2010 security reports [2,3].

Attackers have also become smarter and many countermeasures established by software houses such as Adobe are now bypassed. Most of the attacks focus on bypassing the most advanced protections, so the development of a system

[1] Vulnerabilities discovered and not yet patched.

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 510–524, 2012.

which can be robust against the widest variety of attacks (including possible new ones) will be crucial to address this threat.

In this paper, we present a new tool for the detection of malicious PDF files, where PDF-specific features are employed to build a statistical classifier through machine learning. This paper presents six sections beyond this one. Section 2 provides a basic description of the PDF file format structure. Section 3 presents a basic approach to the most important attacks that harm PDF files and readers. Section 4 is a list of the previous works and tools about PDF security. Section 5 presents the method we have adopted to develop our tool. Section 6 provides the results and the performance of our tool, as well as a comparison with the most important antivirus on the market and with Wepawet [1] and PJScan [17], a powerful tool academically developed. Section 7 closes the paper with the conclusions.

2 An Overview of PDF Technology

2.1 A Brief History

PDF is the acronym of Portable Document format, and it is a widely used standard to exchange documents. Firstly introduced in 1990, it became available for public domain in 1993; in 1994, Adobe Reader, the software used to read PDF files, became free. Nowadays, PDF is one of the most used formats to read and visualize documents, along with DOC (DOCument) and ODT (OpenDocumenT), respectively used by Microsoft Word and Open Office. There is a good reason for this, as the PDF format is flexible, allows for high typesetting quality with relatively small memory usage, and it is recognized among different software platforms and applications.

2.2 PDF Structure

A PDF file is structured as a sequence of dictionary objects (marked by "<<"and ">>"), logically connected to each other: each object can be followed by a compressed stream of data. Each dictionary object contains simpler types of objects (number, array, names), which provide information about the actions performed by the object itself. The stream of data can contain text, images, codes that are processed using the information provided by the objects in the dictionary. Objects can be pages, fonts, images, embedded code. Figure 1 shows an example of this structure, obtained with the PDF CanOpener, an Acrobat plug-in by WindJack Solutions [4].

We will not describe the details of the objects in the picture, as it is not the purpose of this paper. However, it is possible to visualize the same structure in a textual way (RAW mode), in which the information on the type of actions performed by the object are represented by keywords, which are identified by the tag "/".

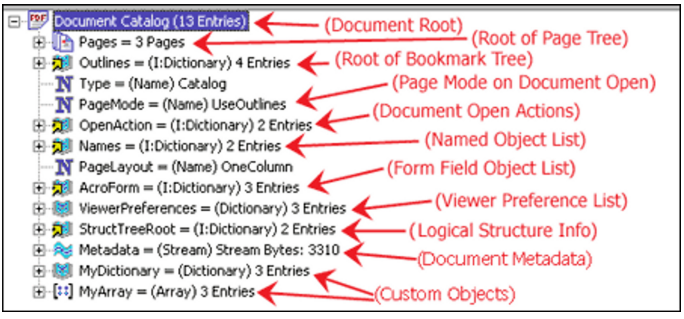


Fig. 1. A graphical representation of a PDF document structure

3 Attacks against PDF Documents

3.1 Attack Types

Due to its flexible object structure, the PDF file format can host Java, Flash code, and a variety of other applications. However, this can bring to a lot of security issues, that may support several types of attacks [5]. In the following we summarize the main security issues.

- Javascript code issue: some malicious Javascript code can be directly injected by the attacker using specific APIs (Application Program Interface) inside a PDF file, in order to exploit a vulnerability of the application that is supposed to read it. An example is reported in the CVE-2009-4324.[2]
- Launch actions: a PDF file may be crafted in order to launch special commands on the operating system, usually after a user confirmation (popup message). A critical example is reported in the CVE-2010-2883.
- Embedded files: a PDF file may contain attached files, which can be extracted and opened by the reader. This may be used to hide malicious executables. See, for instance, the CVE-2010-1240.
- Embedded Flash applications: a PDF file may contain Flash applications (stored as embedded SWF files), as well as malicious ActionScript code. See, for instance, the zero day CVE-2010-3654.

3.2 Evasion Techniques

Attackers often employ obfuscation and encryption techniques in order to bypass IDS and Antivirus. Here is a short description of the most important ones.

[2] CVE stands for Common Vulnerabilities and Exposures and it is a dictionary which classifies the discovered vulnerabilities. It was developed by MITRE, a non-profit organization featuring more than 7000 engineers who work in Applied Systems Engineering and Advanced Technology, and it is considered a vulnerability classification standard.

- GoToEmbedded actions: a PDF file can be embedded inside another PDF file, and a special command can be issued so that Adobe Reader automatically opens the embedded PDF file without notifying the user. This feature may be used to hide a malicious PDF file within a normal PDF file.
- Encryption: a PDF file may be encrypted with a password. However, if an empty password is used, Adobe Reader will open it directly without asking the user.
- Parser "flexibility": it is possible to introduce slight variations on the header of the PDF file in order to bypass too strict antivirus.

3.3 Typical Attack Procedure

Understanding how an attack is typically performed is important to the purposes of our analysis. We will now provide a typical sequence of steps that are adopted when an attack occurs, but we will not go into the details of the exploiting techniques adopted, as it is not the aim of this paper.

1. Opening a malicious PDF file, usually sent by spam. The malicious PDF file could contain SWF, HTML and JS files, Javascript and ActionScript code, and even embedded malicious PDF files. These elements are usually obfuscated in order not to be detected by IDS.
2. Depending on the type code/file embedded, some exploiting techniques are adopted, such as:
   - Buffer Overflow, in which malicious code is inserted in memory areas outside the bounds allowed for the program. [6]
   - Return Oriented Programming (ROP), in which the flow of the program is redirected to a memory area containing the malicious code, using a specific memory address. [7]
   - Heap Spraying (HS), in which the heap area of the memory is filled with multiple copies of an object containing malicious code, in order to bypass some countermeasures against ROP. [8]
   - JIT Spraying (JITS), in which Just In Time (JIT) compilers are used to make writable areas of the memory that should be not writable, and then to inject malicious code into those area. [9]
3. Thanks to the exploit, shellcode embedded inside the PDF file is copied in main memory and executed.
4. The shellcode triggers the download of a "trojan horse"program and automatically executes it, compromising the whole security of the computer system.

4 Related Works on PDF Security

4.1 General PDF Security Research

PDF security is a rather new field of research. Many researchers have been working in order to discover new threats and to produce some tools that can

improve the quality of the PDF analysis. Stevens developed PDFid, a parser that has been used in our experiments [10]: it is able to provide a detailed list of the objects contained in a PDF file, as well as their frequencies (i.e. how many times they appear inside a file). An impressive insight into the vulnerabilities of the PDF files have also been provided by Contagio [11], which also gave us part of the dataset used in our analysis. New tricks and ways of exploiting Adobe vulnerabilities were also described by Cova [12]. Dixon [13] is also helping the scene, providing new tools for malware analysis and a very useful database of malicious files found in the net, along with their characteristics and statistics of the most recurring objects.

4.2 State-of-the-Art Malicious PDF Detection Tools

There are not many academic tools specifically related to PDF security. Most of the available tools detect many types of malware at the same time, and some include PDF as well. We will now provide a brief description of the most important ones.

CWSandbox. This tool has represented an important breakthrough in malware analysis. CWSandbox [14] is a sophisticated platform capable to extract the dynamic behavior of a computer system once a certain (e.g. suspicious) file is opened and executed. Files are executed in a controlled (virtual) environment and a detailed report of the raised operating system events is built. While this tool is not able to tell whether a file is is malicious or not, its reports can be used for manual analysis or even to generate a set of features for automatic classification.

Wepawet. One of the first academic tools specifically designed to detect PDF files is Wepawet [15] [1], an online malware detection system that detects malicious URL and PDF files. It extends the approach introduced by CWSandbox by including a features extraction and classification system. It is a machine learning tool which focuses on Javascript attacks: it extracts, deobfuscates and classifies Javascript code within PDF files. To this end, the tool analyses specific commands associated to malicious files, as well as the order in which those commands are executed. The tool employs a Bayesian classifier, which appears to be good for the purposes of the analysis.

Nozzle. It is a specific tool developed to detect Heap Spraying attacks [8]. Although it has not been specifically designed to detect malicious PDF files, it can be a very useful resource, as many PDF files implement HS attacks.

MDScan. This is one of the most recent and advanced tools created, and it was specifically designed to detect malicious Javascript code inside a PDF file [16]. Basically, it implements a hybrid approach: first, it scans the PDF document in order to retrieve Javascript code, i.e. it searches for the objects related to Javascript routines (static part), and then it executes the code using a Javascript interpreter (dynamic part). The main difference from Wepawet is the way the

malware files are classified: MDScan analyzes the part of the memory in which the Javascript routines are written, and heuristics are adopted to determine whether or not the code is malicious.

PJScan. This tool has been recently developed by Laskov and Šrndić, and it extracts the features used for the classification from the Javascript code embedded in the PDF file, using a static N-gram analysis. It then uses a one-class SVM to classify the files. This tool only analyzes files that have embedded Javascript code [17].

Due to their public availability and the possibility of performing massive scans, we have been able to compare the performance of our tool with Wepawet and PJScan.

5 A New PDF Detector

In this paper a new tool called PDF Malware Slayer (PDFMS) is presented. PDFMS is an advanced tool to the detection of malicious PDF files, based on machine learning. This tool is composed by:

- A data retrieval module, which retrieves the files for the training/testing phases.
- A feature extractor module, which determines the type of features used by the classifier.
- The classifier itself.

We will now focus on the methodology adopted to develop the feature extractor module, as well as the guidelines behind the choice of the classifier. The data retrieval module will be analyzed in section 6.1.

5.1 Feature Extraction

This is the most important phase of the project, as an incorrect choice of the features would not make the classifier work well. Malicious code is always contained inside a data stream, which is compressed. However, analyzing data stream as a whole can be quite complex, due to wide variety of PDF objects. Moreover, focusing on a particular kind of object (e.g. Javascript or ActionScript) may allow to detect only a portion of PDF attacks. To overcome this problem, we characterize PDF files according to the set of embedded keywords. It is worth noting that a PDF reader needs to recognize some specific PDF keywords in order to execute actions, opening images, and so on. Thus, the occurrence of each keyword can be useful to understand the high-level behavior of PDF readers when a PDF file is opened, and discriminate between malicious and legitimate PDFs.

Towards this goal, we perform two steps:

1. Let us consider two sets, composed by malicious or legitimate PDFs, respectively. For each of these two sets, we enumerate PDF keywords and, for each keyword, its relative frequency is computed (multiple occurrences of a keyword within a single file are considered only once). Then, for either malicious or benign files, we identify the group of keywords having the highest frequency (i.e. highest centroid) by means of a K-Means clustering with $K = 2$. The *base* feature set is defined by the union of the corresponding two clusters. For each keyword within this set, we add its obfuscated version (if it appears at least once on benign or malicious files) to build the final feature set. For instance, if keyword /JS is within the *base* feature set, and there is a obfuscated version of this keyword, namely /JSoffus, we will include /JSoffus in the final feature set.
2. The feature vector for each PDF file is obtained by computing the frequency of each keyword in the feature set.

Fig. 2 shows the structure of the feature extractor.

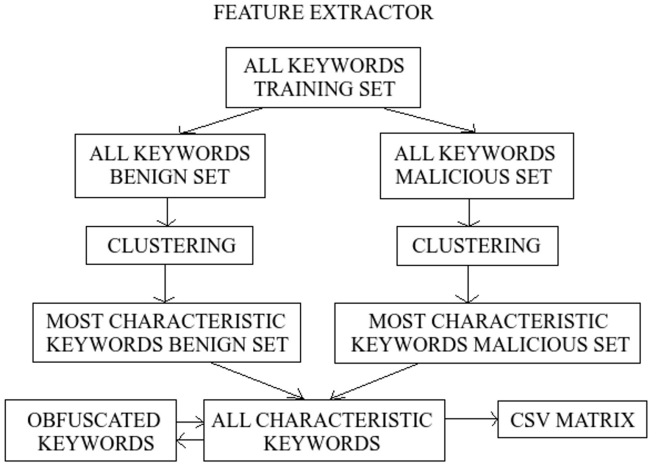


Fig. 2. PDFMS Feature Extractor

5.2 Classification

The type of features chosen for this problem does not provide particular hints for the choice of a classification algorithm. During the experimental phase, different classifiers such as Naive Bayes, SVM and Decision Trees (in particular using Random Forests), have been considered. After having determined which classifier has the best accuracy and stability on the training model, we have performed a comparison between the accuracy of our system on the test set and the one of other systems, both commercial and academic. See the next section for more information.

6 Results

6.1 Data Collection

Basically, both malicious and benign files are used for the training phase. This choice was made because there are not only differences between malicious and benign PDF files, but also common points. Hence, using both classes for training could be a reasonable choice. The data has been retrieved from:

- An interface to the Yahoo search engine [18], which randomly downloaded PDF files using random keys from a dictionary (benign source).
- A huge dataset provided by the Contagio team [11] (benign and malicious files). This choice was made because finding a lot of malicious samples on search engines is a very difficult task, as most of PDF malicious files are sent using spam[3]. Therefore, using the Yahoo Search Engine to retrieve malicious files would not bring good results.

The dataset obtained in this way was divided into a training set and a test set. The division was made randomly, although using two basic criteria:

- The number of the benign and malicious files in the training set must be the same, in order to attain a balanced training set.
- The number of the test set samples must be high, in order to provide reliable results. Hence, we decided to use a number of test samples close to the number of the training ones.

Fig. 3 shows the structure of the data retrieval module.

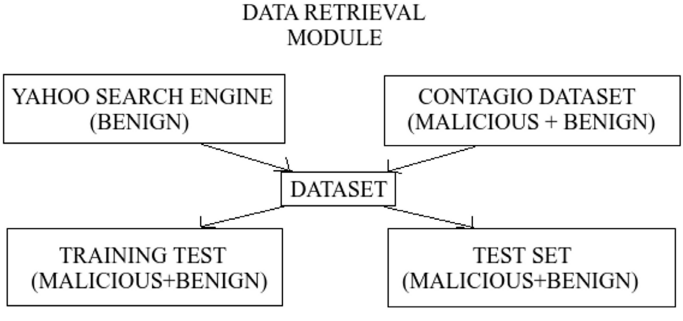


Fig. 3. PDFMS Data Retrieval Module

The 25% of the dataset has been obtained from the Yahoo Search Engine, whilst the other 75% from the Contagio dataset. The total number of files is

[3] With the term SPAM we define advertisements sent through e-mail, blogs or search engines. In particular, an attack performed using SPAM usually requires the user to download a file related to the advertisement and open it (most of times it is an executable one, but also PDF ones).

21146: 11157 malicious files and 9989 benign ones. The training set is composed by 6000 benign plus 6000 malicious files, whilst the remaining samples formed the test set. The malicious files in the training set contain a high variety of attacks (mainly Javascript and ActionScript embedded code), which implement all the techniques described in Section 3.3. The chosen division brought to 12000 files for the training set and 9146 files for the test set, formed by 3989 benign files and 5157 malicious ones.

6.2 Feature Extraction

The number of objects determined from the malicious part of the training set is 10354, and the highest total frequency of a keyword appearing in the malicious set is 5945. Fig. 4 shows the results of the clustering performed on the malicious part. On the y axis the frequency of the object is reported.

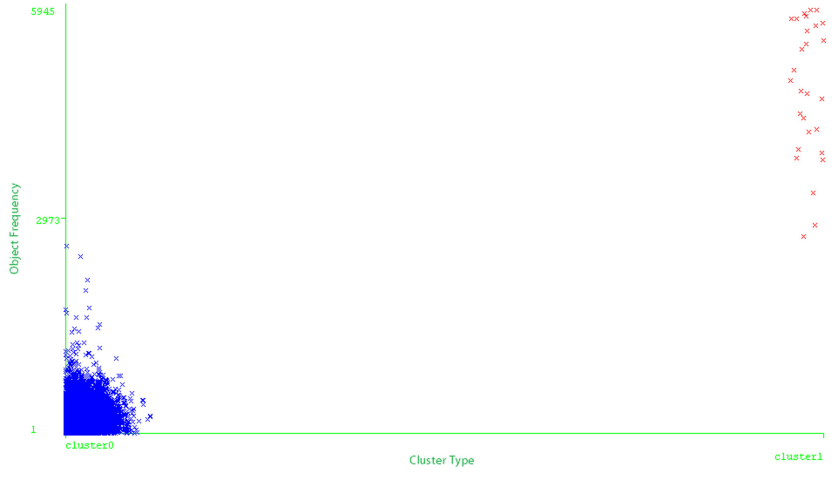


Fig. 4. Malicious training set clustering

As it can be seen, the first 28 objects having higher frequency value (i.e. with the highest occurrence) were considered characteristic by the clusterer.

The number of the objects determined from the benign part of the training set is 650357, and the highest number of occurrences for a single object is equal to 5965. The difference between the malicious part is evident, as genuine files contain a wider variety of objects compared to the malicious ones. Fig. 5 shows the clustering plot for the benign files.

The 156 objects with the highest frequency were considered characteristic by the clusterer (red cluster)[4]. Finally, the objects obtained from the cluster related

[4] One of the last red points seems a bit under the blue one, but this is just because of the jitter, i.e. a random noise introduced to separate the points in order to provide a better visualization of the clusters. With zero jitter the instances are perfectly ordered.

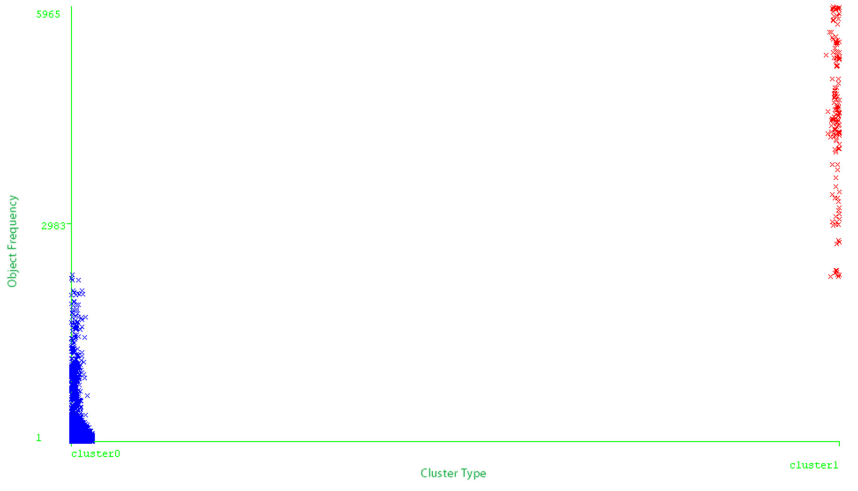


Fig. 5. Benign training set clustering

to the benign files were merged with the ones obtained from the cluster related to the malicious files: this operation returned 168 objects. Considering the presence of the obfuscated instances of the objects considered, the final number of features is 243, as 75 objects out of 168 appear at least once in their obfuscated form.

6.3 Choice of the Classifier

We have performed our tests on Bayesian, SVM, J48 and Random Forests, using a 10-folds Cross Validation repeated 10 times and a split-test (66% training, 34% test) repeated 50 times. Table 1 shows the accuracy (in percentage) attained by different classifiers. The standard deviation is reported between brackets.

Table 1. A comparison of the best classifiers with a 10-folds Cross Validation and a split test

Classifiers	Cross Validation	Split Test (66% training)
Complement Naive Bayes	98.65 (0.32)	98.63 (0.15)
SVM Linear Kernel	99.39 (0.25)	99.29 (0.14)
J48 Decision Tree ($C = 0.25$)	99.59 (0.22)	99.57 (0.14)
Random Forests (18 trees, 165 features)	**99.88 (0.1)**	**99.82 (0.07)**

Random forests provided the highest accuracy. Its accuracy is significantly better than the one provided by other classifiers. This has been proved with a paired T-test with a significance of 0.05.

6.4 Accuracy on the Test Set

After the cross-validation phase, we have analyzed the accuracy of the classifiers on the test set. We have compared the accuracy provided by the proposed PDFMS technique with the 20 most effective antivirus in commerce[5]. The analysis of the test set using these antivirus was made using Virus Total [19], a service which provides the virus scan results from more than 40 antivirus. Table 2 shows the attained results. This table shows also a comparison between PDFMS, Wepawet and PJScan: this comparison is interesting because both tools were developed in an academic environment.

Table 2. A comparison of PDF Malware Slayer accuracy with the other antivirus in commerce

Antivirus	Test Set False Positives	Test Set False Negatives	Total Score
PcTools	0 (0%)	10 (0.19391%)	10
PDF Malware Slayer	**1 (0.02507%)**	**23 (0.446%)**	**24**
GData	33 (0.82728%)	8 (0.15513%)	41
Kaspersky	8 (0.20055%)	35 (0.67869%)	43
Nprotect	6 (0.15041%)	47 (0.91138%)	53
Bitdefender	9 (0.22562%)	49 (0.95017%)	58
Avast5	2 (0.05014%)	59 (1.14408%)	61
Symantec	79 (0.7714%)	31 (1.5319%)	110
Sophos	1 (0.02501%)	134 (2.59841%)	135
ClamAV	26 (0.65179%)	140 (2.71476%)	166
NOD32	4 (0.02501%)	177 (3.43223%)	181
Commtouch	24 (0.60155%)	207 (4.01396%)	231
McAfee-GW-Edition	6 (0.15041%)	259 (5.0223%)	265
Ikarus	1 (0.02507%)	272 (5.27438%)	273
Microsoft	2 (0.05013%)	290 (5.62342%)	292
AVG	32 (0.08022%)	299 (5.79795%)	331
F-Prot	1 (0.02507%)	343 (6.65115%)	344
eTrust-Vet	0 (0%)	385 (7.46558)%)	385
VIPRE	24 (0.60165%)	406 (7.87279%)	430
Fortinet	0 (0%)	530 (10.27773%)	530
Norman	22 (0.55152%)	648 (12.56545%)	670
PJScan	0 (0% out of 25)	483 (11.14958% out of 4332)	483
Wepawet	0 (0% out of 1565)	4664 (90.84534% out of 5134)	4664

Table 3. AND Logic when using PDFMS as a plugin

	PDFMSBenign	PDFMSMalicious
PcToolsBenign	Benign	**Malicious**
PcToolsMalicious	**Malicious**	**Malicious**

[5] Most effective means that have produced the highest accuracy on the test set.

Table 4. Performance improvements using PDFMS as a plugin

Antivirus	Test Set False Positives	Test Set False Negatives	Total Score
PDFMS+PcTools	**1**	**3**	**4**
PcTools	0	10	10
PDF Malware Slayer	1	23	24

Test set False Positives is the number of benign files detected as malicious and Test set False Negatives is the number of malicious files detected as benign; in brackets we provide their percentage values. Total Score is a value given by the sum of false positives and false negatives, and it is used as an index of the total number of errors made by the classifier. As it can be seen, PDFMS performs exceptionally well, since its accuracy is just slightly below the most accurate tool (PCTools antivirus). It performs much better than Wepawet, although our analysis was conditioned by upload errors, probably due to a server overload, and we did not have the possibility to use an offline version of the tool. It also performs better than PJScan, which could not analyze all the test set files because of the internal structure of the tool (see section 4). For these two tools, we have therefore reported only the results related to the files correctly analyzed. To have another interesting idea of the good performance of our PDFMS tool, Fig. 6 shows the ROC chart related to the test set analysis (the Wepawet and PJScan percentage results are related to the correctly analyzed files). On the y axis the percentage of true negatives is reported, whilst on the x axis the percentage of false positive. The more an antivirus stays on the upper left corner of the chart, the more it is considered effective.

Our tool was also designed not only to be used stand-alone, but also as a plug-in for existing antivirus. In particular, we have adopted an AND logic

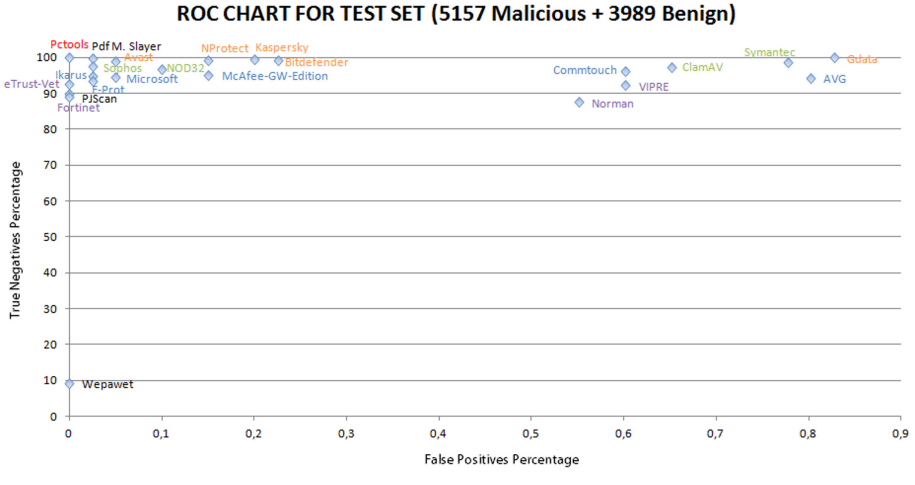


Fig. 6. ROC Chart for test set analysis

approach, in order to improve the malicious files detection rate. We have therefore combined the performance of PDFMS with the ones attained by PC TOOLS, the commercial tool that provided the highest performance.

Table 3 shows the detection mechanism. In other words, classify a sample as benign only if both tools classify it as benign. Table 4 shows the comparison between PDFMS stand-alone, PcTools stand alone and PDFMS used as a plug-in of PcTools. As it can be seen, the malicious detection rate is dramatically improved, making the score index almost perfect. However, there is a trade off, as the false positives score is slightly increased. A zoomed ROC chart (Fig. 7) referred to table 4 better shows this.

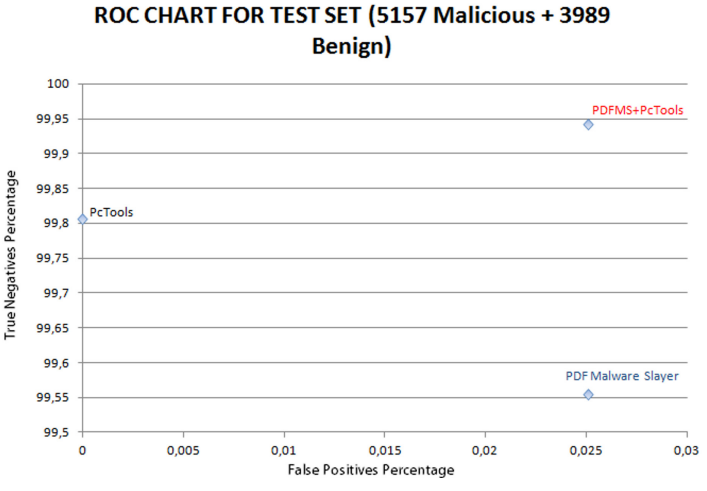


Fig. 7. Zoomed ROC Chart for PDFMS different ways of working

6.5 Weaknesses

Although the system has proved to be very effective, it has some structural weaknesses. Firstly, the same objects that are found in benign files can be also found in malicious ones, meaning that the same object can bring malicious or benign code. What allows PDFMS to correctly establish the maliciousness of the file is the value of the single frequencies of many characteristic objects inside a data stream. Consequently, while PDFMS is able to establish whether or not the PDF file is malicious, it cannot say anything about the type of vulnerability, because it doesn't analyze the code within such a stream. What's more, if an attacker learned how many times certain objects have to appear inside the file in order for it to be considered benign by the tool, it might bypass PDFMS by injecting those specific keywords inside the file. An improvement of the parser is in progress in order to avoid this kind of attack.

7 Conclusions

In this paper, a new system to detect malicious PDF files has been presented. It is a machine learning-based system, strongly related to the internal structure of the file format, easily reproducible, and effective against several types of attacks. It is also possible to use it in different ways according to user needs. The accuracy value shows that the system performs better than the vast majority of commercial antivirus. Moreover, it performs much better than Wepawet, a powerful tool academically developed. In fact, our tool is specialized on the detection of PDF attacks, while Wepawet has been developed to detect a number of threats including malicious PDF files. Thus, specialization appears to be very important to detect this kind of threat. Our tool can also scan any type of PDF file, whilst academic tools such as PJScan can analyze just PDF files carrying Javascript code. The proposed system can be further improved by testing its robustness against new vulnerabilities and improving the parsing process. This tool might also be part of a Multi Classifier System, in which every classifier is specialized in detecting specific threats. As the attacker strategies improve, the challenge for the future is making our security systems more robust against the widest variety of threats, giving them even the possibility to predict new ones.

Acknowledgments. The authors would like to thank the anonymous reviewers for useful insights and suggestions. This research has been carried out within the project "Advanced and secure sharing of multimedia data over social networks in the future Internet" funded by the Regional Administration of Sardinia, Italy (CUP F71J11000690002).

References

1. Wepawet, http://wepawet.iseclab.org/
2. IBM : IBM X-Force 2010 Mid-Year Trend and Risk Report (2010)
3. Symantec : Symantec Global Internet Security Threat Report. Trends for 2009 (2010)
4. Parker, T.: Navigating the Internal Structure of a PDF Document, http://www.planetpdf.com
5. Decalage: PDF Security Isseues, http://www.decalage.info
6. Ogorkiewicz, M., Frej, P.: Analysis of Buffer Overflow Attacks (2004), http://www.windowsecurity.com
7. Ramachandran, V.: Buffer Overflow Primer Video Series, http://www.securitytube.net
8. Ratanaworabhan, P., Livshits, B., Zorn, B.: NOZZLE: A Defense Against Heap-spraying Code Injection Attacks. In: SSYM 2009 Proceedings of the 18th Conference on USENIX Security Symposium (2009)
9. Bania, P.: JIT Spraying and Mitigations. CoRR (2010)
10. Stevens, D.: PDF tools, http://blog.didierstevens.com/programs/pdf-tools/
11. Contagio, http://contagiodump.blogspot.com/
12. Cova, M., http://www.cs.bham.ac.uk/~covam/blog/pdf/
13. Dixon, B., http://blog.9bplus.com

14. Willems, C., Holz, T., Freiling, F.: Toward Automated Dynamic Malware Analysis Using CWSandbox. Journal IEEE Security and Privacy Archive 5(2) (2007)
15. Cova, M., Kruegel, C., Vigna, G.: Detection and Analysis of Drive-by-Downloads Attacks and Malicious Javascript Code. In: Proceedings of International World Wide Web Conference, WWW 2010 (2010)
16. Tzermias, Z., Sykiotakis, G., Polychronakis, M., Markatos, E.P.: Combining Static and Dynamic Analysis for the Detection of Malicious Documents. In: EUROSEC 2011 Proceedings of the Fourth European Workshop on System Security (2011)
17. Laskov, P., Šrndić, N.: Static Detection of Malicious JavaScript-Bearing PDF Documents. In: Annual Computer Security Applications Conference (2011)
18. Yahoo, http://www.yahoo.com
19. Virus Total, http://www.virustotal.com/

Text Categorization Using an Ensemble Classifier Based on a Mean Co-association Matrix

Luís Moreira-Matias[1,2], João Mendes-Moreira[1,2], João Gama[2,3], and Pavel Brazdil [2,3]

[1] Departamento de Engenharia Informática, Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, s/n 4200-465 Porto, Portugal
[2] LIAAD-INESC Porto L.A. Rua de Ceuta, 118, 6º; 4050-190 Porto, Portugal
[3] Faculdade de Economia, Universidade do Porto
Rua Dr. Roberto Frias, s/n 4200-465 Porto, Portugal
{luis.matias,jmoreira}@fe.up.pt, {jgama,pbrazdil}@fep.up.pt

Abstract. Text Categorization (TC) has attracted the attention of the research community in the last decade. Algorithms like Support Vector Machines, Naïve Bayes or k Nearest Neighbors have been used with good performance, confirmed by several comparative studies. Recently, several ensemble classifiers were also introduced in TC. However, many of those can only provide a category for a given new sample. Instead, in this paper, we propose a methodology – MECAC – to build an ensemble of classifiers that has two advantages to other ensemble methods: 1) it can be run using parallel computing, saving processing time and 2) it can extract important statistics from the obtained clusters. It uses the mean co-association matrix to solve binary TC problems. Our experiments revealed that our framework performed, on average, 2.04% better than the best individual classifier on the tested datasets. These results were statistically validated for a significance level of 0.05 using the Friedman Test.

Keywords: Text Categorization, Ensemble Classification, Consensus Clustering, Text Mining.

1 Introduction

In the last decade the Information Retrieval (document management tasks) has attracted a major attention of the machine learning research community due to the high number of electronic documents available on and offline. One of the most relevant tasks is Text Categorization (TC): it consists in labeling automatically a document with a certain category, based on its content.

This problem is solved using supervised classification algorithms. From the document set, a feature space is extracted based on a set of unique, uncommon and frequent terms which are evaluated for each document. Many comparative studies have been presented in the last years to understand which classifiers should be the most adequate to the TC domain problems [1-7].

In the last years, some ensemble approaches were also considered to improve the TC performance. In this paper, we present a distinct methodology to ensemble as

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 525–539, 2012.

many classifiers – distinct algorithms or just the same algorithm with different parameters – as defined into a single one using the mean co-association technique (commonly applied in the consensus clustering area [8]) – the MECAC (Ensemble Classification using Mean Co-Association Matrix).

Our motivation was to build a new ensemble framework returning more than a simple category for a new sample, as many state-of-art TC algorithms (ensemble and individual ones) do. The authors wanted to extract other kind of metrics useful to better understand the results and/or determine how the categories evolve in time (i.e. clusters birth, merge, etc. like it is presented in the novelty detection problems [9]).

We considered four state-of-art classifiers for single-label TC to carry in our experiments: Support Vector Machines with a linear kernel (SVM-linear), k Nearest Neighbors (kNN), Naïve Bayes (NB) and Neural Networks (NNET). Firstly, we build a baseline ensemble method (ENS-b) for comparison against the ensemble approach we propose. ENS-b uses the majority class considered among the base classifiers (we used four models, one model per algorithm). Secondly, we used the same base learners to build two ensembles using MECAC: ENS1, that used all the four models, and ENS2 that used all except NB. Finally, we compared the four base learners, ENS-b, ENS1 and ENS2 using three performance metrics: macro avg./micro avg. F1-measure and Cohen-Kappa to classify document collections of Reuters-21578 dataset [10]. Our experiments show the utility of our methodology for TC: despite the good results presented by all individual classifiers, our best ensemble improved the results of the best individual classifier in each data block by 2.04% (on average). These results were statistically validated for a significance level of 0.05 using the Friedman Test.

This paper is structured as follows. Section 2 states a brief description of the problem and some related work. Section 3 presents formally our approach. Section 4 describes how we tested the methodology to a concrete problem. Firstly we describe the dataset and the preprocessing applied. Secondly, we present the ensemble building process. Finally, we point the metrics we used to evaluate each considered approach. Section 5 presents the experimental setup used and the results obtained. It also presents a discussion about those results. Section 6 concludes and describes the future work we intend to carry on.

2 Problem Overview

Multiple approaches to binary TC were presented in the last decades using some well-known classifiers. However, it is usually difficult to know which one is the best to classify our current text documents [4]. In the literature, there are several comparative studies between distinct classifiers in order to evaluate their performance. In [4], a study to compare Support Vector Machines (SVM), k Nearest Neighbors (kNN) and Naïve Bayes (NB) is presented to perform binary TC. It concludes that all the algorithms should be considered as long as the optimal parameter settings could be used for each one. In [7], SVM, NB, logistic regression and LLSF (Linear Least Square Fit) are also compared. All but NB consistently achieve a top performance. Another algorithm usually considered for this task is the neural network (NNET) one [2].

Despite this straight forwarding knowledge achieved with single supervised learning techniques, the community attention changed its main focus in the last years: the researchers tend to use complex and advanced techniques to solve these problems. Many hybrid techniques to build ensembles of classifiers for TC have been recently used [11]: i) using different subsets of training data with a single learning method, ii) using different parameter settings with a single learning method (e.g. using different initial weights for each neural network in an ensemble) and iii) using different learning methods. The scenario considered in this work is the iii). It is commonly observed that the ensemble accuracy is superior when compared to its base classifiers: in [6] it is used the Dempster's rule of combination to ensemble SVM, kNN and Rocchio [12]: the ensemble accuracy is, on average, 2.68% better than the best base classifier; in [5] the authors present a framework to combine multiple NNET and it is compared to kNN, SVM, single-NNET and Decision Tree (DT) using a single dataset; it achieves an improvement of 2,7% (of F1 measure); the use of AdaBoost.MH and AdaBoost.MH^{KR} to solve multi-label TC problems in [13] shows gains from 1 to 4% (F1).

We found motivation to develop this work due to a particular issue: the majority of the existing approaches to TC give only one kind of information about a new sample (i.e., a text document): the category. How can we know if this is the most suitable one? Should this sample belong to a new and inexistent system category? The unsupervised learning techniques can provide different kinds of information [14-16] about their categories (i.e. clusters). We applied a hybrid approach of both supervised and unsupervised learning trying to get the best from the two learning approaches: different kinds of information and better classification accuracy.

The contribution of this work is a new framework to combine results from different classifiers using a weighted mean co-association matrix. At author's best knowledge, **this kind of methodology was never applied to build a classifier ensemble on TC problems**.

3 Methodology

The problem we solve here is the construction of an ensemble to do binary classification of single-labeled text documents. This is done using a weighted mean co-association matrix (firstly proposed to do consensual clustering in [8]) that **measures the consensus between all the classifiers to attribute the same class to all existing pairs of text documents**. This matrix contains the distances between every pair of documents considered. Finally, we use clustering to separate the documents into distinct categories and SVM-linear to label the clusters. This general idea for our methodology (MECAC) can be divided into three simple steps: 1) the classifiers training; 2) the calculus of the agreement matrix between the test documents and the input classifiers; 3) the documents clustering. The MECAC is briefly presented in Fig. 1 and it is be described in detail further in this section.

Let $X=\{x1, x2...xn\}$ be a set of n single labeled two classes (*Class1* and *Class2*) text documents and $C=\{c1, c2...ck\}$ be a set of k classifiers of interest. These classifiers can be obtained using only one algorithm (with different parameter settings, using different training sets or different feature subsets) or using different algorithms. Then, the classifiers are combined.

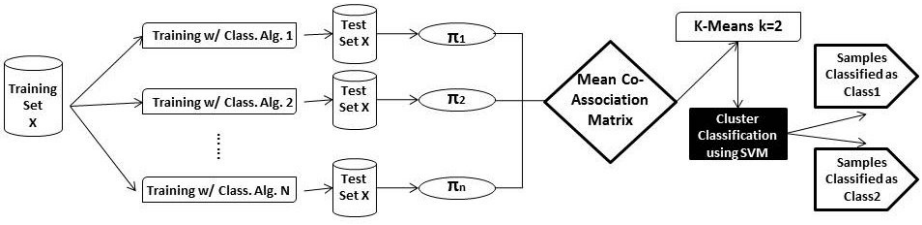


Fig. 1. MECAC: Ensemble Classification using Mean Co-Association Matrix

Step 1 – The Classifiers Training

A set of classifiers is generated by applying k classification algorithms to our training set *X*. $\pi = \{\pi_1, \pi_2 \ldots \pi_k\}$ contains the class determined by the classifiers to our test set."

Step 2 – The calculus of the Agreement Matrix

We present an algorithm that creates **a new distance measurement based on the agreement between the k classifiers**. Let *M* (*s* x *s*) be a quadratic matrix as large as the number of test documents. The values in the matrix measure the agreement between the classifiers to categorize equally both documents. The **mean co-association matrix** [8], represents the classification agreement between all classifiers. The values in the matrix are obtained as follows:

$$M(i,j) = \begin{cases} 2^{a-1} \text{ if } a > 0 \\ 0 \text{ if } a = 0 \end{cases}, i,j \in \{1, \ldots, s\} \quad (3.1)$$

where *a* is the number of classifiers that classified equally the documents *i* and *j*, independently on the true class of both documents. This matrix measures the agreement of the classifiers to label equally each pair of documents. This information is directly about the similarity between each pair of documents - then the category is calculated based on it. **The weights** (the pow used to calculate the agreement instead of a simple sum) **were introduced to enhance the agreement achieved between all classifiers**: it is measured exponentially to express its consensus relevance. **This weighted measure is one of the main contributions of this work because it innovates the calculus of the distance between text documents for binary TC** (the simple sum proposed in [8] performs worst in the current context). Such distance highlights the agreement between the classifiers (i.e. the similarity between the documents).

After its normalization (we divide all the values in the matrix for its maximum value), it is possible to transform the matrix *M* into the quadratic matrix $\mathcal{D}$ (*s* x *s*), as follows:

$$\mathcal{D} = 1 - \frac{M}{ma} \quad (3.2)$$

where *ma* is the previously referred maximum.

Step 3 – The Document Clustering

We use the matrix $\mathcal{D}$ as input for a clustering algorithm of interest like *k-means*. We split the test set into 2 unlabeled partitions because this is a binary classification

problem. To label them, we used a known robust classifier for binary classification: SVM with a linear kernel [2, 4, 11]. We set the labels to the partitions by choosing the resulting majority class for the given partition, training the algorithm with the same training set used to train the classifiers in *C*. A pseudo-code representation of our framework is presented in Fig. 2.

Procedure *Ensemble Classification using Mean Co-Association Matrix (MECAC)*

Input:
a set of ***n*** documents to categorize ***X***={*x1, x2,…,xn*}
a set of ***k*** classifiers ***C***={*c1, c2,…,ck*}
an user-defined percentage ***p*** to form the test set

Declarations:
s is a integer representing the number of documents in the test set ***(n*p)***
class is a matrix of labels: classifiers*labels ***(k*s)***
m is an integer quadratic matrix ***s*s*** defined with zeros

Body:

1. Define the test set ***S*** using **s** documents in ***X***
2. Define the training set ***T*** with the remaining ***t*** documents in ***X***
3. **For** each ***ci*** in ***C***
   {
   3.1 Train the classifier ***ci*** using the categorized documents in ***T***
   3.2 Use the trained classifier ***ci*** to categorize the documents in ***S***
   3.3 Save the resulting labels in ***class[i,]***
   }
4. **For** each ***o*** between **1** and ***s***
   For each ***j*** between **1** and ***s***
   For each ***b*** between **1** and ***k***
   For each ***i*** between ***b+1*** and ***k***
   IF (***class[b,o]*** == ***class[i,j]***)
   IF(***m[o,j]==0***)
   m[o,j]=1;
   ELSE
   m[o,j]=m[o,j]*2;
5. Use ***m*** as input of ***k-means*** algorithm to form ***2 clusters of documents***: ***k1*** and ***k2***.
6. Use the ***SVM-linear*** algorithm trained on the ***T*** set to classify the documents in ***k1*** and ***k2***.
7. The categories corresponding to each cluster are chosen by determining the majority class obtained in each one of them in the previous step.

Fig. 2. Pseudo code of MECAC, the proposed ensemble methodology

4 Methodology Application

In this section, we describe how we carried out our experiments. Firstly we describe the data set used and the preprocessing applied to it. Then, we briefly review how we adapted our methodology to build our ensemble in this case and finally present the metrics applied to compare our methodology *versus* the remaining approaches.

4.1 Dataset

The data contained in the "Reuters-21578, Distribution 1.0 corpus" is freely available for experimentation purposes from [10]. It consists of news stories appeared on the Reuters newswire in 1987. There are 5 groups of categories in the dataset but just the TOPICS group is commonly used in TC experimental research. These groups have 135 categories and a total of 12902 documents.

In Table 1 the categories per dataset and the total number of documents per category is shown. In this work we considered just 7 categories out of the total 135. Those documents were used to form five distinct datasets (DS1 to DS5) with pairs of categories (binary classification).

Table 1. Datasets description and relation with the categories

Category	Nr. Docs.			
Coffee	143			
Crude	334	**Dataset**	**Class1**	**Class2**
Grain	401	**DS1**	***Wheat***	***Money-Fx***
Interest	335	**DS2**	***Sugar***	***Interest***
Money-Fx	344	**DS3**	***Sugar***	***Crude***
Sugar	180	**DS4**	***Interest***	***Coffee***
Wheat	208	**DS5**	***Grain***	***Crude***

4.2 Preprocessing

One of the most important stages in TC is the preprocessing one. The uncommon words must be extracted from each document. Then, such words are used as features for the classification task.

We used the *tm_map* function from R software [17]. Firstly, we removed the existing XML code. Then, we turned the text into plain text. Thirdly, we removed the stop words and then the extra whitespaces. Finally, we converted the text to lowercase and we removed the punctuation and the existing numbers.

Then, we did feature selection using three metrics [1, 2, 4]: (1) the Minimum Word Length (MWL), the minimum term length to consider it informative; (2) the Minimum Document Frequency (MDF), the minimum number of documents containing this term in their corpus to consider it informative; and (3) the Information Gain (IG) to impose an ordering on a set of attributes. Finally, the terms are weighted by term frequency and inverse document frequency (*tfidf*) [18]. We used these heuristics due to its simplicity and good performance [1, 4, 19].

4.3 Ensemble Building

This section demonstrates how we have adapted the previously presented methodology to the problem described in Section 2. The *k-means* was chosen as clustering algorithm due to its simplicity, efficiency and efficacy as well as its many applications over the last decades [21-23]. We set both the MWL and the MDF as 3. The min.info parameter of the IG heuristic was set to 0.005. These parameters where used in all the experiments described in this paper. We developed all the work using

the R-project and the following supervised classification algorithms: NNET, kNN, SVM with a linear kernel and NB. This choice was made due to its high popularity and performance in TC [1, 2, 4-6, 8, 19].

The R packages used to implement the algorithms are identified in Table 2. In general, the functions default parameters (from the R) were used. The exceptions, i.e., the user-defined parameters, are described next.

The NNET implementation used to carry out the experiments had 4 main parameters: the size (measures the number of units in the hidden layer); the *decay* and *rang* (they set the weight decay and measure the initial random weights, respectively) and the *maxit* (it sets the maximum number of iterations). They were set to 2, $5e^{-4}$, 0.1 and 400, respectively.

Table 2. Identification of the R packages used to test the considered classifiers

Classification Algorithm	*R package*	*Package Reference*
Neural Network	[nnet]	[20]
K Nearest Neighbors	[class]	[20]
Support Vector Machine	[e1071]	[21]
Naïve Bayes	[RWeka]	[22]

The classifiers described were used to build three ensembles: ENS1 is an ensemble build using MECAC from all base classifiers and; ENS2 only uses the three most accurate base classifiers as input to MECAC. Finally, we built a baseline ensemble to compare our approaches with – ENS-b. It uses the majority class among all classifiers. In a tie scenario, the class is randomly chosen. Some metrics were used to compare the ensembles accuracy *versus* each individual algorithm considered.

4.4 Evaluation Metrics

The metrics to evaluate prediction accuracy uses the confusion table with the notions of true positive, false positive, false negative and true negative (TP, FP, FN and TN). However, these notions are not meaningful in TC. A contingency table is presented in Table 3 naming the correspondences between the two classes and that nomenclature.

Table 3. Contingency table for TC

	Classifier Observation		
Document Classification	*Predicted Class1*	**TP** *(hits)*	**FP** *(incorrect classif.)*
	Predicted Class2	**FN** *(incorrect classif.)*	**TN** *(hits)*
Total		Observed *Class1*	Observed *Class2*

Two widely used metrics in TC to test classifiers' accuracy are the **macro averaged F1 measure [3]** and **the micro averaged one [23].** The macro avg. F1 is computed locally over each category. It can be obtained as a weighted average of two other metrics: **MPrecision** and **MRecall** (averages of the precision and recall for both classes). In micro-averaging, F1 is obtained by computing globally over all category decisions. The referred metrics can be obtained as follows:

$$\text{MPrecision} = \left(\frac{\frac{\text{TP}}{\text{TP+FP}} + \frac{\text{TN}}{\text{TN+FN}}}{2} \right), \text{MRecall} = \left(\frac{\frac{\text{TP}}{\text{TP+FN}} + \frac{\text{TN}}{\text{TN+FP}}}{2} \right) \quad (4.1)$$

$$\text{Macro avg. F}_{1\text{ measure}} = 2 \times \frac{\text{MPrecision} \times \text{MRecall}}{\text{MPrecision+MRecall}} \quad (4.2)$$

$$\omega = \frac{\text{TP}}{\text{TP+FP}} = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M} (TP_i + FP_i)}, \rho = \left(\frac{\text{TP}}{\text{TP+FN}} \right) = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M} (TP_i + FN_i)} \quad (4.3)$$

$$\text{Micro avg. F}_{1\text{ measure}} = 2 \times \frac{\omega\rho}{\omega+\rho} \quad (4.4)$$

We also considered the kappa coefficient (or Cohen kappa) [24] as accuracy metric. It is a statistical measure of inter-rater agreement for qualitative (categorical) events. It is computed as:

$$\text{k} = \frac{\text{P(a)} - \text{P(e)}}{1 - \text{P(e)}} \quad (4.5)$$

where $P(a)$ is the relative observed agreement between the predicted and the actual categories. $P(e)$ is the hypothetical probability of chance agreement. The observed data is used to calculate the probabilities of each observer randomly identifying each category. If the classifiers are in complete agreement then $k = 1$. If there is no agreement among the predictions (other than what would be expected by chance) then $k < 0$.

5 Results

In this section, the results obtained with our dataset are presented, statistically validated and discussed. Our experimental setup was the well-known 5-fold cross validation. We used it for each one of the five datasets considered, forming 25 data blocks. We used the results obtained in each partition of the cross validation process with both F1 measures (macro and micro) and the Cohen Kappa. In Fig. 3, an averaged comparison between all the classifiers using the three considered metrics is displayed. The results for both F1 measures are presented in detail in Table 4 and Table 5.

Secondly, we validated these results using the Friedman Test like we present below. Finally, we discuss the experiments achievements.

5.1 Results Validation (Friedman Test)

The statistical validation of these results was done using the Friedman rank test as proposed by Iman and Davenport [25]. We have compared the best individual classifier (SVM-linear), our best ensemble (ENS2) and our baseline ensemble ENS-b using the results obtained through the macro averaged F1-measure on the 25 data blocks extracted. The ranks obtained are presented in Table 6.

The P-value obtained for the null hypothesis of equivalence between the three predictors was 0.01256. This hypothesis was rejected for a significance level of 0.05.

The post-hoc analysis is presented in Fig. 4: the tests validated for a significance level of 0.05 are displayed in white and the remaining in grey. The boxes represent the variance as well as the lower and upper limits of the tests.

The positive significance in the left hand box means that the ENS2 is significantly better than ENS-B and the negative in the right hand means that the SVM is significantly worse than ENS2, **demonstrating that our method is significantly superior to the remaining ones for binary text classification problems.**

Table 4. Performance obtained from the seven considered algorithms on the 25 blocks extracted using the Macro-Averaged F1-Measure. The ENS-2 performed, on average, 2,04% better than the best individual classifier: SVM.

Block	SVM	kNN	NB	Nnet	ENS1	ENS2	ENS-B
DS1-1	0.8950	0.8901	0.6720	0.8978	0.9058	0.9058	0.8750
DS1-2	0.9044	0.8811	0.7442	0.8699	0.9138	0.9138	0.9058
DS1-3	0.9230	0.8776	0.6822	0.9170	0.9327	0.9327	0.9327
DS1-4	0.9133	0.8603	0.5496	0.9133	0.9133	0.9133	0.8902
DS1-5	0.9037	0.8621	0.7218	0.9532	0.9335	0.9335	0.9242
DS2-1	0.9271	0.9373	0.8594	0.9353	0.9193	0.9373	0.9271
DS2-2	0.9140	0.9011	0.8264	0.9429	0.9136	0.9271	0.9373
DS2-3	0.9067	0.9067	0.8308	0.9371	0.9067	0.9067	0.8605
DS2-4	0.9140	0.8767	0.8504	0.9420	0.9203	0.9476	0.9679
DS2-5	0.9034	0.9346	0.8943	0.9489	0.9566	0.9465	0.9275
DS3-1	0.9271	0.8409	0.8957	0.9037	0.9044	0.9271	0.9271
DS3-2	0.9149	0.9281	0.8415	0.9496	0.9149	0.9149	0.9044
DS3-3	0.9067	0.8455	0.9139	0.9065	0.8752	0.8836	0.8940
DS3-4	0.9373	0.9313	0.7619	0.9571	0.9350	0.9476	0.9476
DS3-5	0.8932	0.8604	0.8026	0.9118	0.9230	0.9355	0.8922
DS4-1	0.9024	0.9043	0.8571	0.8421	0.8868	0.8865	0.8341
DS4-2	0.9388	0.8456	0.8627	0.8846	0.8452	0.9267	0.8470
DS4-3	0.9510	0.9145	0.7170	0.8772	0.9197	0.9510	0.8879
DS4-4	0.9145	0.8944	0.8302	0.8519	0.8901	0.9267	0.8497
DS4-5	0.9388	0.8705	0.8889	0.8727	0.9283	0.9388	0.9388
DS5-1	0.8067	0.7560	0.7193	0.7869	0.7840	0.8229	0.8090
DS5-2	0.8133	0.7065	0.6441	0.8305	0.8033	0.8385	0.8105
DS5-3	0.8620	0.8336	0.8673	0.7692	0.7967	0.8540	0.8682
DS5-4	0.8705	0.8482	0.8595	0.8444	0.9008	0.9249	0.9113
DS5-5	0.8772	0.8358	0.7879	0.8661	0.8636	0.9052	0.8980

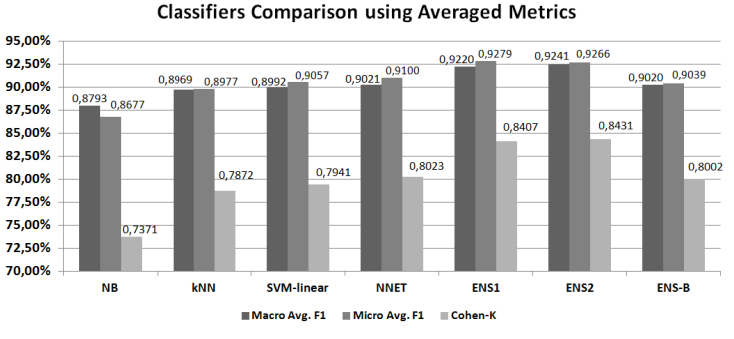


Fig. 3. Comparison between the individual classifiers and the ensembles using all metrics

Table 5. Performance obtained from the seven considered algorithms on the 25 blocks extracted using the Micro-Averaged F1-Measure

Block	SVM	kNN	NB	Nnet	ENS1	ENS2	ENS-B
DS1-1	0.9241	0.9167	0.6720	0.9241	0.9315	0.9315	0.9103
DS1-2	0.9306	0.9150	0.7442	0.9091	0.9371	0.9371	0.9315
DS1-3	0.9429	0.8923	0.6822	0.9333	0.9496	0.9496	0.9496
DS1-4	0.9362	0.9041	0.5496	0.9362	0.9362	0.9362	0.9200
DS1-5	0.9296	0.8921	0.7218	0.9650	0.9510	0.9510	0.9444
DS2-1	0.9504	0.9571	0.8594	0.9353	0.9420	0.9571	0.9504
DS2-2	0.9420	0.9275	0.8264	0.9429	0.9437	0.9504	0.9571
DS2-3	0.9371	0.9371	0.8308	0.9371	0.9371	0.9371	0.9091
DS2-4	0.9420	0.9118	0.8504	0.9420	0.9412	0.9640	0.9778
DS2-5	0.9333	0.9565	0.8943	0.9489	0.9710	0.9635	0.9466
DS3-1	0.9504	0.8722	0.8957	0.9037	0.9362	0.9504	0.9504
DS3-2	0.9429	0.9496	0.8415	0.9496	0.9429	0.9429	0.9362
DS3-3	0.9371	0.8741	0.9139	0.9065	0.9155	0.9231	0.9296
DS3-4	0.9571	0.9583	0.7619	0.9571	0.9571	0.9640	0.9640
DS3-5	0.9254	0.9037	0.8026	0.9118	0.9385	0.9559	0.9265
DS4-1	0.8400	0.8571	0.8571	0.8421	0.8235	0.8302	0.7547
DS4-2	0.9057	0.7586	0.8627	0.8846	0.7860	0.8846	0.7692
DS4-3	0.9259	0.8727	0.7170	0.8772	0.8772	0.9259	0.8421
DS4-4	0.8627	0.8400	0.8302	0.8519	0.8302	0.8846	0.7500
DS4-5	0.9057	0.8077	0.8889	0.8727	0.8889	0.9057	0.9057
DS5-1	0.7521	0.6903	0.7193	0.7869	0.7840	0.7937	0.7778
DS5-2	0.7387	0.6182	0.6441	0.8305	0.8033	0.8033	0.7705
DS5-3	0.8235	0.7899	0.8673	0.7692	0.7967	0.8167	0.8333
DS5-4	0.8527	0.8254	0.8595	0.8444	0.9008	0.9173	0.9023
DS5-5	0.8615	0.8125	0.7879	0.8661	0.8636	0.8906	0.8837

5.2 Discussion

Some of the results presented confirm previous studies [2, 4, 7]: Fig. 3 shows that SVM performs, on average, better than kNN and NB algorithms, confirming the good results of SVM for binary classification, as reported in the literature [4]. It does not share this characteristic with the other two classifiers which are known for their good results in multi-class problems. However, several comparative studies in TC also compare algorithms known as good binary classifiers against other algorithms known as good multi-class classifiers [2, 4, 7].

It is possible to conclude directly from Table 4 and Table 5 that both ensembles (ENS1 and ENS2) present, in all datasets, better or equal results than the best base classifier. Anyway, all classifiers presented an excellent performance for the datasets considered (all F1- measure values are above 78%). The ensembles built also present a superior performance to our baseline ensemble – ENS-b, the majority class one – in both metrics considered. On average, our best ensemble **ENS2 performed 2.04% better than the best individual classifier in each block** (using the F1-macro aver. metric).**This methodology presents two advantages** that distinguish it among other ensemble approaches also used in TC. Firstly**, it can be run using parallel computing**: the classifiers can operate independently in different machines and the results can be

Table 6. On the right hand, the ranks of the Friedman test. On the left hand, the F1 obtained for each fold considered

	Data Group			Friedman Rank		
Block	**SVM**	**ENS2**	**ENS-B**	**SVM**	**ENS2**	**ENS-B**
DS1-1	0.8950	0.9058	0.8750	2	1	3
DS1-2	0.9044	0.9138	0.9058	3	1	2
DS1-3	0.9230	0.9327	0.9327	3	1.5	1.5
DS1-4	0.9133	0.9133	0.8902	1.5	1.5	3
DS1-5	0.9037	0.9335	0.9242	3	1	2
DS2-1	0.9271	0.9373	0.9271	2.5	1	2.5
DS2-2	0.9140	0.9271	0.9373	3	2	1
DS2-3	0.9067	0.9067	0.8605	1.5	1.5	3
DS2-4	0.9140	0.9476	0.9679	3	2	1
DS2-5	0.9034	0.9465	0.9275	3	1	2
DS3-1	0.9271	0.9271	0.9271	2	2	2
DS3-2	0.9149	0.9149	0.9044	1.5	1.5	3
DS3-3	0.9067	0.8836	0.8940	1	3	2
DS3-4	0.9373	0.9476	0.9476	3	1.5	1.5
DS3-5	0.8932	0.9355	0.8922	2	1	3
DS4-1	0.9024	0.8865	0.8341	1	2	3
DS4-2	0.9388	0.9267	0.8470	1	2	3
DS4-3	0.9510	0.9510	0.8879	1.5	1.5	3
DS4-4	0.9145	0.9267	0.8497	2	1	3
DS4-5	0.9388	0.9388	0.9388	2	2	2
DS5-1	0.8067	0.8229	0.8090	3	1	2
DS5-2	0.8133	0.8385	0.8105	2	1	3
DS5-3	0.8620	0.8540	0.8682	2	3	1
DS5-4	0.8705	0.9249	0.9113	3	1	2
DS5-5	0.8772	0.9052	0.8980	3	1	2
			Average:	**2.22**	**1.52**	**2.26**

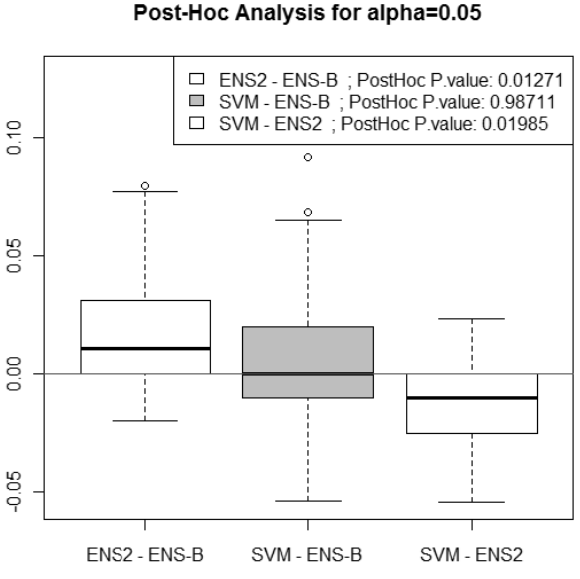


Fig. 4. Post-Hoc Analysis for the considered algorithms

concentrated in a single computer to ensemble them, saving processing time – many state-of-art ensemble methodologies (e.g., boosting) have not such characteristic. For a large set of text documents, this characteristic can provide a true major advantage *versus* the majority of the ensemble frameworks proposed in TC [5, 6, 12, 13].

Secondly, **we can extract several statistics from the obtained clusters.** These statistics can be useful to validate the results that, commonly, are not possible to obtain neither using other ensemble methods. In Table 7 we present three evaluation metrics for the clusters obtained in our datasets.

We used the *clusters.stats* function on the [fpc] R package to proceed with this experiments. The Pearson Gamma [15] (normalized gamma) is a metric that measures the correlation between the distances and a 0-1 vector where 0 means the same cluster and 1 for different clusters. The Entropy [16] (also called variation of information) measures the amount of information lost and gained in changing from clustering C to clustering C'. The Dunn index [14] aims to identify dense and well-separated clusters.

The clusters obtained present, in general, a good quality. We want to enhance the high correlations obtained in the Pearson Gamma to justify it. However, there are several types of metrics that can be extracted. To see more about this issue, the author should read the section 4 in [15]. Such metrics can be accurately used to detect new and/or unknown categories in the system (i.e. novelty detection [26]).

The ensemble with the best performance is the ENS2. This ensemble only uses kNN, NNET and SVM-linear classifiers while ENS1 has an additional base classifier: NB. Since NB is globally the worst among the four base classifiers (Fig. 3), this explains the worse results of ENS1 against ENS2. Despite these results, the authors believe that increasing the number of distinct algorithms used in the ensemble will increase its accuracy but we cannot sustain this based in this specific study.

Some questions remain open: 1) are all the base classifiers of the ensemble useful in all the input space or better results could be obtained by selecting locally the subset of classifiers to predict each given example? 2) Can MECAC be useful in other binary classification problems than TC ones? 3) Can we use the extracted statistics to do novelty detection on the system categories? The dynamic selection of classifiers is an issue already explored in other research areas [27]. The multi label classification is an important problem in TC that can be addressed by MECAC…but how well does it performs in that contest? The novelty detection in TC is not a new topic [26] but can our methodology be also accurate in such task?

All these issues should be explored for TC in our future research.

Table 7. Statistics about the clusters obtained using co-mean association matrix ensemble method

DataSet	PearsonGamma	Dunn	Entropy
DS1	0.8578	0.5714	0.6249
DS2	0.8894	0.5714	0.6320
DS3	0.8330	0.5714	0.6273
DS4	0.9371	0.6667	0.5771
DS5	0.8829	0.5000	0.6852

6 Conclusions and Future Work

In this paper, we proposed a new ensemble method for classification to improve the accuracy of TC (single-label documents to be categorized into a binary classification problem).

The MECAC (Ensemble Classification using Mean Co-Association Matrix) algorithm uses **the mean co-association matrix**, usually used in consensual clustering problems [8]. We have used this method for the resolution of a TC problem. However, it can be used for the resolution of any binary classification problem. Different base learners can be used. At author's best knowledge, such approach was never considered for classification and, consequently, it was never considered for TC problems.

To test it, we decided to use four different classifiers with the same preprocessing and parameters: k Nearest Neighbors, Neural Networks, SVM with a linear kernel and Naïve Bayes. We compared it with three different ensembles: a common baseline, ENS-B that uses the majority class voted among all individual classifiers; ENS1, that used all the referred classifiers as input of our methodology and ENS2 that used all like ENS1 except Naïve Bayes. Finally, we compared those using well known accuracy metrics (both macro and micro averaged F1- measure and Cohen-K) on five datasets of interest. This research pointed out two advantages of this methodology over other ensembles used in TC: 1) **it can be run using parallel computing -** which other commonly used TC ensemble classifiers cannot **-** and (2) **we can extract** useful **statistics from the obtained clusters** , that are not available neither using other ensemble approaches nor using individual classifiers.

Our results also demonstrated that our methodology is a real contribution to the practical application of TC: **our ensemble performed on average 2.04% better than the best individual classifier** and this results were **statistically validated using a significance level of 0-05**.

In author's opinion, this methodology still has some points to work out in its different steps like 1) pruning classifiers by choosing them dynamically for each given example or 2) reduce the well-known k-means' random start effects. However, **we want to highlight the main contribution of this work**: the introduction of **the mean co-association matrix is a new and unused way to simultaneously measure the similarities between a pair of text documents and to define an ensemble of classifiers, improving the decision process**. The work in this paper is a validation of this concept. Therefore, our studies pointed out new issues on this research topic:

- *How this approach performs in other problems than TC?*
- *Can this approach be successfully adapted for multi-class problems*?
- *How can we use this statistics to find out when you need to create new categories in our system?*

The last question is important to highlight one possible main advantage of MECAC facing other ensembles in a TC streaming classification problem over the time: in author's opinion, **the obtained clusters can also be used to discover whether it is**

necessary to create a new category for a new sample. It could be done using the extracted metrics like it is proposed in other novelty detection research works [9].

Experiments will be carried out to proceed with this work.

Acknowledgements. We would like to thank the support of the project Knowledge Discovery from Ubiquitous Data Streams (PTDC /EIA-EIA/098355/2008).

References

1. Yang, Y., Pedersen, J.: A Comparative Study on Feature Selection in Text Categorization. In: ICML 1997, pp. 412–420 (1997)
2. Yang, Y., Liu, X.: A Re-Examination of Text Categorization Methods. In: 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 42–49 (1999)
3. Yang, Y.: An Evaluation of Statistical Approaches to Text Categorization. Information Retrieval 1, 69–90 (1999)
4. Colas, F., Brazdil, P.: Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks. In: Artificial Intelligence in Theory and Practice, pp. 169–178 (2006)
5. Cho, S., Lee, J.: Learning Neural Network Ensemble for Practical Text Classification. In: Liu, J., Cheung, Y.-m., Yin, H. (eds.) IDEAL 2003. LNCS, vol. 2690, pp. 1032–1036. Springer, Heidelberg (2003)
6. Bi, Y., Bell, D.A., Wang, H., Guo, G., Greer, K.: Combining Multiple Classifiers Using Dempster's Rule of Combination for Text Categorization. In: Torra, V., Narukawa, Y. (eds.) MDAI 2004. LNCS (LNAI), vol. 3131, pp. 127–138. Springer, Heidelberg (2004)
7. Zhang, T., Oles, F.: Text Categorization Based on Regularized Linear Classification Methods. Information Retrieval 4, 5–31 (2001)
8. Monti, S., Tamayo, P., Mesirov, J., Golub, T.: Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. Machine Learning 52, 91–118 (2003)
9. Bottcher, M., Hoppner, F., Spiliopoulou, M.: On Exploiting the Power of Time in Data Mining. SIGKDD Explor. Newsl. 10, 3–11 (2008)
10. http://www.daviddlewis.com/resources/testcollections/reuters21578/
11. Khan, A., Baharudin, B., Lee, L., Khan, K.: A Review of Machine Learning Algorithms for Text-Documents Classification. Journal of Advances in Information Technology 1 (2010)
12. Joachims, T.: A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In: 14th International Conference on Machine Learning, ICML 1997, pp. 143–151 (1997)
13. Nardiello, P., Sebastiani, F., Sperduti, A.: Discretizing Continuous Attributes in AdaBoost for Text Categorization. Advances in Information Retrieval (2003)
14. Dunn, J.: Well-Separated Clusters and Optimal Fuzzy Partitions. Journal of Cybernetics 4, 95–104
15. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On Clustering Validation Techniques. Journal of Intelligent Information Systems 17, 107–145 (2001)
16. Meila, M.: Comparing clusterings–an information based distance. Journal of Multivariate Analysis 98, 873–895 (2007)

17. R Development Core Team: R: A Language and Environment for Statistical Computing., Vienna, Austria (2005)
18. Salton, G., Allan, J., Buckley, C., Singhal, A.: Automatic analysis, theme generation, and summarization of machine-readable texts. Readings in Information Retrieval, 478–483 (1997)
19. Rogati, M., Yang, Y.: High-performing feature selection for text classification. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management, pp. 659–661. ACM, McLean (2002)
20. Venables, W., Ripley, B.: Modern Applied Statistics with S, New York, USA (2002)
21. Chang, C., Lin, C.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 1–27 (2011)
22. Hornik, K., Buchta, C., Zeileis, A.: Open-source machine learning: R meets Weka. Computational Statistics 24, 225–232 (2009)
23. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. 34, 1–47 (2002)
24. Cohen, J.: A Coefficient of Agreement for Nominal Scales. Educational and Psychological Measurement 20, 37–46 (1960)
25. Iman, R., Davenport, J.: Approximations of the critical region of the Friedman statistic. Communications in Statistics 571–595 (1980)
26. Yang, Y., Zhang, J., Carbonell, J., Jin, C.: Topic-conditioned novelty detection. In: 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Canada, pp. 688–693 (2002)
27. Mendes-Moreira, J., Jorge, A.M., Soares, C., de Sousa, J.F.: Ensemble Learning: A Study on Different Variants of the Dynamic Selection Approach. In: Perner, P. (ed.) MLDM 2009. LNCS, vol. 5632, pp. 191–205. Springer, Heidelberg (2009)

A Pattern Discovery Model for Effective Text Mining

Luepol Pipanmaekaporn and Yuefeng Li

School of Electrical Engineering and Computer Science,
Queensland University of Technology, Brisbane, Australia
n7047282@student.qut.edu.au, y2.li@qut.edu.au

Abstract. The quality of extracted features is the key issue to text mining due to the large number of terms, phrases, and noise. Most existing text mining methods are based on term-based approaches which extract terms from a training set for describing relevant information. However, the quality of the extracted terms in text documents may be not high because of lot of noise in text. For many years, some researchers make use of various phrases that have more semantics than single words to improve the relevance, but many experiments do not support the effective use of phrases since they have low frequency of occurrence, and include many redundant and noise phrases. In this paper, we propose a novel pattern discovery approach for text mining. This approach first discovers closed sequential patterns in text documents for identifying the most informative contents of the documents and then utilise the identified contents to extract useful features for text mining. We develop a novel fusion method based on Dempster-Shafer's evidential reasoning which allows to combine the pieces of document to discover the knowledge (features). To evaluate the proposed approach, we adopt the feature extraction method for information filtering (IF). The experimental results conducted on Reuters Corpus Volume 1 and TREC topics confirm that the proposed approach could achieve excellent performance.

Keywords: Text Mining, Information Filtering, Pattern Summarization, Sequential Patterns

1 Introduction

As the increasing amounts of documents stored electronically, text mining (TM) has more attentions to support users for coping with the problem of information overload. TM typically involves the process of finding non-trivial and useful knowledge in a text collection. Typically, TM tasks include text categorization, text clustering, and document summarization [19]. The key challenge of TM is how to guarantee the quality of extracted knowledge (features) from text documents due to lot of noise in text.

Most existing text mining methods were developed based on term-based approaches. These approaches basically extract a set of keywords (terms) in a document to form a vector for a text representation. Weights associated with terms,

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 540–554, 2012.

such as frequency counts, are given to represent the importance of the terms in a document. A variety of machine learning (ML) techniques, such as SVMs, Rocchio, Genetic Algorithms, and Neural Network, are often applied to extract knowledge according to training documents [14]. The advantages of term-based methods include efficient computational performance and good statistic quality of terms. However, term-based methods often encounter the challenging problems such as very high dimensionality of text data and uncertain meaning of words. For many years, phrases have been used in some TM methods [12,27,15,5], as they have more discriminative and more semantics than single words. However, many experiments do not support the utilisation of phrases for text mining since they have poor statistic quality and many of them are generated without meaning [18].

In the presence of these set backs, some studies adopted data mining to discover various patterns in text [7,22,28]. Such patterns have the potential for text mining since they have predictive power, and allow to capture semantic relationships existing among terms in sentences, paragraphs, or even the whole document. Many noisy patterns could be automatically removed w.r.t. a certain frequency. Moreover, data mining has developed advanced methods for eliminating redundant patterns and noisy patterns (e.g., closed patterns [26], Maximal patterns [1], and pattern summarization [24,25]). However, a new challenge for pattern mining is *how to guarantee the quality of extracted knowledge for effective text mining.*

Motivated by the above problems, we propose a novel pattern discovery approach for effective text mining. This approach basically discovers closed sequential patterns in text documents, and then utilise them to extract the most informative contents for the documents. After thatm we extract useful features from the identified contents using a novel data fusion method based on Dempster-Shafer's evidential reasoning which allows to combine the pieces of document. The main advantages of using the fusion method include: 1) the no-requirement of a complex training process and parameter tuning to build an accurate model, 2) this model is efficient and effective for processing a high volume of documents, and 3) the model can be interpretable. We evaluate the proposed approach by implementing the feature extraction method for information filtering (IF) and compared to state-of-the-art IF models. The experimental results conducted on Reuters Corpus Volume 1 and TREC topics confirm that the proposed model could achieve the excellent filtering performance.

In summary, our contributions include

- We propose a novel feature extraction method for text mining based on pattern discovery.
- We present how to utilise the results of pattern mining to improve the efficiency and effectiveness of text mining.

2 Related Work

Trying to discover high-quality features in text has become a major concern in both information retrieval and text mining communities [18]. Some researchers in

text mining have tried to explore new kinds of features whose semantic information is rich such as [5], multi-words [12,27], and concepts [15]) since they believe that the high-level features should perform better than single words to describe the topic of document. Although some works illustrated the effective use of the high-level features for text mining, the challenging issue is that the number of the features generated is typically much higher than that of single words. This problem can lead to not only the efficiency challenges, but also the degraded performance. Some of them have tried to avoid these problems by extracting meaningful features with the help of natural language processing (NLP) techniques [15]. However, the quality of the features extracted by the NLP methods often relies on the accuracy of NLP techniques used.

From the perspective of data mining, some works adopted data mining to extract various text patterns such as itemsets and sequential patterns [11,7,22]. Such patterns have the potential for text mining since they have predictive power and meaningful context information w.r.t. frequency. Among these patterns, closed sequential patterns have the great potential for improved text categorization and filtering [7,22,21,28] due to the reasons of a reasonable number of generated patterns and a lossless compression. In [7], the experimental results demonstrated the text classifier formed by closed sequential patterns outperforms the classifier generated by frequent ones and can comparable with SVM's classifier. In [21,28], largely improvements of filtering performance was achieved by utilising closed sequential patterns discovered in text documents to extract useful terms for information filtering.

Nevertheless, the question how to deal with the large amounts of discovered patterns in text to improve both the efficiency and effectiveness of text mining remains open [28]. Although data mining has developed advanced methods for extracting a concise yet informative representation that describes the whole collection of patterns such as pattern profiles [25], compressed sets [24], and discriminative patterns [6]. [6]. However, these methods mostly focus on extracting the results to a user for further analyzing, but may remain difficult to make good use of them to improve performance of text mining. According to [21,28], the utilisation of discovered patterns in text has been limited by the problem of low frequency of occcuring patterns in text, especially large patterns that are never utilised. Unlike these approaches, we summarize the results of pattern mining to extract the useful contents of document, and then combine them to improve the efficiency and effectiveness of text mining.

3 Basic Definitions

In this section, we give the brief definitions of sequential patterns and closed sequential patterns in text. In this paper, all documents are divided into paragraphs. So, a given document d yields a set of paragraphs $PS(d)$. Let D be a collection of documents, including a set of positive (relevant) documents, D^+, and a set of negative (irrelevant) ones, D^-.

Let $T = \{t_1, t_2, \ldots, t_m\}$ be a set of terms which are extracted from D^+. Given X be an ordered list of terms, or a *sequence of terms*, i.e., $X =< t_1, \ldots, t_r >$

($t_i \in T$) in document d, $coverset(X)$ denotes the covering set of X for d, i.e., $coverset(X) = \{d_p | d_p \in PS(d), X \subseteq d_p\}$, where d_p denotes a paragraph of document. Based on this, the following definitions are given.

Definition 1 (Absolute and Relative Supports). *The* absolute *support of X is the number of occurrences of X in $PS(d)$: $sup_a(X) = |coverset(X)|$. The* relative *support of X is the fraction of the paragraphs that contain the sequence* : $sup_r(X) = \frac{|coverset(X)|}{|PS(d)|}$.

Definition 2 (Sequential patterns). *A sequence of terms X is called* sequential pattern *if its sup_a (or sup_r) $\geq min\_sup$, a minimum support.*

Definition 3 (Closed patterns). *We say that a pattern p is* closed *if there is no any super-pattern q of p such that $sup_a(p) = sup_a(q)$.*

Paragraph	*Terms*
dp_1	t_1 t_2
dp_2	t_3 t_4 t_6
dp_3	t_3 t_4 t_5 t_6
dp_4	t_3 t_4 t_5 t_6
dp_5	t_1 t_2 t_6 t_7
dp_6	t_1 t_2 t_6 t_7

(a) A set of paragraphs

Frequent Pattern	*Covering Set*
$\{\mathbf{t_3}, \mathbf{t_4}, \mathbf{t_6}\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_3, t_4\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_3, t_6\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_4, t_6\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_3\}$	$\{dp_2, dp_3, dp_4\}$
$\{t_4\}$	$\{dp_2, dp_3, dp_4\}$
$\{\mathbf{t_1}, \mathbf{t_2}\}$	$\{dp_1, dp_5, dp_6\}$
$\{t_1\}$	$\{dp_1, dp_5, dp_6\}$
$\{t_2\}$	$\{dp_1, dp_5, dp_6\}$
$\{\mathbf{t_6}\}$	$\{dp_2, dp_3, dp_4, dp_5, dp_6\}$

(b) Sequential patterns and covering sets

Fig. 1. An example of sequential and closed patterns

Figure 1(a) lists six paragraphs for a given document d, where $PS(d) = \{dp_1, dp_2, \ldots, dp_6\}$, and duplicate terms are removed. Assume $min\_sup = 3$ (Sup_a), ten sequential patterns would be extracted as shown in Table 1(b). As shown in Figure 1(b), patterns $\{t_3, t_4, t_6\}$ and $\{t_6\}$ are *closed*; whereas the remaining ones are *non-closed*, and could be removed.

To improve the efficiency, we use an algorithm $SPMining(D^+, min\_sup)$ developed in our previous work [22] (also used in [21,28]). The $SPMining$ algorithm finds all closed sequential patterns (hereafter *patterns*) in paragraphs of a document that have a frequency above a minimum frequency constraint min_sup (We do not repeat this algorithm here because of the length limitation of the paper).

For all positive documents $d_i \in D^+$, the *SPMining* algorithm generates a set of patterns as the following vector:

$$\overrightarrow{d_i} = \langle (p_{i_1}, f_{i_1}), (p_{i_2}, f_{i_2}), \ldots, (p_{i_m}, f_{i_m}) \rangle \quad (1)$$

where p_{i_j} in pair (p_{i_j}, f_{i_j}) denotes a pattern p_j in document d_i associated with its frequency count f_{i_j} (i.e., Sup_a). The result of applying this algorithm to all

positive documents is to yield a collection of n vectors, which can be expressed as follows:

$$\eta = \left\{\overrightarrow{d_1}, \overrightarrow{d_2}, \ldots, \overrightarrow{d_n}\right\} \tag{2}$$

where $n = |D^+|$.

4 Mining Informative Contents

In general, a text contains pieces of elements such as terms, phrases, sentences, or even paragraphs involving spans of text on different semantic levels. When a text was mined, patterns discovered in the text capture interesting relationships existing among terms at a defined level of granularity. For example, given a text is segemented into paragraphs, all patterns discovered in the text can be interpreted as pieces of knowledge that represent frequencies of some terms containing within the paragraphs. Thus, we utilise the discovered patterns in the text to extract the most informative content for describing the topic of document. Consequently, we define the summarization problem as follows:

Definition 4 (Pattern Summarization). *Given a set of patterns* $P = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ *that are mined from a text* T *contains a set of* m *text elements* $T = \{\phi_1, \phi_2, \ldots, \phi_m\}$ *where* $m < n$. *Pattern summarization is to extract the* K *most useful elements of text that contain the patterns where* $K \leq m$.

Thus, the problem of using patterns in text is transformed into the problem of extracting the K most informative set of text elements. In this paper, we adopted a Maximal Marginal Relevance based feature seslection method, called Maximal Marginal significance (MMS), proposed in [23]. Basically, the MMS method selects features through a gain function g defined as $g(\alpha) = S(\alpha) - \underset{\beta \in F_S}{Max}\, R(\alpha, \beta)$, where $S(\alpha)$ equals a function assigning a significant score to the pattern α, $R(\alpha, \beta)$ denotes a function that computes the redundancy between the two patterns α and β, and F_S is a set of features already selected. According to the gain function, a pattern is selected if it is highly significant and contains very low redundancy to the features already selected. However, the definitions of the functions S and R needs to be specified according to a given domain.

Let d_i be a positive document that contains m text elements, i.e., $d_i = \{\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_m\}$, we define the significance score to measure the importance of each element according to the following function:

$$S_i(\phi_j) = \frac{\sum_{\alpha_k \subseteq \phi_j} w(\alpha_k) \times Sup_a(\alpha_k)}{|\{\alpha_k | \alpha_k \subseteq \phi_j\}|} \tag{3}$$

where ϕ_j be an element in the document d_i, $Sup_a(\alpha_k)$ indicates the support (frequency) of the pattern α_k in the document, and $w(\alpha_k)$ returns the weight of the pattern α_k associated with a given measure. In this work, we adopted a weighting algorithm proposed in [22] to determine the weights of the patterns as follows:

$$w(\alpha_k) = \frac{|\{d_a | d_a \in D^+, \alpha_k \in d_a\}|}{|\{d_b | d_b \in D, \alpha_k \in d_b\}|} \tag{4}$$

According to Eq.(3), the weight of the text element ϕ_j can be determined by summarizing the weights of the patterns contained by it. The denominator of this function is to normalise long elements.

To maximize the information of text, the selected elements should overall provide very low redundancy each other. We calculate the semantic similarity between two segments ϕ_k and ϕ_j by using the Jaccard's coefficient [20] (also used in [23]), and then define the part of redundancy measurement as follows:

$$R_i(\phi_j, \phi_k) = \frac{|\phi_k \cap \phi_j|}{|\phi_k \cup \phi_j|} \times \min(S_i(\phi_k), S_i(\phi_j)) \tag{5}$$

where $|\phi_k \cap \phi_j|$ returns the intersection of patterns contained in both the segments ϕ_k and ϕ_j respectively.

Once the gain function was defined, we perform a greedy search method as follows. For all positive documents $d_i \in D^+$ that consists of a set of n text segments, we start an empty set for F_S. In each iteration, we expand the set F_S with the segment α that maximizes the gain function g with the remaining segments in this document. We keep on expanding F_S till the number of selected segments equals k or no segments remain.

It is possible that the selected elements may contain many extraneous (noisy) terms. For example, considering paragraphs dp_5 and dp_6 in Figure 1(a) contains term t_7 that are infrequent. It is necessary to remove the noisy terms contained in these elements when they are selected for profiling the document. Let ϕ_j be a text segment containing patterns $\alpha_1, \alpha_2, \ldots, \alpha_\ell$. We perform to remove infrequent terms contained in each element selected.

5 A Pattern Fusion Model

In this section, we present a pattern fusion model for using the patterns extracted from text documents. This model is developed based on evidential reasoning which allows to formally combine knowledge to make inferences.

5.1 Transferable Belief Model (TBM)

Although many evidential reasoning techniques have been developed for applications, such as a combination of classifiers [2], the method proposed in this paper is to adopt Transferable Belief Model (TBM) [16], a statistical technique for formally representing and combining knowledge based on Dempster-Shafer theory. In comparison with other reasoning methods, the TBM allows to represent uncertainty (probability) about related events to perform reasoning, and does not require much training data and prior information like Bayesian technique [16]. In TBM, a two-level structure is present: the *credal* level where beliefs are entertained and the *pignistic* level where beliefs are used to make decisions. When a decision must be made, beliefs at the credal level are transformed into beliefs at the pignistic level. Figure 1 describes the process of pattern fusion for information filtering. According to Figure 1, a user profile that represents user interest

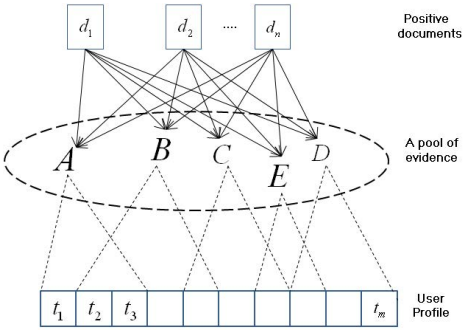


Fig. 2. The pattern fusion model for information filtering

consists of a set of m terms extracted from positive documents D^+. Each positive document represented by a list of k patterns can be viewed as an expert that states their opinions (weights) to specific propositions (i.e., termsets) in a pool of evidence according to their matching in the expert document. The weights assigned by these documents are pooled, and then are encoded into supports (weights) of each term in the user profile.

5.2 Evidential Mapping

We start to extract a set of m terms, denoted as Ω, from positive documents for profile representation. To reduce the dimensionality of text documents, we extract the set of terms from all the specific patterns extracted in positive documents, i.e., $\Omega = \{t_1, t_2, \ldots, t_m\}$. After that, we define a support (mass) function $m : 2^{\Omega} \rightarrow [0, 1]$ that assigns weights to some propositions in the space (i.e., 2^{Ω}) associated with a document.

For all positive documents $d_i \in D^+$, the mass function m_i can be defined as follows:

$$m_i(A) = \begin{cases} 0 & if\ A = \emptyset; \\ \frac{S_i(\{\lambda_j | \lambda_j \subseteq d_i, \lambda_j = A\})}{\sum_{B \subseteq \Omega} S_i(\{\lambda_r | \lambda_r \subseteq d_i, \lambda_r = B\})} & , otherwise \end{cases} \tag{6}$$

where $A, B \in 2^{\Omega}$, $m_i(A)$ denotes a mass function that assigns a weight to proposition A associated with the pattern λ_j extracted in the document d_i.

5.3 Weight Fusion

A challenging problem of data fusion methods is how to handle conflicts from multiple sources of data. Dempster's rule of combination is often used to combine evidence supported by two mass functions defined on a common space. Given m_i and m_j be two mass functions associated with sources i and j respectively, this rule is given as follows:

$$m_{i \oplus j}(Z) = \frac{1}{\kappa} \times \sum_{X \cap Y = Z} m_i(X) \times m_j(Y) \tag{7}$$

where $X, Y \in 2^{\Omega}$ and $\kappa = \sum_{X \cap Y \neq \emptyset} m_i(X) \times m_j(Y)$. This rule computes a measure of agreement between two bodies of evidence concerning propositions discerned from a common space. It is a commutative and associative operation that focuses only on propositions supported by both bodies of evidences. This numerator is the sum over all conjunctions that support a proposition. The denominator is a normalisation factor that ensures the mass function constraint.

For situations when belief functions are fused as an accumulate function of the evidence, the Dempster's combination constraint often produces counter-intuitive results [10]. The objective of the proposed approach is to combine belief functions for induction that will necessarily accumulate discovered knowledge in text documents.

Let m_i and m_j be two support functions associated with documents d_i and d_j respectively. Assume that m_i contributes weights to a body of evidence: $(X_1, X_2, \ldots, X_l)$ and m_j contributes weights to a body of evidence: $(Y_1, Y_2, \ldots, Y_n)$. The following rule of combination is used to fuse the two mass functions to a new one.

$$m_{i \oplus j}(Z) = \frac{1}{2} \begin{cases} m_i(X_s) + m_j(Y_r) \; if \; X_s = Z, Y_r = Z \\ m_i(X_s) \;\; if \; X_s = Z, \forall r : Y_r \neq Z \\ m_j(Y_r) \;\; if \; Y_r = Z, \forall s : X_s \neq Z \end{cases} \tag{8}$$

where $m_i \oplus w_j(Z)$ be the combined mass function that assigns a weight to proposition $Z \in 2^{\Omega}$ The combination rule is also cumulative, commutative, and associative. We use this rule to combine all mass functions associated with positive documents to obtain a new function m_{D^+}, i.e.,

$$m_{D^+}(Z) = (((m_1 \oplus m_2) \oplus m_3) \otimes \ldots \otimes m_{|D^+|}(Z)) \tag{9}$$

for all propositions $Z \in 2^{\Omega}$.

5.4 Reasoning

Once the weights were pooled, we use the belief knowledge to accurately evaluate weights to each element t_i in the user profile Ω according to the following function:

$$w_i^{\Omega} = \sum_{t_i \in \Omega, t_i \in Z} \frac{m_{D^+}(Z)}{|Z|} \tag{10}$$

where $\forall A \in \Omega$ and $m_{D^+}(Z)$ be a function obtained by Eq. (8), and $|Z|$ equals the cardinality of proposition Z in the space Ω. It is easy to examine the term weighting function is a *pignistic probability* function in TBM [16].

In order to utilise the profile Ω for document filtering, a document evaluation function is built as follows:

$$rel(d) = \sum\nolimits_{j \in d} w_j^{\Omega} \tag{11}$$

where $rel(d)$ returns a score assigned to the document d, w_j^{Ω} is the weight of term j in the profile Ω. A high value assigned to the document can imply that the document tends to be highly *relevant* to the user.

6 Experimental Evaluation

6.1 Experimental Dataset

Reuters Corpus Volume 1 (RCV1) data collection [13] is used to test the proposed model, namely the *Pattern Fusion* model (PFM). This dataset consists of all and only English language stories proposed by Reuter's journalists between 1996-08-20 to 1997-08-19, a total of 806,791 documents that cover very large topics and information. TREC (2002) has developed and provided 50 assessor topics [17] for the filtering track, aiming to building a robust filtering system. These topics were developed by human assessors of the National Institute of Standards and Technology (NIST), called *assessor* topics. According to [4], 50 topics are stable and sufficient for conducting high quality experiments. This research hence uses RCV1 and the 50 assessor topics to evaluate the proposed model.

6.2 Data Preprocessing and Measures

For each assessor topic, its data collection is split into two sets: a training set and a test set. All documents are marked in XML and some meta-data information. In order to avoid bias in experiments, we remove all meta-data information and perform a common basic text processing for all documents, including stop-words removal according to a given stop-words list and stemming terms.

The effectiveness is measured by five different means: The precision of the top 20 returned documents ($top-20$), F_1 measure, Mean Average Precision (MAP), the break-even point (b/p), and Interpolated Average Precision (IAP) on 11−points. Precision (p), Recall (r), and F_1 are calculated by the following functions:

$$p = \frac{TP}{TP + FP}, r = \frac{TP}{TP + FN}, F_1 = \frac{2 * p * r}{p + r}$$

where TP is the number of documents the system correctly identifies as positives; FP is the number of documents the system falsely identifies as positives; FN is the number of relevant documents the system fails to identify.

6.3 Baseline Models and Settings

We grouped baseline models into two main categories. The first category includes a number of data mining (DM) based models for IF:

– **PTM** [22]: This method efficiently extracts closed sequential patterns from relevant training documents for document filtering. Weights assigned to each pattern are used to represent the relevance.

- **PDM** [21]: This work proposed a novel term weight method for information filtering (IF) using data mining, called a *pattern deploying* method (PDM). Instead of normal term evaluations, each term in a training document is evaluated based on their appearance in all closed sequential patterns extracted in the document. The experimental results in [21] showed that PDM can largely improve filtering performance as compared with PTM and state-of-the-art IF models such as Rocchio and Pr.
- **IPE** [28]: This work proposed to improve the effectiveness of PDM [21]. A method for revising weights of features (terms) in positive documents was developed to reduce the effects of noisy terms with the help of non-redundant documents.

For data mining models, the minimum support threshold (min_sup) is an important parameter and is sensitive to a given data set. We set this constraint to 0.2, which means 20% of the number of paragraphs in a document for all the models since it was recommended as the best value for this data collection [22,21,28].

The second category includes the two state-of-the-art term-based relevance feedback methods for IF:

- **Rocchio** [8]: This method generates a Centroid for representing user profiles by extracting terms from positive documents and performing to revise weights of the terms with negative documents. The centroid $\boldsymbol{c}$ of a topic can be generated as follows:

$$\alpha \frac{1}{|D^+|} \sum_{\overrightarrow{d} \in D^+} \frac{\overrightarrow{d}}{||\overrightarrow{d}||} - \beta \frac{1}{|D^-|} \sum_{\overrightarrow{d} \in D^-} \frac{\overrightarrow{d}}{||\overrightarrow{d}||} \tag{12}$$

  where $||\overrightarrow{d}||$ be normalized vector for document d. α and β be a control parameter for the effect of relevant and non-relevant data respectively. According to [8,3], there are two recommendations for setting the two parameters: $\alpha = 16$ and $\beta = 4$; and $\alpha = \beta = 1.0$. We have tested both accommodations on assessor topics and found the latter recommendation was the best one. Therefore, we let $\alpha = \beta = 1.0$.
- **Support Vector Machine (SVM)** : Several researchers have shown the linear SVM is one of the most effective text classifiers [9,14]. We would compare it with the proposed model. However, the SVM here is used to rank documents rather than to make a binary decision, and it only uses terms based features extracted from training documents. We describe the details as following: Given a linear function $h(x) =< w.x > +b$ where $h(x) = +1$ if $< w.x > +b \geq 0$; otherwise $h(x) = -1$. where x is the input vector; $b \in \mathbb{R}$ is the bias and $< w.x >$ is the dot product of w and x. $w = \sum_{i=1}^{l} y_i \alpha_i x_i$ for the given training data:$(x_i, y_i), \ldots, (x_l, y_l)$, where $x_i \in \mathbb{R}^n$ and $y_i = +1(-1)$, if document x_i is labelled positive (negative). $\alpha_i \in \mathbb{R}$ is the weight of the sample x_i and satisfies the constraint:

$$\forall_i : \alpha_i \geq 0 \; and \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{13}$$

For the purpose of ranking, b can be ignored. For the documents in a training set, we know only what are positive (negative), but not which one is important. We assign the same α_i value (i.e., 1) to each positive document first, and then determine the same α_i (i.e., α') value to each negative document based on the Eq. (8). Therefore, a testing documents d is scored by the function $r(d) = w.d$ where . means *inner products*; d is the term vector of the testing document; and

$$w = \left(\sum_{d_i \in D^+} d_i \right) + \left(\sum_{d_j \in D^-} d_j \alpha' \right)$$

For each topic, we also choose 150 terms in the positive documents, based on tf $\times$ idf values for all ML-based models since it is the best average value for all the assessor topics.

6.4 Quality of Extracted Features

Here we aim to determine the best quality of extracted features with respect to top$-k$ elements extracted from training documents. Figure 3 illustrates the results of varying top$-k$ percentages of extracted features in each document associated with Mean Average Precision (MAP) on all assessor topics. According to Figure 3, the best MAP performance on the assessor topics was achieved using just top-40 percentages of extracted features in a document for constructing the user profile while selecting more features tended to reduce the performance. This is since more extraneous (noisy) terms may be included.

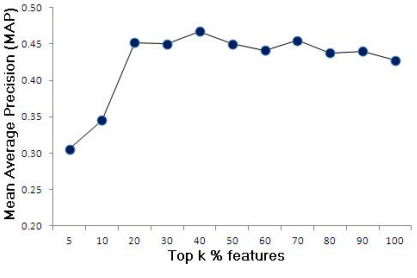


Fig. 3. Mean Average Precision (MAP) w.r.t. top$-k$ percentages of selected features

6.5 PFM vs. Pattern Mining Models

The results of overall comparisons between the proposed model and all data mining models have shown in Table 1. The most important findings revealed in this table are that both PDM and IPE models largely outperforms PTM that uses directly patterns over all the standard measures with the slight increase in IPE as compared to PDM. The results support the effective use of patterns in text to extract useful terms in text documents.

Table 1. Comparison results of PFM with all DM-based methods on all assessor topics

Model	*top-20*	*MAP*	*b/p*	$F_{\beta=1}$
PFM	**0.545**	**0.473**	**0.457**	**0.458**
IPE [28]	0.493	0.441	0.429	0.440
PDM [21]	0.496	0.444	0.430	0.439
PTM [22]	0.406	0.364	0.353	0.390
%Chg	+10.54%	+7.26%	+6.53%	+4.10%

Table 2. Comparison results of PFM with all ML-based methods on all assessor topics

Model	*top-20*	*MAP*	*b/p*	$F_{\beta=1}$	*Avgt*
PFM	**0.545**	**0.473**	**0.457**	**0.458**	53.32
SVM	0.447	0.408	0.409	0.421	150.0
Rocchio [8]	0.416	0.391	0.392	0.408	150.0
%Chg	+21.92%	+15.91%	+11.73%	+8.78%	−64.45%

Table 3. $p-$values for the representative models comparing with PFM on all the assessor topics

Model	*top-20*	*MAP*	*b/p*	$F_{\beta=1}$
IPE	0.0078	0.0096	0.0165	0.0283
PDM	0.0100	0.0025	0.0023	0.0026
SVM	0.0002	0.0064	0.0042	0.0122

We also compare PFM with IPE. As seen in Table 1, PFM perform better than IPE with +7.10% (max +10.54% on $top-20$ and min +4.10% on $F_{\beta=1}$) in percentage change on average over the standard measures. The encouraging improvements of PFM is also consistent and significant on 11−points as shown in Figure 4. These results support the highlights of effectively handling the discovered patterns in text.

6.6 PFM vs. Term-Based Models

As shown in Table 2 and Figure 5, both Rocchio and SVM models that are based on keyword-based models perform over PTM, where SVM performs better than Rocchio over all the measures. This illustrates keywords remain the very effective concept for text mining since they have good statistic quality. However, the results compared between the term-based models and the remaining data mining models (i.e., IPE, PDM, and PFM) confirm that patterns are much more effective to accurately evaluate weights of terms in text documents than normal statistical evaluations that rely on raw data. In comparisons with SVM, the excellent performance was achieved by PFM with +14.58% increasing in average (max +21.92% on $top-20$ and min +8.78% on $F_{\beta=1}$). Furthermore, the average number of terms used in the user profile extracted by PFM is much less than that of both the term-based models with −64.45%.

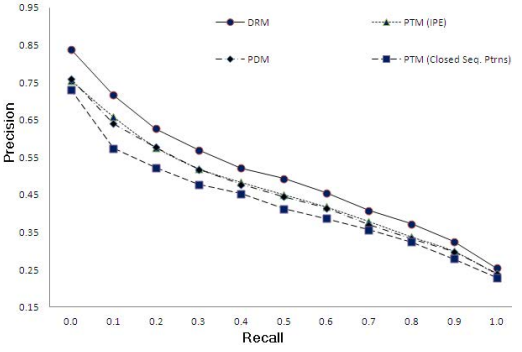


Fig. 4. Comparison performance results with data mining based models

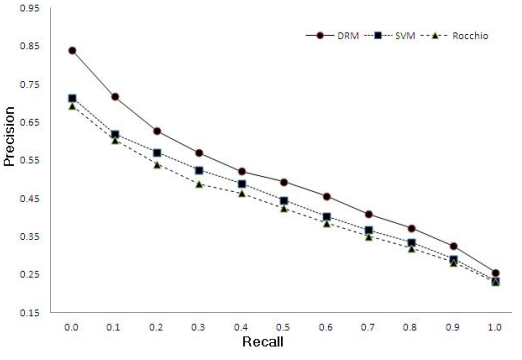


Fig. 5. Comparison performance results with term-based models

7 Conclusion

In the present paper, we solved the challenging problem of dealing with sequential patterns discovered in text documents for effective text mining by proposing a novel pattern discovery method for extracting useful features in text. This method discovers closed sequential patterns from documents, and then summarizes them to identify the most informative contents for the documents. We also present a novel fusion method based on evidential reasoning for dealing with the extracted contents to discover useful features for text mining. We promote the proposed approach by implementing a novel information filtering (IF) model. The experimental results conducted on Reuters Corpus Volume 1 data collection and TREC topics support that the IF model could significantly achieve the best performance of filtering as compared to state-of-the-art IF models. We believe that the proposed approach is very promising for effective text mining.

References

1. Bayardo Jr., R.: Efficiently mining long patterns from databases. ACM Sigmod Record 27, 85–93 (1998)
2. Bi, Y., Wu, S., Wang, H., Guo, G.: Combination of evidence-based classifiers for text categorization. In: 2011 23rd IEEE International Conference on Tools with Artificial Intelligence, pp. 422–429. IEEE (2011)
3. Buckley, C., Salton, G., Allan, J.: The effect of adding relevance information in a relevance feedback environment. In: ACM SIGIR 17th International Conf., pp. 292–300 (1994)
4. Buckley, C., Voorhees, E.: Evaluating evaluation measure stability. In: 23th ACM SIGIR International Conf. on Research and Development in Information Retrieval, pp. 33–40 (2000)
5. Caropreso, M., Matwin, S., Sebastiani, F.: Statistical phrases in automated text categorization. Centre National de la Recherche Scientifique, Paris, France (2000)
6. Cheng, H., Yan, X., Han, J., Hsu, C.: Discriminative frequent pattern analysis for effective classification. In: 23rd IEEE ICDE International Conf. on Data Engineering, pp. 716–725 (2007)
7. Jaillet, S., Laurent, A., Teisseire, M.: Sequential patterns for text categorization. Intelligent Data Analysis 10(3), 199–214 (2006)
8. Joachims, T.: A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In: 14th ICML International Conf. on Machine Learning, pp. 143–151 (1997)
9. 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)
10. Lefevre, E., Colot, O., Vannoorenberghe, P.: Belief function combination and conflict management. Information Fusion 3(2), 149–162 (2002)
11. Malik, H., Kender, J.: High quality, efficient hierarchical document clustering using closed interesting itemsets. In: 6th IEEE ICDM International Conf. on Data Mining, pp. 991–996 (2006)
12. Nanas, N., Vavalis, M.: A "Bag" or a "Window" of Words for Information Filtering? In: Darzentas, J., Vouros, G.A., Vosinakis, S., Arnellos, A. (eds.) SETN 2008. LNCS (LNAI), vol. 5138, pp. 182–193. Springer, Heidelberg (2008)
13. Rose, T., Stevenson, M., Whitehead, M.: The reuters corpus volume 1-from yesterday's news to tomorrow's language resources. In: 3th International Conf. on Language Resources and Evaluation, pp. 29–31 (2002)
14. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
15. Shehata, S., Karray, F., Kamel, M.: A concept-based model for enhancing text categorization. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 629–637. ACM (2007)
16. Smets, P.: Data fusion in the transferable belief model. In: Proceedings of the Third International Conference on Information Fusion, FUSION 2000, vol. 1, pp. PS21–PS33. IEEE (2000)
17. Soboroff, I., Robertson, S.: Building a filtering test collection for trec 2002. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, p. 250. ACM (2003)
18. Stavrianou, A., Andritsos, P., Nicoloyannis, N.: Overview and semantic issues of text mining. ACM SIGMOD Record 36(3), 23–34 (2007)

19. Tan, A.: Text mining: The state of the art and the challenges. In: Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases, pp. 65–70 (1999)
20. van der Weide, T., van Bommel, P.: Measuring the incremental information value of documents. Information Sciences 176(2), 91–119 (2006)
21. Wu, S., Li, Y., Xu, Y.: Deploying approaches for pattern refinement in text mining. In: 6th IEEE ICDM International Conf. on Data Mining, pp. 1157–1161 (2006)
22. Wu, S., Li, Y., Xu, Y., Pham, B., Chen, P.: Automatic pattern-taxonomy extraction for web mining. In: 3th IEEE/WIC/ACM WI International Conf. on Web Intelligence, pp. 242–248 (2004)
23. Xin, D., Cheng, H., Yan, X., Han, J.: Extracting redundancy-aware top-k patterns. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 444–453. ACM (2006)
24. Xin, D., Han, J., Yan, X., Cheng, H.: Mining compressed frequent-pattern sets. In: Proceedings of the 31st International Conference on Very Large Databases, pp. 709–720. VLDB Endowment (2005)
25. Yan, X., Cheng, H., Han, J., Xin, D.: Summarizing itemset patterns: a profile-based approach. In: 11th ACM SIGKDD International Conf. on Knowledge Discovery in Data Mining, pp. 314–323 (2005)
26. Zaki, M.: Generating non-redundant association rules. In: 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 34–43 (2000)
27. Zhang, W., Yoshida, T., Tang, X.: Text classification using multi-word features. In: IEEE International Conference on Systems, Man and Cybernetics, ISIC 2007, pp. 3519–3524. IEEE (2007)
28. Zhong, N., Li, Y., Wu, S.: Effective pattern discovery for text mining. IEEE Transactions on Knowledge and Data Engineering, doi: http://doi.ieeecomputersociety.org/10.1109/TKDE,211

Investigating Usage of Text Segmentation and Inter-passage Similarities to Improve Text Document Clustering

Shashank Paliwal and Vikram Pudi

Center for Data Engineering, International Institute of Information Technology, Hyderabad
paliwal@students.iiit.ac.in, vikram@iiit.ac.in

Abstract. Measuring inter-document similarity is one of the most essential steps in text document clustering. Traditional methods rely on representing text documents using the simple Bag-of-Words (BOW) model. A document is an organized structure consisting of various text segments or passages. Such single term analysis of the text treats whole document as a single semantic unit and thus, ignores other semantic units like sentences, passages etc. In this paper, we attempt to take advantage of underlying subtopic structure of text documents and investigate whether clustering of text documents can be improved if text segments of two documents are utilized, while calculating similarity between them. We concentrate on examining effects of combining suggested inter-document similarities (based on inter-passage similarities) with traditional inter-document similarities following a simple approach for the same. Experimental results on standard data sets suggest improvement in clustering of text documents.

Keywords: Text Document Clustering, Text Segmentation, Document Similarity.

1 Introduction

With a large explosion in the amount of data found on the web, it has become necessary to devise better methods to classify data. A large part of this web data (like blogs, webpages, tweets etc.) is in the form of text. Text document clustering techniques play an important role in the performance of information retrieval, search engines and text mining systems by classifying text documents. The traditional clustering techniques fail to provide satisfactory results for text documents, primarily due to the fact that text data is very high dimensional and contains a large number of unique terms in a single document. Most of these documents do not particularly deal with a single topic, which makes it difficult to classify them under a single category. Such a scenario, thus gives rise to need for clustering methods which can classify documents on the basis of topic on which the document is primarily written i.e. theme of most of the passages or segments which combine together to form the whole document.

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 555–565, 2012.

Text documents are often represented as a vector where each term is associated with a weight. The Vector Space Model [13] is a popular method that abstracts each document as a vector with weighted terms acting as features. Most of the term extraction algorithms follow "Bag of Words" (BOW) representation to identify document terms. While such a representation is simple and easy to understand, it suffers from two problems. One, it relies heavily on vocabulary used by the author to calculate similarity between two documents. A pair of documents might be on similar topics, but still score very low on similarity value because of different set of terms being used in two documents. Two, it considers whole document as a single semantic unit. Two documents may talk about related topics and share common vocabulary, but they could still be judged dissimilar because of other unrelated topics present in the two documents, if one or both of the documents consist of varying topics. While the first problem can be tackled using dictionaries or a "wordnet" like lexical database [7], applying clustering algorithm to semantically independent units of text might help in reducing the defiling effect of drifting topics (as text segments on similar topics would be judged similar while those on unrelated topics as dissimilar) and varying length problem (as text segments are going to be of same fixed size). It is our intuition that calculating document-document similarity with the help of text segments of a particular length may help in improving quality of clustering by solving varying length problem and drifting topics problem to a small extent.

In this paper, our primary aim is to investigate whether segmenting a document into various independent units could help in improving the clustering of text documents or not. So, we present a simple algorithm to efficiently calculate inter-passage similarities between text segments of two different documents and then effectively integrate these values with those obtained from considering each document as a single semantic unit, to obtain better clustering of text documents. Throughout this paper, we use text segments or text windows interchangeably and assume them to be same i.e. a segment of document consisting of a particular number of words which we refer to as "**Window Size**".

The rest of the paper is organized as follows. Section 2 briefly describes the related work. Section 3 explains the process of term segmentation in a text document and motivation behind this paper. Section 4 describes our approach to the calculation of similarity between two documents. Section 5 and 6 describe experimental results and the conclusion respectively.

2 Related Work

Many Vector Space Document based clustering models make use of single term analysis only. To further improve clustering of documents and rather than treating a document as a bag of words, considering term dependency while calculating document similarity has gained attention.[8, 14]

Passage retrieval is the task of retrieving only those segments of text which are relevant to a particular information need. It has been extensively utilized in the field of information retrieval to improve the quality of retrieval [2, 3] and improve

performance of question answering systems [1]. [6] utilizes segmentation of web pages to improve the quality of web search.

In [4], fragments of legal text documents are clustered. However, no segmentation algorithm is needed as legal documents are decomposable. [12] proposes passage-based text categorization model, which segments a document and then passage categories are merged into document categories to achieve final categorization of documents. Perhaps, the more closely related works are [5] and [11]. In [5], authors evaluate the impact of text segmentation on query specific clustering of text documents. [11] focuses on clustering of multi-topic documents using text segments. Our work is different from [11] in two aspects majorly. First, our focus is not on multi topic documents and second, we attempt to investigate effects on hard clustering, if similarity between text segments is also included in combined similarity between two documents, while [11] attempts to improve soft clustering of multi-topic documents utilizing each text segment as an independent semantic unit.

3 Basic Idea

The basis of this work is the intuition that two documents should be considered more similar for the purpose of clustering, if the set of common terms between the two documents are contained in a small region as compared to two other documents in which these terms are highly scattered across the documents. Traditional vector space model based techniques ignore the density of region in which these common terms fall and thus judge many similar (dissimilar) documents as dissimilar (similar).

3.1 Text Segmentation

Text segments can be categorized into three kinds of passages: discourse, semantic, and window. Discourse passages rely on the logical structure of the documents marked by punctuation. Semantic passages are obtained by partitioning a document into topics or sub-topics according to its semantic structure (e.g. TextTiling [10]). The third type of passages which are fixed-length passages or windows, are defined to contain a fixed number of words and were introduced in [9].

For the sake of simplicity, we use the fixed length passages in our experiments. We use both **non-overlapping** and **overlapping passages** to investigate effect of combining inter-document and inter-passage similarities on text document clustering.

Example: Document = "The flash washes out the photos, and the camera takes very long to turn on." Window Size = 4

1. **Non-Overlapping Passages** are following

Passage 1: "The flash washes out"	**Passage 2**: "the photos and the"
Passage 3: "camera takes very long"	**Passage 4**: "to turn on"

2. **Overlapping Passages** with size of overlap = (Window size / 2) are following

Passage 1: "The flash washes out"	**Passage 2**: "washes out the photos"	**Passage 3**: "the photos and the"
Passage 4: "and the camera takes"	**Passage 5**: "camera takes very long"	**Passage 6**: "very long to turn"
Passage 7: "to turn on"		

4 Similarity Computation

Let D be a document set with N number of documents:

$$D = \{d_1, d_2, d_3 \ldots\ldots d_N\}$$

Where $d_n = \{ t_1, t_2, t_3 \ldots\ldots t_m \}$ and d_n is the n^{th} document in corpus and t_i is i^{th} term in document d_n.

4.1 Traditional Inter-document Similarity

We calculate inter-document similarity by calculating cosine similarity between two document vectors with each feature weighted using *tf-idf* method.

$$\text{Tf-idf weight} : \log\left(1 + tf_{(t,d)}\right) * \log\left(1 + {}^{N}/_{x_t}\right) \quad (1)$$

where $tf_{(t,d)}$ is term frequency of term t in document d and N is the total number of documents in corpus and x_t is the number of documents in which term t occurs.

Cosine similarity between two document vectors $\overrightarrow{d_1}$ and $\overrightarrow{d_2}$ is calculated as,

$$Sim_d(d_1, d_2) = \frac{\overrightarrow{d_1}.\overrightarrow{d_2}}{|\overrightarrow{d_1}||\overrightarrow{d_2}|} \quad (2)$$

4.2 Passage-Based Inter-document Similarity

For a document d consisting of m terms and assuming window size of w, document d will be segmented into

1. k windows for non-overlapping text windows
   where $k = \left[\frac{m}{w}\right]$ $if\,(m \,\%\, w = 0)$ and $\left(\left[\frac{m}{w}\right] + 1\right) if\ (m \,\%\, w > 0)$.

2. k windows for overlapping text windows with size of overlap equal to $({}^{w}/_{2})$
   where $k = \left(\left[\frac{m}{w}\right] - 1\right) if\ (m \,\%\, w = 0)\ and\ \left[\frac{m}{w}\right]\ if\ (m \,\%\, w > 0)$.

A window or passage too is represented using a feature vector with terms present in the passage being its features and tf-idf weighting scheme used to weigh these features. However, for weighting terms of passages, each passage is considered as a

single document and all the passages of a single document together are treated as the full corpus.

Let d_1 consists of $\{ P_1, P_2 \ldots\ldots P_r \}$ and d_2 of $\{ P_1', P_2' \ldots\ldots P_s' \}$, and assuming r < s, then passage-based inter-document similarity for d_1 and d_2 is:

$$Sim_p(\overrightarrow{d_1}, \overrightarrow{d_2}) = \frac{\sum_{i=1}^{r} \max\,(Sim(P_i, P'j))}{r} \quad (3)$$

Where, j varies from 1 to s and inter-passage similarity $Sim(P_i, P_j')$ is cosine similarity between feature vectors of two passages.

4.3 Combined Similarity Measure

Let traditional inter-document similarity for documents d_1 and d_2 be represented as $Sim_d(d_1,d_2)$ and suggestedpassage-based inter-document similarity as $Sim_p(d_1,d_2)$, then combined or effective similarity between d_1 and d_2 is :

$$Sim(d_1,d_2) = \alpha * Sim_p(d_1,d_2) + (1-\alpha) * Sim_d(d_1,d_2) \quad (4)$$

Where α is similarity blend factor [8] and $0 \leq \alpha \leq 1$.

5 Experimental Results

We conducted experiments to investigate the effectiveness of our method i.e. using both inter-document and inter-passage similarities together in improving text document clustering. The experiments were conducted for two types of fixed-length passages i.e. **overlapping** and **non-overlapping.** It is important to note we do not apply any kind of dimensionality reduction on original document vector which consists of only single term features since our aim is to investigate whether inter-passage similarities can be successfully utilized to improve clustering or not. In other words, we want to credit any improvement or deterioration in clustering to the suggested similarity measure.

5.1 Data Sets

We used two data sets, out of which one is a web document data set[1], manually collected and labeled from Canadian websites and second is a collection of articles posted on various USENET newsgroups. It is a subset of full 20-newsgroup dataset. It is available from the UCI KDD archive[2]. While web data set has moderate overlap between different classes, mini 20-newsgroup data set has varying overlap between different classes. Average length of a document in UW-Can data set is much greater than that of a document from mini 20-newsgroup dataset.

[1] Link to web data set : `http://pami.uwaterloo.ca/~hammouda/webdata`
[2] Link to mini newsgroup data set: `http://kdd.ics.uci.edu/`

Table 1. Showing data description

Data Set	*Name*	*Type*	*# of docs.*	*Classes*	*Avg. #of words /doc*
1.	UW-Can	HTML	314	10	469
2.	Mini 20-newsgroups	USENET	2000	20	151

5.2 Evaluation Measure

We use F-measure score to evaluate the quality of the clustering. F-measure combines *precision* and *recall* by calculating their harmonic mean. Let there be a class *i* and cluster *j*, then *precision* and *recall* of cluster *j* with respect to class *i* are as follows:

$$Precision(i,j) = \frac{n_{ij}}{n_j},\ Recall(i,j) = \frac{n_{ij}}{n_i} \tag{5}$$

where

- n_{ij} is the number of documents belonging to class *i* in cluster *j*.
- n_i is number of documents belonging to class *i*.
- n_j is the number of documents in cluster *j*.

Then F-score of class *i* is the maximum F-score it has in any of the clusters :

$$\mathrm{F(i)} = \frac{2*Precision*Recall}{(Precision+Recall)} \tag{6}$$

The overall F-score for clustering is the weighted average of F-score for each class *i*:

$$F_{overall} = \frac{\sum_i (n_i * F(i))}{\sum_i n_i} \tag{7}$$

Higher F-score suggests better clustering as produced clusters are mapping to original classes with higher accuracy.

5.3 Clustering Algorithm

For clustering, we use Group Hierarchical Agglomerative Clustering with complete linkage with the help of a java based tool[3].

5.4 Baseline Approach

We chose traditional tf-idf weighting based single term approach as our baseline approach since our aim is to investigate whether clustering can be improved by

[3] Link to tool : `http://www.cs.umb.edu/~smimarog/agnes/agnes.html`

Table 2. Showing baseline value of F-score with traditional vector based approach, for both the data sets

Data Set	*Baseline Value*
UW-Can	**0.7782**
Mini 20-newsgroups	**0.35126**

combing traditional inter-document similarities with inter-passage similarities, as suggested by us.

5.5 Results

Results have been summarized in Table3. For experiments with non-overlapping segments, we obtained maximum improvement of **7.39 %** and **10.86 %** in F-Score for UW-Can dataset and mini 20-newsgroups data set respectively. For experiments with over-lapping segments, we obtained maximum improvement of **10.04 %** for UW-Can data set and **7.02 %** for mini 20-newsgroup data set. For every experiment with overlapping segments, size of overlap is equal to half of window size.

Table 3. showing maximum improvement in terms of F-score over baseline approach with values of parameters like Window Size and Similarity Blend Factor

Data set	*Text Segments*	*Window Size*	*Similarity Blend Factor α*	*Maximum % improvement in F-score*
UW-Can	Non-Overlapping	225	0.45	**7.39 %**
UW-Can	Overlapping	425	0.45	**10.04 %**
Mini 20-Newsgroup	Non-Overlapping	150	0.45	**10.86 %**
Mini 20-Newsgroup	Overlapping	225	0.6	**7.02 %**

5.5.1 Graphs for Selected Values of Parameters Window Size and Similarity Blend Factor α

For all the experiments, similarity blend factor α assumes only five values i.e. 0.4, 0.45, 0.5, 0.55 and 0.6 as we want α to be moderate, so that the effectiveness of our method could be judged fairly. Similarity blend factor of 0.45 performs best for most of the experiments with both the data sets as evident from **Fig 2**, **Fig 4**, **Fig 6** and **Fig 8**. If **Fig 1**and **Fig 5** are compared with **Fig 3** and **Fig 7**, it is clear that a larger value of window size is required for better performance when dealing with overlapping windows. Window sizes used for mini 20-newsgroups are smaller as compared to those used for UW-Can. This is in accordance with their average document length. Performance will be reduced if larger windows are used for smaller documents.

1. For Data Set UW-Can

1.1 For Non-overlapping Text Segments

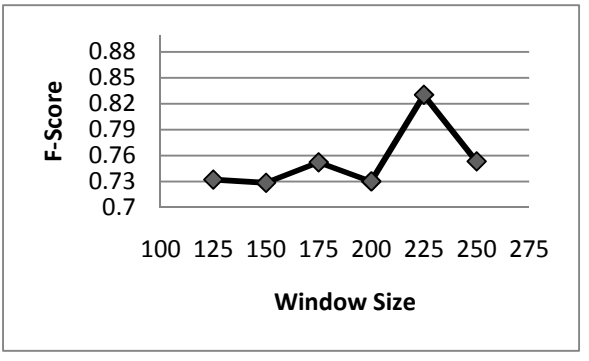


Fig. 1. Varying F-Score for different values of Window Size with α = 0.45

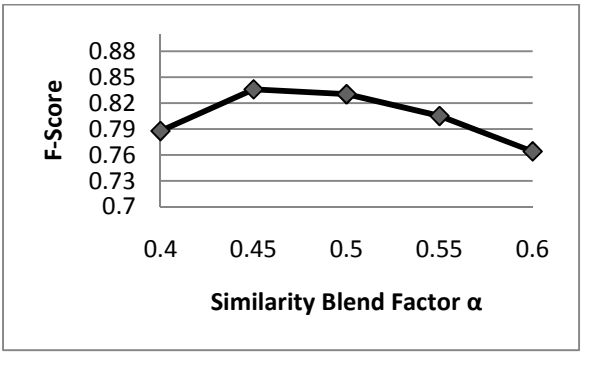


Fig. 2. Varying F-Score for different values of α with Window size of 225.

1.2 For Overlapping Text Segments

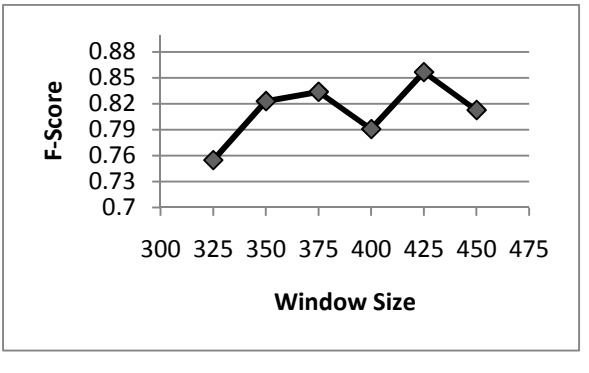


Fig. 3. Varying F-Score for different values of Window Size with α = 0.45

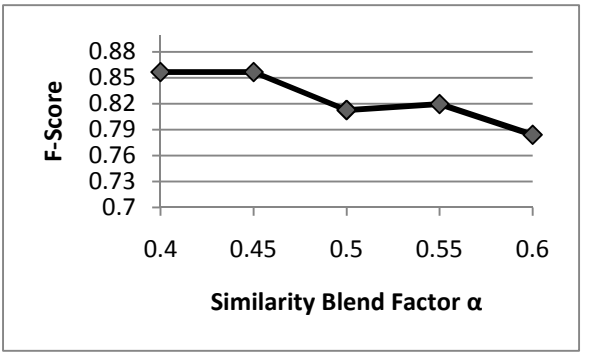


Fig. 4. Varying F-Score for different values of α with Window size of 225

2. For Data Set Mini 20-Newsgroups

2.1.1 For Non-Overlapping Text Segments

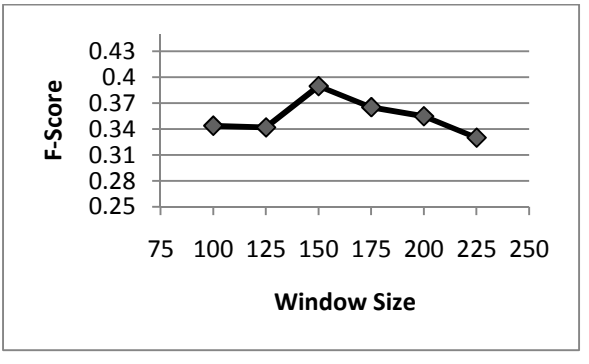


Fig. 5. Varying F-Score for different values of Window Size with α = 0.45

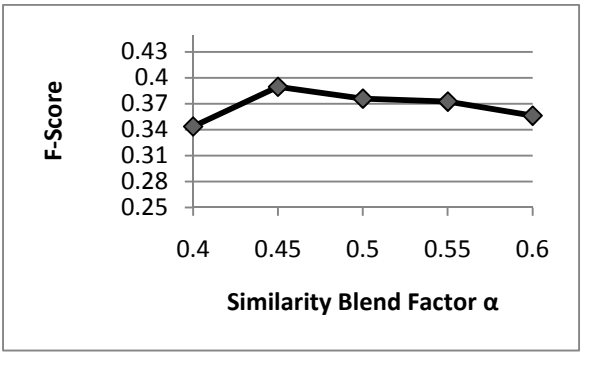


Fig. 6. Varying F-Score for different values of α with Window size of 225

2.2 For Overlapping Text Segments

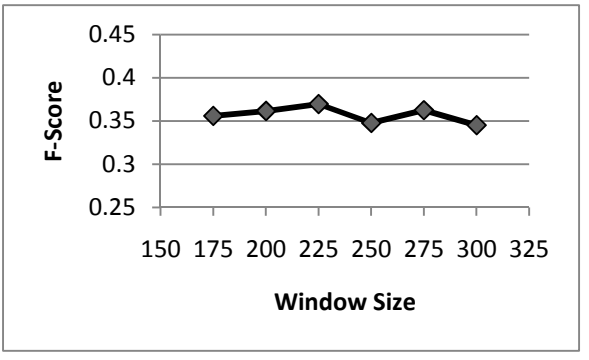


Fig. 7. Varying F-Score for different values of Window Size with α = 0.45

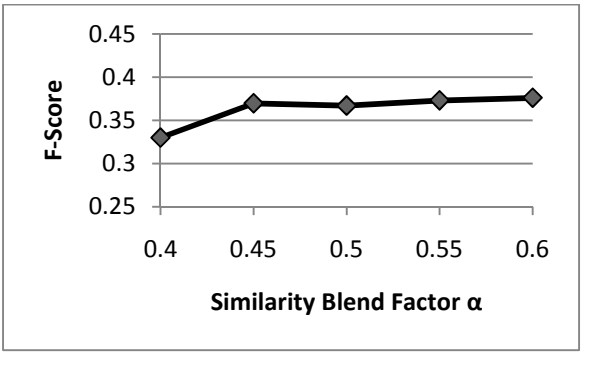


Fig. 8. Varying F-Score for different values of α with Window size of 225

6 Conclusion and Future Work

The presented approach might not provide best results but are definitely promising. It is to be kept in mind that the purpose of this paper is not suggesting an alternative clustering algorithm for text documents, but to determine whether document clustering can be improved or not, by combined usage of both inter-document and inter-passage similarities. There are many other possibilities such as to investigate effect on models other than vector space model, to take different similarity measure, to apply different weighting schemes for terms belonging to a text segment. In the future, we are working on developing a model which is suitable and makes use of inter-passage similarities more efficiently. Based on the results obtained, it is our intuition that if such a simple approach can improve the clustering then a more complex and complete approach can prove to be very useful and produce much better clustering.

References

1. Tellex, S., Katz, B., Lin, J., Fernandes, A., Marton, G.: Quantitative evaluation of passage retrieval algorithms for question answering. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval (SIGIR 2003), pp. 41–47. ACM, New York (2003)
2. Salton, G., Allan, J., Buckley, C.: Approaches to passage retrieval in full text information systems. In: Korfhage, R., Rasmussen, E., Willett, P. (eds.) Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1993), pp. 49–58. ACM, New York (1993)
3. Kaszkiel, M., Zobel, J.: Effective ranking with arbitrary passages. J. Am. Soc. Inf. Sci. Technol. 52(4), 344–364 (2001)
4. Conrad, J.G., Al-Kofahi, K., Zhao, Y., Karypis, G.: Effective document clustering for large heterogeneous law firm collections. In: Proceedings of the 10th International Conference on Artificial Intelligence and Law, ICAIL 2005 (2005)
5. Lamprier, S., Amghar, T., Levrat, B., Saubion, F.: Using Text Segmentation to Enhance the Cluster Hypothesis. In: Dochev, D., Pistore, M., Traverso, P. (eds.) AIMSA 2008. LNCS (LNAI), vol. 5253, pp. 69–82. Springer, Heidelberg (2008)
6. Cai, D., Yu, S., Wen, J.-R., Ma, W.-Y.: Block-based web search. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004), pp. 456–463. ACM, New York (2004)
7. Hotho, A., Staab, S., Stumme, G.: Wordnet improves text document clustering. In: Proceedings of the Semantic Web Workshop at SIGIR-2003, 26th Annual International ACM SIGIR Conference (2003b)
8. Hammouda, K.M., Kamel, M.S.: Efficient Phrase-Based Document Indexing for Web Document Clustering. IEEE Trans. on Knowl. and Data Eng. 16(10), 1279–1296 (2004)
9. Callan, J.P.: Passage-level evidence in document retrieval. In: Bruce Croft, W., van Rijsbergen, C.J. (eds.) Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1994), pp. 302–310. Springer-Verlag New York, Inc., New York (1994)
10. Hearst, M.A., Plaunt, C.: Subtopic structuring for full-length document access. In: Korfhage, R., Rasmussen, E., Willett, P. (eds.) Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development In Information Retrieval (SIGIR 1993), pp. 59–68. ACM, New York (1993)
11. Tagarelli, A., Karypis, G.: A segment-based approach to clustering multi-topic documents. In: Proceedings of the Text Mining Workshop, SIAM Data Mining Conference (2008)
12. Kim, J., Kim, M.H.: An Evaluation of Passage-Based Text Categorization. J. Intell. Inf. Syst. 23(1), 47–65 (2004)
13. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM 18(11), 613–620 (1975)
14. Chim, H., Deng, X.: A new suffix tree similarity measure for document clustering. In: Proceedings of the 16th International Conference on World Wide Web (WWW 2007), pp. 121–130. ACM, New York (2007)

Mining Ranking Models from Dynamic Network Data

Lucrezia Macchia, Michelangelo Ceci, and Donato Malerba

Dipartimento di Informatica, Università degli Studi di Bari,
via Orabona, 4 - 70126 Bari, Italy
lucrezia.macchia@uniba.it, {ceci,malerba}@di.uniba.it

Abstract. In recent years, improvement in ubiquitous technologies and sensor networks have motivated the application of data mining techniques to network organized data. Network data describe entities represented by nodes, which may be connected with (related to) each other by edges. Many network datasets are characterized by a form of autocorrelation where the value of a variable at a given node depends on the values of variables at the nodes it is connected with. This phenomenon is a direct violation of the assumption that data are independently and identically distributed (i.i.d.). At the same time, it offers the unique opportunity to improve the performance of predictive models on network data, as inferences about one entity can be used to improve inferences about related entities. In this work, we propose a method for learning to rank from network data when data distribution may change over time. The learned models can be used to predict the ranking of nodes in the network for new time periods. The proposed method modifies the SVM-Rank algorithm in order to emphasize the importance of models learned in time periods during which data follow a data distribution that is similar to that observed in the new time period. We evaluate our approach on several real world problems of learning to rank from network data, coming from the area of sensor networks.

1 Introduction

In recent years, learning preference functions has received increasing attention due to its potential application to problems raised in information retrieval, machine learning, data mining and recommendation systems [1], [6], [11].

As in many data mining tasks, when facing the problem of learning preference functions, new research frontiers in new application domains require capabilities of dealing with structured and complex data that in most of cases can be represented as data networks. In fact, networks have become ubiquitous in several social, economical and scientific fields ranging from the Internet to social sciences, biology, epidemiology, geography, finance and many others. Indeed, researchers in these fields have proven that systems of different nature can be represented as networks [18]. For instance, the Web can be considered as a network of webpages, which may be connected with each other by edges representing various

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 566–577, 2012.

explicit relations, such as hyperlinks. Sensor networks are networks where nodes represent sensors and edges represent the (spatial) distance between two sensors.

This particular organization of data adds additional complexity to the task at hand since networked data are characterized by a particular form of autocorrelation [14] according to which a value observed at a node depends on the values observed at neighboring nodes in the network [20]. The major difficulty due to the autocorrelation is that the independence assumption (i.i.d.), which typically underlies machine learning methods, is no longer valid. The violation of the instance independence has been identified as the main responsible of poor performance of traditional machine learning methods [17]. To remedy the negative effects of the violation of independence assumptions, autocorrelation has to be explicitly accommodated in the learned models.

Moreover, in the real world, network data may evolve over time. This evolution can be both in the structure of the network (nodes can be added or removed, edges can be added or removed) and in the distribution of the attribute values associated with the nodes. As an example, consider a sensor network whose nodes collect temperature, humidity, etc. at single positions in a specific environment. In this case, new sensors can be either added to the network or removed from it as well as the underlying data distribution of some variables may change. Indeed, as observed by Swanson [21], in this situation, data can be affected by temporal autocorrelation according to which two values of the some variable are cross correlated over a certain time lag.

In this paper, we argue that a method for learning preference functions from network data should take both network and temporal autocorrelation into account. At this aim, we propose two solutions, both based on the well known SVMRank algorithm [15]. The proposed solutions allow us to learn preference functions over a series of consecutive time intervals by taking network autocorrelation into account. The first solution uses a fading factor that allows the algorithm to give more importance to models learned in recent time periods than models learned in the past. The second solution allows the algorithm to give more importance to models associated to more correlated time intervals than models associated to less correlated time intervals. This means that, while models learned according to the first solution give more importance to the order in which the time intervals are considered, in the second solution, more importance is given to the similarity between data distributions observed in two distinct time intervals (periodicity, if present, can be captured).

The paper is organized as follows. The next section reports relevant related work. Section 3 describes the proposed approaches. Section 4 describes the datasets, experimental setup and reports relevant results. Finally, in Section 5, some conclusions are drawn and some future work are outlined.

2 Related Work

The task considered in this work is that of preference leaning functions. The aim of the methods developed in this field is to learn a ranking model which returns

the output predictions in the form of a ranking of the examples given in input. It is possible distinguish three types of ranking problems [5]:

- *Label ranking*: the goal is to learn a "label ranker" in the form of an $X \rightarrow S_Y$ mapping, where the input space X is the feature space and the output space S_Y is given by the set of all total orders (permutations) of the set of labels Y.
- *Instance ranking*: an instance x ∈ X belongs to one among a finite set of classes $Y = y_1, y_2, \ldots, y_k$ for which a natural order $y_1 < y_2 < \ldots < y_k$ is defined.
- *Object ranking*: the goal is to learn a ranking function $f(\cdot)$ which, given a subset Z of an underlying referential set Z of objects as an input, produces a ranking of these objects.

In this work we consider the object ranking problem, where objects $x \in X$ are described in terms of an attribute-value representation, but can be linked each other on the basis of a network structure. As training information, an object ranker has access to exemplary rankings or pairwise preferences of the form $x_i > x_j$ suggesting that x_i should be ranked higher than x_j.

Studies reported in the literature solve this problem by resorting to two alternative solutions. The first solution determines a function that assigns a numerical value to each element of a set, then the same is used to sort the items. The second solution aims at learning preference functions, which permit to perform pairwise comparisons in order to define a relative order between two objects [8,13,3]. The first solution is generally more efficient but it is applicable only when a single total order between objects is acceptable. While, when not all the objects have to necessarily be included in the ranking, the second solution is preferable. Since pairwise total orders can lead to define partial orders, in this work we consider the first solution.

Concerning this solution, Herbrich et al. [12] propose to learn a function which, given an object description, returns an item belonging to an ordered set. The function is determined so that a loss function is minimized. A similar approach was proposed by Crammer et al. [4], in which the learned functions are modeled by perceptrons. Tesauro [22] proposed a symmetric neural network architecture that can be trained with representations of two states and a training signal that indicates which of the two states is preferable. In the framework of *constraint classification* [9,10], some authors exploit linear utility functions to find a way to express a constraint in the form $f_i(x) - f_j(x) > 0$, in order to transform the original ranking problem into a single binary classification problem.

However, most of the works presented in the literature neither consider the possible network structure according to which examples can be arranged nor consider the possible evolution of the network. Exceptions are represented by ranking algorithms used in information retrieval to rank web pages by taking hyperlinks into account (e.g. PageRank-like algorithms [19]). In this case, however, the possible evolution of the network is not taken into account. In addition, they do not consider autocorrelation properly since they do not consider the fact that the values observed at a node depend on the values observed at linked nodes

in the network. Other exceptions are represented by approaches that resort to a multi-relational data mining framework which implicitly takes autocorrelation into account [2,16]. These works, however, resort to the second solution since they are able to obtain a relative order between two objects.

3 Learning Ranking Functions with Different Time Windows

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. The foundations of SVMs have been presented by Vapnik in [23] and are related to the computational learning theory.

In the classical classification case, the problem solved by SVMs can be formalized in the following way: given a set of positive and negative examples $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where $x_i \in X \subseteq \mathbb{R}^m$ (x_i is a feature vector) and $y_i \in \{-1, +1\}$, an SVM identifies the hyperplane in $\mathbb{R}^m$ that linearly separates positive and negative examples with the maximum margin (optimal separating hyperplane). In the case of network data, the weights associated to the edges allow us to represent nodes as examples and their position in the feature space. In this way, the identified hyperplane takes into account the "position" of the examples in the feature space.

In this work, we exploit SVMs in order to rank examples instead of generating an optimal separating hyperplane. Indeed, this idea is not novel and in SVMRank[1] [15] an optimization problem that permits the definition of a ranking function is defined.

In detail, SVMRank algorithm resolves the following optimization problem:

Given a training set $(x_1, y_1), \ldots, (x_n, y_n)$ *with* $x_i \in \mathbb{R}^m$ *and* $y_i \in Y$*, where* x_i *represents the example and* y_i *its ordinal value in the ranking, the problem is to find a ranking function* $h : \mathbb{R}^m \to \mathbb{R}$ *defined as* $h(x) = w^T x$*, such that the following optimization problem is solved:*

$$argmin_{w,\xi \geq 0} : \frac{1}{2}\vec{w}^T \cdot \vec{w} + C\xi$$
$$s.t. \forall (i,j) \in P, \forall c_{ij} \in \{0,1\} : \frac{1}{|P|} w^T \Sigma_{i=1}^{|P|} c_{ij}(x_i - x_j) \geq \frac{1}{|P|} \Sigma_{i=1}^{|P|} c_{ij} - \xi$$

where ξ is a slack variable, P is the set of pairs (i, j) for which example x_i has a higher rank than example x_j, i.e. $P = \{(i, j)|y_i > y_j\}$ and C is a positive regularization constant. The regularization constant in the cost function defines the trade-off between a large margin and misclassification error (i.e. empirical risk minimization). Intuitively, this formulation finds a large-margin linear function $h(x)$ that minimizes the number of pairs of training examples that are swapped w.r.t. their desired order.

However, this optimization, permits us to learn a ranking model for *static* training data. In our case, we can learn a different ranking model for each time

[1] Downloaded from
http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

interval. The models can then be combined in order to identify the final model to be used for the next time interval.

More formally, let

$$S' = \bigcup_{t=1}^{z} S_t \tag{1}$$

be the complete training dataset where $S_t = \{(x_{t,1}, y_{t,1}), \ldots, (x_{t,n_t}, y_{t,n_t})\}$ represents the training dataset for the t-th time-interval, z be the total number of time-intervals.

Then the following optimization problem is solved:

$$argmin_{w_t,\xi\geq 0} : \frac{1}{2} w_t^T \cdot w_t + C\xi \tag{2}$$

$$s.t. \forall (t = 1, \ldots, z), \forall (x_{t,i}, x_{t,j}) \in S_t, \forall c_{ij} \in \{0, 1\} :$$
$$\frac{1}{|P|} w_t^T \Sigma_{i=1}^{|P|} c_{ij}(x_{t,i} - x_{t,j}) + 1 - \varsigma(x_{t,i}, x_{t,j}, S', S_{z+1}) \geq \frac{1}{|P|} \Sigma_{i=1}^{|P|} c_{ij} - \xi$$

where S_{z+1} represents the nodes of the network at the $z+1$ time interval (on which prediction is performed) and $\varsigma(x_{t,i}, x_{t,j}, S', S_{z+1})$ modifies the constraint by considering both network autocorrelation and possible dependence with models observed at previous time windows.

In order to compute $\varsigma(x_{t,i}, x_{t,j}, S', S_{z+1})$, two variants are considered. In the first variant (called $SVMRank_R$), the function $\varsigma(x_{t,i}, x_{t,j}, S', S_{z+1})$ emphasizes recent data with respect to past data:

$$\varsigma(x_{t,i}, x_{t,j}, S', S_{z+1}) = \frac{t}{z} + dist(x_{t,i}, x_{t,j}) \tag{3}$$

where $dist(x_{t,i}, x_{t,j})$ is the distance between node i and node j and $\frac{t}{z}$ is a fading factor.

In the second variant (called $SVMRank_T$), the function $\varsigma(x_{t,i}, x_{t,j}, S', S_{z+1})$ emphasizes the models observed at previous time windows which are (supposed to be) more similar to the model at time $z+1$. This similarity is computed according to the Durbin-Watson statistic [7] $d(S', S_{z+1})$ defined on all the features and all the nodes[2]:

$$d(S', S_{z+1}) = \frac{1}{m} \sum_{l=1,\ldots,m} \left(\frac{1}{n} \sum_{i=1,\ldots,n} \frac{\Sigma_{t=2}^{z+1} (e_{t,i}^{(l)} - e_{t-1,i}^{(l)})^2}{\Sigma_{t=1}^{z} {e_{t,i}^{(l)}}^2} \right) \tag{4}$$

where $e_{t,i}^{(l)}$ is the value of the l-th feature of $x_{t,i}$. When there is high positive (negative) temporal autocorrelation, $d(S', S_{z+1})$ approaches to 0 (4), if there is no temporal autocorrelation, $d(S', S_{z+1})$ approaches to 2.

Once all w_t, $t = 1, \ldots, z$ are computed, the average vector $w = 1/z \sum_{t=1,\ldots,z} w_t$ is the ranking vector used in the prediction of the ranking for nodes at time

[2] Computation of $d(S', S_{z+1})$ only considers nodes that are present in all time windows in S'.

$z+1$. In particular, the orthogonal projection of nodes over w implicitly defines the ranking (see figure 1).

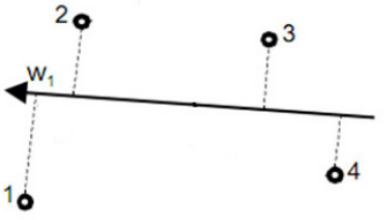


Fig. 1. A ranking function that perfectly ranks objects according to their label

In this variant, the function $\varsigma(x_{t,i}, x_{t,j}, S', S_{z+1})$ is defined as:

$$\varsigma(x_{t,i}, x_{t,j}, S', S_{z+1}) = d(S', S_{z+1}) + dist(x_{t,i}, x_{t,j}) \tag{5}$$

Intuitively, the first solution returns a final model that gives more importance to models learned in the recent time period than models learned in the past, while the second solution returns a final model that considers the relationship between values separated from each other by a given time lag, computed with the Durbin-Watson statistic.

The two variants are implemented by exploiting a modified version of the algorithm proposed in [15], where new constraints are used. Similarly to what proposed in [15], also in our case, the considered optimization problem in (2) can be solved in logarithmic time by means of the algorithm reported in Algorithm 1, where $c^+ = [c_i^+]_{i=1,\ldots,n}$ and $c^- = [c_i^-]_{i=1,\ldots,n}$ are vectors whose values are defined as follows:

$$c_i^+ = |\{j|(x_j, y_j) \in S_t \wedge y_i > y_j \wedge ((w^T x_i) - (w^T x_j) < 1 + 1 - \varsigma(x_{t,i}, x_{t,j}, S', S_{z+1}))\}|$$
$$c_i^- = |\{j|(x_j, y_j) \in S_t \wedge y_j > y_i \wedge ((w^T x_j) - (w^T x_i) < 1 + 1 - \varsigma(x_{t,i}, x_{t,j}, S', S_{z+1}))\}|$$

Intuitively, these are constraints over false positive and false negative errors that take constraints in (2) into account. Coherently, constraints in (2) are also taken into account in the definition of the stopping criterion (line 24).

4 Experiments

In order to evaluate the effectiveness of the proposed solution, we performed experiments on three real world datasets, that is, Intel Lab Database, California Truck and Portuguese rivers database. In all the experiments, we set C=20 (after preliminary experiments aiming at identify the best C value among the values in the set {10, 15, 20, 30}). In the experiments, in turn, we used the last time interval as testing set.

Algorithm 1. Training Ord. Regr. SVMs

Input: $S = ((x_1, y_1), ..., (x_n, y_n)), C$

$W \leftarrow \oslash$
repeat
  $(w, \xi) \leftarrow argmin_{w,\xi>0} \frac{1}{2}\vec{w}^T \cdot \vec{w} + C\xi$
  $s.t. \forall (c^+, c^-) \in W : \frac{1}{|P|} w^T \Sigma_{i=1}^n (c_i^+ - c_i^-) x_i \geq \frac{1}{2|P|} \Sigma_{i=1}^n (c_i^+ - c_i^-) - \xi$
  sort S by decreasing $w^T x_i$
  $c^+ \leftarrow 0; c^- \leftarrow 0$
  $n_r \leftarrow$ number of examples with $y_i = r$
  for $r = 2$ to R **do**
    $i \leftarrow 1; j \leftarrow 1; a \leftarrow 0; b \leftarrow 0$
    while $1 \leq n$ **do**
      if $y_i = r$ **then**
        while $(j \leq n) \bigwedge (w^T x_i - w^T x_j < 1)$ **do**
          if $y_i < r$ **then**
            $b++; c_j^- \leftarrow c_j^- + (n_r a + 1)$
          end if
          $j++$
        end while
        $a++; c_i^+ \leftarrow c_i^+ + b$
      end if
      $i++$
    end while
  end for
  $W \leftarrow W \bigcup \{(c^+, c^-)\}$
until $\frac{1}{2|P|} \Sigma_{i=1}^n (c_i^+ - c_i^-) - \frac{1}{|P|} \Sigma_{i=1}^n (c_i^+ - c_i^-)(w^T x_i) + 1 - \varsigma(x_{t,i}, x_{t,j}, S', S_{z+1})) \geq \xi + \epsilon$

In order to evaluate the learned ranking models, we used the Spearman's rank correlation coefficient.

Spearman's rank correlation coefficient is the non-parametric alternative to correlation and can be used when the data do not meet, as in this case, the assumptions about normality, homoscedasticity and linearity. Let $S_{z+1} = \{(x_{z+1,1}, y_{z+1,1}), \ldots, (x_{z+1,n_{z+1}}, y_{z+1,n_{z+1}})\}$ be the real dataset at time $z+1$ and $y'_{z+1,i} = h(x_{z+1,i})$ be the estimated ranking for the example $x_{z+1,i}$, the Spearman's rank correlation coefficient is defined as:

$$\rho = \frac{\sum_{i=1,\ldots,n_{z+1}} (y_{z+1,i} - \overline{y_{z+1}})(y'_{z+1,i} - \overline{y'_{z+1}})}{\sqrt{\sum_{i=1,\ldots,n_{z+1}} (y_{z+1,i} - \overline{y_{z+1}})^2 (y'_{z+1,i} - \overline{y'_{z+1}})^2}} \tag{6}$$

where $\overline{y_{z+1}}$ ($\overline{y'_{z+1}}$) is the average ranking. ρ ranges in the interval [-1,1], where -1 means negative correlation in the ranking and 1 means perfect ranking.

Intel Lab Database contains real information collected from 54 sensors deployed in the Intel Berkeley Research lab between February 28th and and March 21st, 2004. The sensors which we consider in this experiment have collected timestamped temperature, humidity and luminosity values once every 31 seconds. Networks are built by considering the spatial distance between sensors

Table 1. Intel lab Dataset: Spearman's rank coefficient

Train	Test	SVMRank	$SVMRank_R$	$SVMRank_T$
1 2	3	0.8743061	0.8743061	**0.9021739**
1 2 3	4	0.9531683	**0.9532839**	0.9529370
1 2 3 4	5	0.9237974	**0.9382516**	0.9364014
1 2 3 4 5	6	0.9153561	0.9210222	**0.9240286**
1 2 3 4 5 6	7	0.9084181	**0.9087650**	0.9084181
1 2 3 4 5 6 7	8	**0.5570074**	0.5498381	0.5514569
1 2 3 4 5 6 7 8	9	0.8592738	0.8591581	**0.8632053**
1 2 3 4 5 6 7 8 9	10	**0.8953515**	0.8943108	0.8931544
1 2 3 4 5 6 7 8 9 10	11	**0.8441258**	0.8236586	0.8375346
1 2 3 4 5 6 7 8 9 10 11	12	0.8316373	0.8156799	**0.8341813**
1 2 3 4 5 6 7 8 9 10 11 12	13	**0.7602624**	0.7513586	0.7514743
1 2 3 4 5 6 7 8 9 10 11 12 13	14	0.7264974	0.7209470	**0.8169229**
1 2 3 4 5 6 7 8 9 10 11 12 13 14	15	0.8147259	0.8111413	**0.9021739**

and the target attribute is represented by the temperature (we removed temperature and we used the temperature ranking as y value). In the experiments, we only considered working days and we used 1-day time-intervals, this means that we built 15 networks in all.

In Table 1, results of the Spearman's rank correlation coefficient are reported. They show, as expected, that $SVMRank_R$ and $SVMRank_T$ in most of cases outperform the $SVMRank$ algorithm. By comparing $SVMRank_T$ with $SVMRank_R$, it is possible to see that $SVMRank_T$ shows better performances. This is due to the consideration that taking also into account temporally distant time windows is beneficial. Moreover, in presence of a fast concept drift, the algorithm is not able to immediately adapt to the data. This last aspect is confirmed by results on day 12, when there is small temporal autocorrelation with day 11 (see Table 2).

The Portuguese rivers dataset holds water's information of the rivers Douro e Paiva. The dataset may be incomplete because the controls are manually done and are not done systematically. The original dataset is composed of a fact table and six additional relational tables: The fact table (ANALYSIS) contains information on the measures under control (pH, % Coliformi Bacteria, conductivity, turbidity, % Escherichia Coli Bacteria) and the gathering method. Additional tables are directly (or indirectly) connected to ANALYSIS according to a snowflake logic schema. They are: PARAMETERS (that are considered in the analysis), INSTITUTIONS (that collected data), DAY, CONTROL POINTS and PATH (that specifies the position of a control point according to the course of the rivers). From the table PATH we got the course of the river and the position of the control points in order to build the network structure. The weights on the edges represent the navigation distance between the control points (in all we have 115 control points). We considered data aggregated by year and for each node, we represented institution, gathering method, pH, % Coliformi Bacteria, conductivity, turbidity, % Escherichia Coli Bacteria. Aggregation is performed

Table 2. Durbin-Watson - Intel lab dataset

Temporal Series	Durbin-Watson
1 - 2	0.0010
2 - 3	0.0017
3 - 4	0.0010
4 - 5	0.0004
5 - 6	0.0028
6 - 7	0.0006
7 - 8	0.0006
8 - 9	0.0033
9 - 10	0.0048
10 - 11	0.0008
11 - 12	0.2649
12 - 13	0.0003
13 - 14	0.0013

Table 3. Portuguese rivers dataset: Spearman's rank coefficient

Train	Test	SVMRank	$SVMRank_R$	$SVMRank_T$
2004-2005	2006	0.2969208	0.3643695	**0.4024926**
2004-2005-2006	2007	0.3592375	0.3526392	**0.4761730**
2004-2005-2006-2007	2008	0.2459677	0.2203079	**0.4769061**
2004-2005-2006-2007-2008	2009	0.2078445	0.2214076	**0.4226539**

by considering mode (average) for discrete (continuous) values. In all, we considered 6 years (from 2004 to 2009). The experiments are performed using the pH feature as target since it is recognized to be a good indicator of river pollution.

Results of the Spearman's rank correlation coefficient (reported in Table 3) show that $SVMRank_R$ is not able to improve SVMRank algorithm. This is mainly due to the fact that this dataset does not exhaustively represent the network structure. However, as in the case of the Intel Lab dataset, $SVMRank_T$ significantly outperforms SVMRank and $SVMRank_R$.

The California traffic dataset concerns the traffic on the highways of California. The dataset is taken from http://traffic-counts.dot.ca.gov/index.htm. The experiments are carried out using the following independent attributes observed by sensors on highways: the percentage of trucks, the percentage of 2-axle vehicles, the percentage of 3-axle vehicles, the percentage of 4-axle vehicles, the

Table 4. Durbin-Watson - Portuguese rivers dataset

Temporal Series	Durbin-Watson
2004-2005	0.0004
2005-2006	0.0002
2006-2007	0.0001
2007-2008	0.0001
2008-2009	0.0338

Table 5. California traffic dataset: Spearman's rank coefficient

Train	Test	SVMRank	$SVMRank_R$	$SVMRank_T$
2001-2002	2003	**0.7489309**	0.7398230	0.7484557
2001-2002-2003	2004	**0.7484557**	0.7419486	0.7417657
2001-2002-2003-2004	2005	**0.7402829**	0.7255635	0.7402272
2001-2002-2003-2004-2005	2006	0.7465108	0.7412590	**0.7468870**
2001-2002-2003-2004-2005-2006	2007	0.7456102	0.7376070	**0.7447633**
2001-2002-2003-2004-2005-2006-2007	2008	0.7462663	0.7490326	**0.7500496**
2001-2002-2003-2004-2005-2006-2007-2008	2009	0.7370597	0.736184	**0.7375568**

percentage of 5-axle vehicles. The goal is to rank sensors' positions on the basis of the sum of the volumes of traffic on a road in both directions. Each sensor represents a node in the network (in all, we have 969 sensors), while weights on the edges represent the driving distance between two sensors. However, the network is not fully connected and only nodes whose driving distance is less than 25 miles are connected (in all, there are 34093 edges). The dataset refers to the period 2001-2009 and for each year, a network is created.

Results of the Spearman's rank correlation coefficient confirm results obtained on previously analyzed datasets. Moreover, $SVMRank_T$ performances improve with an increasing history. This result can be motivated by the high temporal autocorrelation of the dataset (see Table 6).

Table 6. Durbin-Watson - California traffic dataset

Temporal Series	Durbin-Watson
2001-2002	0.0013
2002-2003	0.0017
2003-2004	0.0014
2004-2005	0.0009
2005-2006	0.0005
2006-2007	0.0007
2007-2008	0.0010
2008-2009	0.0002

5 Conclusions

In this paper we have faced the problem of mining ranking models from networked data whose data distribution may change over time. The proposed method modifies the well known SVMRank algorithm in order to emphasize the importance of models learned in time periods during which data follow a data distribution that is similar to that observed in the time period for which prediction has to be made. Extensions are framed in an ensemble learning framework and allow us to take both network and temporal autocorrelation into account. At this aim, we propose two solutions. The first solution uses a fading factor that allows the algorithm to give more importance to models learned in recent

time periods than models learned in the past. The second solution allows the algorithm to give more importance to models associated to more correlated time intervals than models associated to less correlated time intervals.

We evaluate our approach on several real world problems of learning to rank from network data, coming from the area of sensor networks. Experimental results empirically prove that the second solution significantly outperforms the first solution in capturing the concept drift.

Acknowledgment. The authors thank Luis Torgo for kindly providing Portuguese rivers dataset. This work is supported in fulfillment of the research objectives of the project: "EMP3: Efficiency Monitoring of Photovoltaic Power Plants" funded by "Fondazione Cassa di Risparmio di Puglia".

References

1. Aiolli, F.: A preference model for structured supervised learning tasks. In: ICDM, pp. 557–560. IEEE Computer Society (2005)
2. Ceci, M., Appice, A., Loglisci, C., Malerba, D.: Complex objects ranking: a relational data mining approach. In: Shin, S.Y., Ossowski, S., Schumacher, M., Palakal, M.J., Hung, C.C. (eds.) SAC, pp.1071–1077. ACM (2010)
3. Cohen, W.W., Schapire, R.E., Singer, Y.: Learning to order things. J. Artif. Int. Res. 10, 243–270 (1999)
4. Crammer, K., Singer, Y.: Pranking with ranking. In: NIPS, pp. 641–647. MIT Press (2001)
5. Dembczyski, K., Kotlowski, W., Slowiski, R., Szelag, M.: Learning of rule ensembles for multiple attribute ranking problems. In: Fürnkranz, J., Hüllermeier, E. (eds.) Preference Learning, pp. 217–247. Springer (2010)
6. Doyle, J.: Prospects for preferences. Computational Intelligence 20(2), 111–136 (2004)
7. Draper, N., Smith, H.: Applied Regression Analysis, 2nd edn. Wiley, New York (1981)
8. Fürnkranz, J., Hüllermeier, E.: Pairwise Preference Learning and Ranking. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) ECML 2003. LNCS (LNAI), vol. 2837, pp. 145–156. Springer, Heidelberg (2003)
9. Har-Peled, S., Roth, D., Zimak, D.: Constraint Classification: A New Approach to Multiclass Classification. In: Cesa-Bianchi, N., Numao, M., Reischuk, R. (eds.) ALT 2002. LNCS (LNAI), vol. 2533, pp. 365–379. Springer, Heidelberg (2002)
10. Har-Peled, S., Roth, D., Zimak, D.: Constraint classification for multiclass classification and ranking. In: Becker, S., Thrun, S., Obermayer, K. (eds.) Advances in Neural Information Processing Systems 15 (NIPS 2002), pp. 785–792 (2003)
11. Herbrich, R., Graepel, T., Bollmann-sdorra, P., Obermayer, K.: Learning preference relations for information retrieval (1998)
12. Herbrich, R., Graepel, T., Obermayer, K.: Large margin rank boundaries for ordinal regression. MIT Press (2000)
13. Hüllermeier, E., Fürnkranz, J., Cheng, W., Brinker, K.: Label ranking by learning pairwise preferences. Artif. Intell. 172(16-17), 1897–1916 (2008)
14. Jensen, D., Neville, J.: Linkage and autocorrelation cause feature selection bias in relational learning. In: Proc. 9th Intl. Conf. on Machine Learning, pp. 259–266. Morgan Kaufmann (2002)

15. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2002, pp. 133–142. ACM, New York (2002)
16. Malerba, D., Ceci, M.: Learning to Order: A Relational Approach. In: Raś, Z.W., Tsumoto, S., Zighed, D.A. (eds.) MCD 2007. LNCS (LNAI), vol. 4944, pp. 209–223. Springer, Heidelberg (2008)
17. Neville, J., Simsek, O., Jensen, D.: Autocorrelation and relational learning: Challenges and opportunities. In: Wshp. Statistical Relational Learning (2004)
18. Newman, M.E.J., Watts, D.J.: The structure and dynamics of networks. Princeton University Press (2006)
19. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab (November 1999), http://ilpubs.stanford.edu:8090/422/, previous number = SIDL-WP-1999-0120
20. Stojanova, D., Ceci, M., Appice, A., Džeroski, S.: Network Regression with Predictive Clustering Trees. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011. LNCS, vol. 6913, pp. 333–348. Springer, Heidelberg (2011)
21. Swanson, B.J.: Autocorrelated rates of change in animal populations and their relationship to precipitation. Conservation Biology 12(4), 801–808 (1998)
22. Tesauro, G.: Connectionist learning of expert preferences by comparison training. In: Advances in Neural Information Processing Systems 1, pp. 99–106. Morgan Kaufmann Publishers Inc., San Francisco (1989)
23. Vapnik, V., Golowich, S.E., Smola, A.: Support vector method for function approximation, regression estimation, and signal processing. In: Advances in Neural Information Processing Systems 9, pp. 281–287. MIT Press (1996)

Machine Learning-Based Classification of Encrypted Internet Traffic

Talieh Seyed Tabatabaei, Mostafa Adel, Fakhri Karray, and Mohamed Kamel

Centre for Pattern Analysis and Machine Intelligence (CPAMI)
University of Waterloo, Waterloo, Ontario, Canada
{tseyedta,m22hassa,fakhri,mkamel}@uwaterloo.ca

Abstract. Peer-to-peer (P2P) networking has introduced a major shift in the application and traffic mix of the Internet and established itself as the main driver of increasing traffic volume. The high requirements of some P2P applications result in network operational issues: these applications consume vast amounts of network resources and can prevent mission critical applications from accessing the network. Therefore the ability to correctly identify them can be crucial for many network management and measurement tasks. In this paper some flow-based statistical features of Internet traffic are investigated in order to detect P2P traffic. We propose a system to identify the BT traffic, which is one of the most popular and problematic P2P applications using support vector machines. The accuracy of 94.5% was achieved for recognizing encrypted traffic which is a very promising result.

Keywords: Internet traffic classification, support vector machines, feature selection, mutual information.

1 Introduction

The concept of identifying protocols and applications through analysis of network traffic is known as "traffic classification". Techniques for traffic classification are used in many different applications, such as Quality of Service (QoS) assignments, traffic shaping, Intrusion Detection Systems (IDS) [2] and in network forensics solutions.

In recent years, Peer-to-Peer (P2P) file-exchange applications have overtaken Web applications as the major contributor of traffic on the Internet [1]. Recent estimates put the volume of P2P traffic at 70% of the total broadband traffic. P2P is often used for illegally sharing copyrighted music, video, games, and software. The legal ramification of this traffic combined with its aggressive use of network resources has necessitated a strong need for identification of network traffic by application type.

The P2P paradigm has proven to be much more efficient than client-server communication especially for fast distribution of large amounts of data, since bottlenecks at servers are avoided by distributing requested data and the available access capacity over a global community of recipients.

The high requirements of some P2P applications result in network operational issues: these applications consume vast amounts of network resources and can prevent mission

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 578–592, 2012.

critical applications from accessing the network. On the other hand, P2P applications can cause security and legal troubles for network administrators. Therefor having the ability to correctly identify them can be essential for several network management and measurement tasks, including traffic engineering, service differentiation, performance/failure monitoring, and security. P2P not only results in increased network complexity due to the enormous volume of traffic, but also requires huge extra costs such as the cost of upgrading the infrastructure and network repartitioning [3][4].

In this work we are trying to identify encrypted BitTorrent traffic by using some flow-based statistical features. Support vector machines (SVMs) are used to classify BitTorrent traffic from the aggregate Internet traffic.

The rest of this paper is organized as follows: Section 2 is about related works. Section 3 is a brief review of P2P and BitTorrent architecture. Section 4 demonstrates the structure of the P2P traffic identification system proposed in this work and the corresponding steps. In Section 5 the theory of SVM is concisely discussed. In Section 6 the experimental results are presented and Section 7 is the conclusion.

2 Related Work

There are two conventional approaches for Internet traffic classification: 1) port-based method and 2) deep packet inspection (DPI). The first method classifies the application type using the official Internet Assigned Numbers Authority (IANA) list. Initially it was considered to be simple and easy to implement port-based in real time. However, mapping traffic to applications based on port numbers is now ineffective since many P2P applications now use dynamically assigned ports and other known port numbers (e.g. http and ftp) for their sub transactions to disguise their traffic [5][6].

In the DPI approach packet payloads are examined to search for exact signatures of known applications [6][7]. In this method the protocol specific string in the payload can be used for identification, so the P2P traffic can be identified through analyzing the characteristic bit strings in packet payload [7][8]. Sen et. al [8] suggest an efficient method for detecting the P2P application traffic through application level signatures. They identify the application-level signatures by examining some available documentations and packet-level traces, and then use the identified signatures to develop online filters that can efficiently and accurately track the P2P traffic even on high-speed network links. M. Roughan et. al [9] provide a solution framework for measurement-based identification of traffic for QoS based on statistical application signatures.

DPI technique is broadly used in commercial tools such as Ipoque, CISCO NBAR, Sandvine, and SonicWall. However, there are a couple of limitations associated with this method: this technique only identifies traffic for which signatures are available and therefor is not adaptive to the new emerging protocols. Second, it can cause privacy concerns. And finally this technique will fail if payload is encrypted. The last problem is the most important reason that makes DPI techniques impractical for detecting P2P traffic since most of the popular P2P protocols, such as BitTorrent (BT) are using payload encryption to avoid detection.

Considering the limitations and drawbacks of the aforementioned approaches, machine learning (ML) techniques have become a popular alternative in classifying

flows based on application protocol payload-independent statistical features such as packet length and inter-arrival times, flow lengths, etc.

Each traffic flow is characterized by the same set of payload-independent statistical features. A ML classifier is built by training on a representative set of flow instances where the network applications are known. The trained classifier can be applied to determine the class of unknown flows. Statistical analysis based approach treats the problem of application classification as a statistical problem.

ML-based approach is independent of packet payload inspection so it is robust to encryption. Also it is scalable and adaptive when new protocols join. Different classes of traffic based on different applications, different features, and various supervised and unsupervised, deterministic and probabilistic ML methods have been utilized during the recent years in order to classify network traffic flows [5][10][11]. Each flow is described by a set of statistical features and associated feature values and accordingly classify by a machine learned classifier. The idea of using ML techniques for flow classification was first introduced in the context of intrusion detection [13]. Moore et al. [12] used a Naive Bayes classifier which was a supervised machine learning approach to classifying internet traffic. Williams et al. [10] compared five machine learning algorithms, among these algorithms, C4.5 achieved the highest accuracy in their results. McGregor et al. [14] used Expectation Maximization (EM) algorithm to cluster flows, but the reported results are not very promising. Zander et al. [15] extended this work by using an EM algorithm called Auto Class, and found the optimal feature subset for classifying traffic.

In this work our focus is on identification of BitTorrent (BT) traffic only as currently it is the most popular p2p file-sharing protocol in North America and pretty much all around the world. Based on the latest Internet traffic report by Sandvine, BitTorrent, and P2P traffic in general, is still dominant in all geographical regions. In North America, Latin America and Asia-Pacific, P2P traffic is responsible for the vast majority of all upstream traffic. BitTorrent remains the most used file-sharing protocol in North America, and the total amount of P2P traffic is still very significant. Sandvine's research reveals that on an average day, 53.3% of all upstream traffic can be attributed to P2P applications. The bandwidth usage patterns during peak hours are slightly different, but still a massive 34.31% of all upstream traffic can be attributed to BT at these times.

According to Sandvine's traffic analysis, the normalized aggregate of all traffic (up/down) during peak hours puts P2P traffic at 19.2% during the first months of 2010. Interestingly, this is up from 15.1% in 2009, which shows that P2P traffic is growing strongly, not only in absolute numbers but also as a share of total Internet traffic in North America.

Different flow-based statistical features are going to be investigated in order to find the most effective and discriminative variables for identifying BT traffic. SVMs which is a powerful supervised classifier is going to be used for classifying the traffic.

3 BitTorrent Architecture Overview

The P2P data dissemination model has become widely popular for a variety of applications including streaming multimedia, voice-over-IP (VoIP) and file sharing. This is

largely a consequence of the many benefits that P2P architectures offer in comparison to the traditional client/server communication model. The P2P data broadcasting model has become extensively popular for a variety of applications including streaming multimedia, voice-over-IP (VoIP) and file sharing. This is mainly a result of the many advantages that P2P architectures offer in comparison to the traditional client/server communication model.

BT is the most popular P2P protocol for file sharing applications. Sharing a file with BT is very simple and proceeds as follows:

1. The file is broken into several fixed-sized pieces. A cryptographic hash is computed for each piece to ensure integrity as pieces are shared. This aims to prevent data corruption by malicious peers who distribute invalid pieces. Pieces are further sub-divided into fixed-sized blocks which are typically 16KB in size.
2. To advertise a file's availability for download with BT, a metadata file is created and published on the web. This metadata file contains a unique identifier called an info hash that is derived from the semantic description of the file, the cryptographic hashes of each piece, a link to a tracker server that helps to organize the peers, the number of pieces, and the piece size. Optionally, other information may be included.
3. Once the metadata file has been obtained, a new peer wishing to start sharing the file queries the tracker server to obtain a list of other peers who are currently sharing the file. Since the peer list may be arbitrarily large, the tracker typically replies with a fixed number of randomly selected peers. This also helps to balance the traffic load over the participating peers.
4. The peer requests specific parts of the file, in a non-sequential manner, addressed by piece number and offset.

The tracker functionality may be implemented as a centralized server, a distributed hash table, or a gossip-based peer discovery mechanism and uses a standard HTTP interface. By querying the tracker, a peer implicitly registers itself with the tracker's peer list and may subsequently receive requests from other peers for particular parts of the file that the peer has obtained. In BitTorrent's vernacular, peers who possess a full copy of the file being shared are called seeders and peer who are still downloading are called leechers. The general protocol behavior is shown in Fig 1.

Once a peer has completed a download, they may continue to participate in the protocol to help other peers download, or they may leave. BitTorrent does attempt to mitigate selfish peer behavior by incorporating a "tit-for-tat" mechanism into the piece request process. In essence, it attempts to promote fairness and reciprocity in the piece sharing.

BT has achieved excessive popularity for its ability to efficiently share files. However, it creates several important issues for network operators and copyright enforcement agencies:

Peers sharing files use a substantial amount of bandwidth in both down and upload directions due to the aggressive behavior of BT. On the other hand, broadband Internet service providers (ISPs) have an obligation to their customers for providing a reasonable QoS, even during the peak hours. Excessive BT usage on these networks

complicates effective network management. Consequently, ISPs from around the world have adopted various policies for throttling or explicitly blocking BT traffic.

The second issue with BT is the copyright enforcement concerns. Experience has shown that BT's decentralized nature has made it an attractive tool for users wishing to share copyright protected media content such as popular music, movies, and television programs. In the client/server model, copyright holders or law enforcement agencies may easily identify infringing content and request the hosting server to remove the content, or face a penalty defined by the local legal system. However, with a decentralized P2P protocol like BT, it is challenging for copyright holders or law enforcement agencies to contact each individual peer, since there could be millions of peers distributed. Definitively identifying peers who participate in illegal file sharing is a significant challenge.

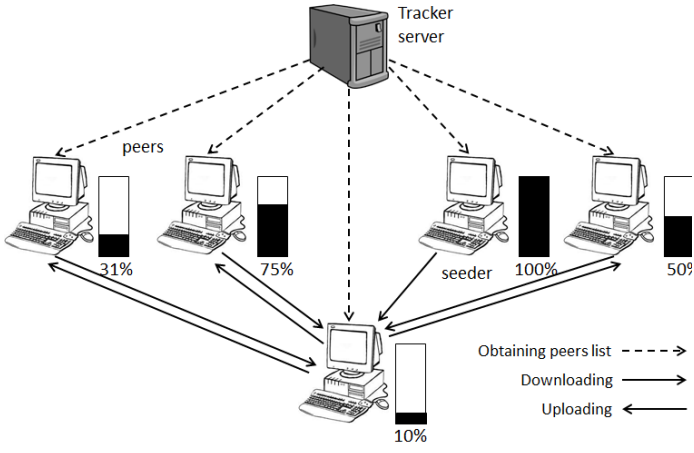


Fig. 1. File transfer with BT

4 Proposed P2P Identification System

Figure 2 shows the basic structure of the proposed P2P identification system. Details about each step are explained in the following sections.

4.1 Database

The database used in this research is captured from a local computer using WireShark tool. Wireshark is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education.

A total of 5 hours of traffic was collected while running a couple of BitTorrent and µTorrent clients for downloading video and music files. All other applications were closed during P2P application running. Another 2 hours of traffic was collected while running other non-P2P applications including file downloading, live streaming, web

browsing, email, and Skype. So the database is divided into two separate files: BitTorrent P2P traces and non-P2P traces.

4.2 Pre-filtering

In the pre-filtering stage, those packets corresponding to some protocols that we can confidently say are not used by P2P applications are removed from the dataset by setting proper filters in WireShark tool. By eliminating these packets we are aiming at improving the performance of the system both in terms of computational complexity and accuracy. List of filtered protocols were decided after consulting with experts in the field. Here is the list of protocols which were removed at this stage:

- Address resolution protocol (ARP)
- Internet group management protocol (IGMP)
- Internet control massage protocol (ICMP)
- NetBIOS name service (NBNS) protocol
- Common Unix printing system (cups)
- Dynamic host configuration protocol (DHCP)
- Dropbox LAN sync protocol
- Broadcast packets (i.e. packets with destination address 255.255.255.255)

4.3 Flow Conversion and Features Computation

Network monitoring solutions operate on the notion of network flows. A flow is defined as a series of packet exchanges between two hosts, identifiable by the 5-tuple (i.e. source address, source port, destination address, destination port, transport protocol). For applications running over TCP flow termination happens upon proper connection tear-down or by a flow timeout, whichever occurs first. UDP flows are terminated by a flow timeout. A 600 second flow timeout is considered in this work.

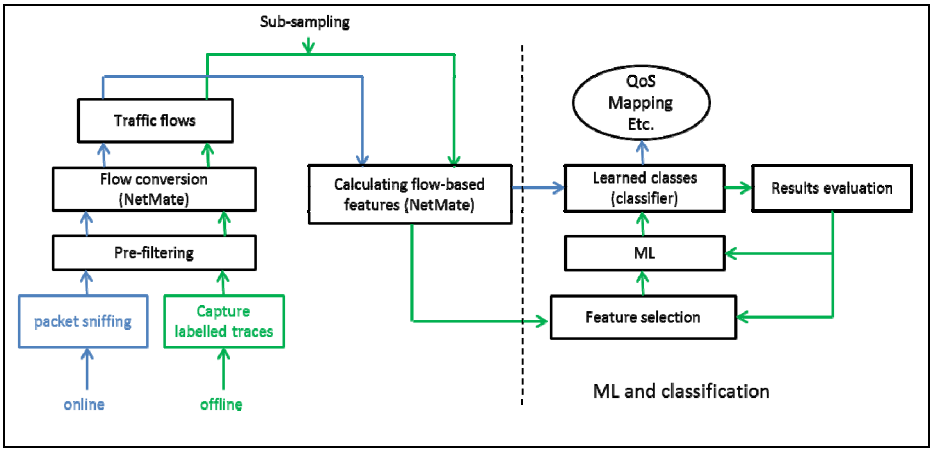


Fig. 2. The proposed ML-based P2P traffic identification system

Each flow is represented by a set of statistical features and associated feature values. A feature is a descriptive statistic that can be calculated from one or more packets. Thus, a flow can be thought of as a D^{th} dimensional vector, where D is the total number of calculated features and where each feature value in the vector represents a different statistic about the packets collected by that flow. NetMate (Network Measurement and Accounting System) was used to generate flows, and to compute feature values. NetMate is a flexible and extensible network measurement tool that can be used for accounting, delay/loss measurement, packet capturing and much more. The main advantage over other existing tools is that it can be easily extended due to its modular (class-based) structure and dynamic loadable packet processing and information export modules.

Table I shows the 30 features calculated by NetMate. Flows are bidirectional with the first packet determining the forward direction. Our system is completely independent of the payload data and features like IP addresses and source/destination port numbers to avoid problems concerned with this type of information described in section 2.

4.4 Feature Selection and Classification

The task of selecting the most relevant features in a classification task can be viewed as one of the fundamental problems in the field of machine learning. The performance, robustness, and usefulness of a classification algorithm are improved when relatively few features are involved in the classification. By selecting the most relevant subset from the original feature set, we can increase the performance of the classifier and at the same time decrease the computation cost.

In essence, the reduction of the original feature set to a smaller one preserving the relevant information while discarding the redundant ones is referred to as feature selection (FS). The aim is to achieve the minimum classification error rate. Many methods have been proposed in the literature for feature selection. In this work for FS we are using *minimum redundancy-maximum relevance* criteria which is based on maximum statistical dependency. This method is briefly explained in the following section.

Feature Selection Using Maximum Statistical Dependency Criterion

In feature selection using maximum statistical dependency criteria, minimal error is achieved by maximizing the statistical dependency of the target class c on the data distribution in the subspace (and vice versa). This scheme is called maximal dependency (Max Dependency) [16].

The success of a feature selection algorithm depends critically on how much information about the output class is contained in the selected features. Using Fano's inequality, the minimal probability of incorrect estimation P_e of class c using inputs $\vec{x}$ is lower bounded by

$$P_e \geq \frac{H(c \mid \vec{x}) - 1}{\log N} = \frac{H(c) - I(\vec{x}; c) - 1}{\log N} \tag{1}$$

Where H and I are entropy and mutual information respectively. Because the entropy of class, $H(c)$, and the number of classes N is fixed, the lower bound of P_e is minimized when $I(x;c)$ becomes the maximum. Thus, it is necessary for good feature selection methods to maximize the mutual information $I(x;c)$.

In other words, the purpose of feature selection is to find a feature set S with m features $\{x_i\}$, which jointly have the largest dependency on the target class c:

$$\max D(S,c), D = I(\{x_i, i = 1,...,m\};c) \tag{2}$$

But since finding the joint probability in equation (2) is not easy, an alternative criterion could be maximal relevance which is defined as:

$$\max D(S,c), D = \frac{1}{|S|}\sum_{x_i \in S} I(x_i;c) \tag{3}$$

which approximates equation 2 with mean value of all mutual information values between individual feature x_i and class c.

However, there might still exit redundant features in the selected feature subset. When two features are redundant, they highly depend on each other and as a result the respective class-discriminative power would not change if one of them were removed [17]. The minimum redundancy is defined as:

$$\min R(s), R = \frac{1}{|S|^2}\sum_{x_i, x_j \in S} I(x_i;x_j) \tag{4}$$

The two criteria defined in equation 3 and equation 4 can be combined as a maximum-relevant minimum-redundancy (MRMR) criterion. The combined criterion is defined as:

$$\max \phi(D,R), \phi = D - R \tag{5}$$

Classification

Machine learning methodology is an artificial intelligence approach to establish and train a model to recognize the pattern or underlying mapping of a system based on a set of training examples consisting of input and output patterns. There are two phases in ML algorithms: *learning* or *training* the system with known data and *testing* where the system performance is tested with new data.

In this work Least Squares SVMs (LS-SVMs) is used as a classifier. A brief overview of SVMs is presented in the next section.

5 Support Vector Machines

SVM was introduced first by Vapnik and co-workers [19], and it is such a powerful classification method that in the few years since its introduction has outperformed most other classifiers in a wide variety of applications. SVM is used in applications of regression and classification; however, it is mostly used as a binary classifier. SVM is based on the principle of structural risk minimization. The optimal boundary is found in such a way that maximizes the margin between two classes of data points.

SVM is based on kernel functions, which are used to map data points to a higher dimensional feature space in order to be linearly separable. The optimization problem here is the dual optimization problem which is solved by Lagrangian method and making use of very important Karush-Kuhn-Tucker (KKT) conditions. Equation 6 shows the dual optimization problem for SVM classifiers:

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i \, y_j \, \alpha_i \, \alpha_j \, K(\vec{x}_i . \vec{x}_j) \tag{6}$$

subject to constraints

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \qquad \alpha_i \geq 0, \; i = 1, ..., n$$

where $\alpha_i \geq 0$ are the Lagrange multipliers and $K(\vec{x}_i . \vec{x}_j)$ is the kernel function. Among the different kernel functions, the most common kernels are polynomial, Gaussian Radial Basis function (RBF) and multi-layer perception (MLP).

The final decision rule can be expressed as:

$$f(\vec{x}, \vec{\alpha}^*, b_o) = \sum_{i=1}^{N_{sv}} y_i \alpha_i^* \, K(\vec{x}_i . \vec{x}) + b_o \tag{7}$$

where N_{sv} and $\vec{\alpha}_i^*$ denote the number of support vectors and the non-zero Lagrange multipliers corresponding to the support vectors respectively.

In this work Least Squares Support Vector Machines (LS-SVMs) are used. LS-SVMs are reformulations to the original SVMs which lead to solving linear KKT systems. In LS-SVMs the inequality constraints in SVM are replaced with equality constraints. As a result the solution follows from solving a set of linear equations instead of a quadratic programming problem which we have in original SVM formulation of Vapkin, and obviously we can have a faster algorithm.

The primal problem of the LS-SVMs is defined as:

$$\min_{\vec{w},b} \qquad _{w,b,e} J_p(\vec{w}, b, \vec{e}) = 1/2 \left\| \vec{w} \right\|^2 + \gamma 1/2 \sum_{i=1}^{d} e_i^2 \tag{8}$$

Subject to

$$y_i [\vec{w}^T \phi(x_i) + b] = 1 - e_i, \;\; i = 1, ..., d$$

Table 1. List of calculated features by NetMate

List of Calculated Features	
1. protocol (TCP, UDP)	16. max forward inter arrival time
2. total forward packets	17. std dev forward inter arrival time
3. total forward volume	18. min backward inter arrival time
4. total backward packets	19. mean backward inter arrival time
5. total backward volume	20. max backward inter arrival time
6. min forward packet length	21. std dev backward inter arrival time
7. mean forward packet length	22. duration of the flow
8. max forward packet length	23. min active time
9. std dev forward packet length	24. mean active time
10. min backward packet length	25. max active time
11. mean backward packet length	26. std dev active time
12. max backward packet length	27. min idle time
13. std dev backward packet length	28. mean idle time
14. min forward inter arrival time	29. max idle time
15. mean forward inter arrival time	30. std dev idle time

where γ is a parameter analogous to SVM's regularization parameter (C).

The main characteristic of LS-SVMs is the low computational complexity comparing to SVMs without quality loss in the solution.

6 Experimental Results

MRMR feature selection algorithm sorts out the features based on the criteria given in Equation 5. Order of features is given in Table 2. LS-SVM is used in order to find the first *m* features in the set which gives the best result in terms of classification accuracy. As Fig 3 illustrates, the first 19 features presented in Table 2 give us the best identification rate. After that the accuracy starts to drop again.

Figure 4 shows the histograms for some of the features, where the blue line corresponds to BT traffic and the green line to non-P2P traffic. As it can be seen for forward packet length non-P2P traffic flows expand over higher values compared to BT flows. For backward packet length, on the other hand, BT traffic has higher values. This could be justified since in BT architecture the actual data is going in both direction among peers whereas in the client/server scheme the actual data is going from server to client mostly. Also as part (e) demonstrates, BT traffic is mostly running over UDP while TCP is used for the mail part of non-P2P applications.

The selected features by the feature selection unit are fed to a LS-SVM with RBF and linear kernel functions. For the purpose of comparison study linear discriminant classifier with perceptron criterion function, k-nearest neighbors (k-NNs) classifier and K-means clustering algorithm were also applied to the data set.

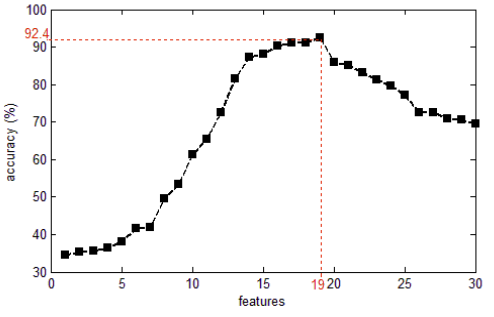


Fig. 3. Feature selection procedure using MRMR criteria

Table 2. Order of features selected based on MRMR criteria

Order of features by MRMR criteria	
1. total forward volume	16. mean idle time
2. total backward packets	17. protocol (TCP, UDP)
3. total backward volume	18. total forward packets
4. mean forward packet length	19. min forward packet length
5. max forward packet length	20. min backward packet length
6. std dev forward packet length	21. mean backward packet length
7. std dev active time	22. std dev backward packet length
8. min active time	23. max forward inter arrival time
9. mean active time	24. max backward packet length
10. max active time	25. min idle time
11. duration of the flow	26. mean backward inter arrival time
12. mean forward inter arrival time	27. min backward inter arrival time
13. std dev forward inter arrival time	28. min forward inter arrival time
14. max idle time	29. max backward inter arrival time
15. std dev idle time	30. std dev backward inter arrival time

Table 3. Error rate and training time achieved by different classifiers for overall BT classification

	Classification Method				
	SVM linear kernel	SVM RBF kernel	LDA	k-NNs	K-means
Error rate	20.9%	**3.7%**	36.1%	10.2%	45.4%
Training time (sec)	25.6	31.8	4.5	No training!	No training!

Table 4. Comparision of ML approach and DPI for encrypted and non-encrypted P2P traffic

Application	BT-movie downloading not encrypted	µT-movie downloading encrypted
Recognition rate by ML	91.6%	94.5%
Recognition rate by DPI	92.3%	2.4%

Table 3 reports the error rates and training times achieved by different classifiers. As it shows, the minimum error rate of 3.7% is obtained by SVM with RBF kernel function. Using a linear kernel degrades the performance to 20.9% error rate but it is slightly faster than SVM with RBF kernel. Linear discrimant classifier does not perform very well in terms of accuracy but compared to SVM the training time is shorter. Another drawback with this method is that it is dependent on the initial values and there is always the problem of local minimas. k-NN algorithm does not involve any training phase and it is a very simple algorithm. The error rate obtained by this method is 10.2%. An obvious disadvantage of the k-NN method is the time complexity of making predictions when number of training samples is very large. Suppose that there are n training examples in R^m. Then applying the k-NN method to one test example requires $O(nm)$ time, compared to just $O(m)$ time to apply a linear classifier such as a perceptron. K-means algorithm is a very fast and easy to implement clustering algorithm because of simplicity of the algorithm. However, for rather high-dimensional problems such as this one, it does not yield a satisfying result. Also with different initial values different results are obtained.All the parameters, such as kernel function parameters, learning rate, and number of nearest neighbors, were adjusted empirically. The best overall accuracy of 93.6% was achieved by SVM with RBF kernel which is a very promising result. Table 4 compares the recognition results for BT traffic achieved by our ML-based system proposed in this paper using SVM with RBF kernel and DPI method. For the traffic which is not encrypted the performance of DPI method is accurate. However, for encrypted traffic DPI completely fails since it is not possible to examine the payload for finding the string patterns. Our method on the other hand performs very well because it is totally independent of payload. The encrypted traffic was collected using µTorrent (µT) which is a client of BT. µT program is designed to use minimal computer resources while offering functionality comparable to larger BT clients and is mostly encrypted.

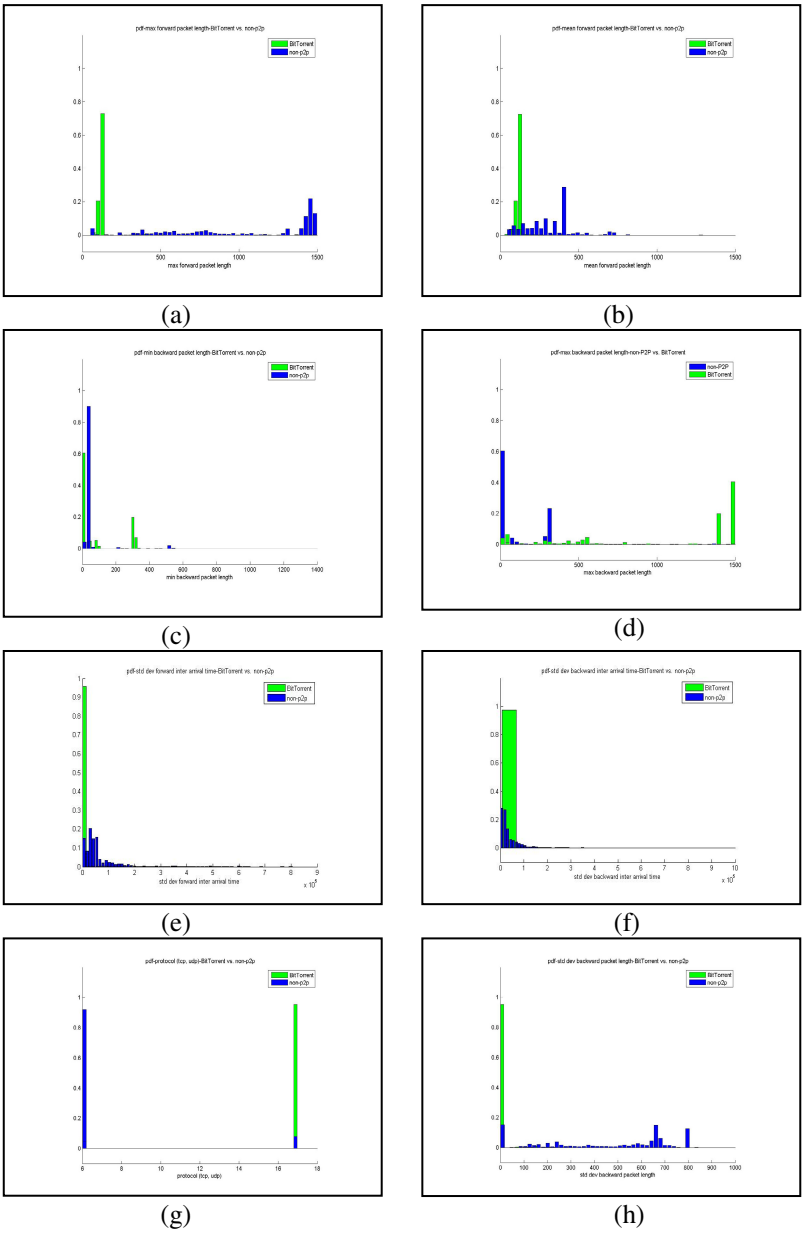


Fig. 4. PDF's for different features. In each graph the green bars show the BT's PDF and the blue bars show the non-P2P's PDF. (a) max forward packet length (b) mean forward packet length (c) mix backward packet length (d) max backward packet length (e) std forward inter-arrival time (f) std backward inter-arrival time (g) protocol (TCP/UDP) (h) std backward length.

7 Conclusion

In this work first we proposed a system to identify the BT traffic, which is one of the most popular and problematic P2P applications, based on ML methodology. The proposed system is independent of both IP addresses and payload information in order to evade the issues concerned with these two methods explained in Section 2. Our system is merely based on some statistical values calculated from traffic flows and LS-SVMs. Thirty flow-based statistical features are calculated from the flows and the most effective subset of feature is identified using feature selection algorithm. The overall accuracy of 93.6% was achieved through this approach. Our experimental results show that the most effective features for identifying BT traffic are forward packet lengths, backward volume, backward packet lengths, forward inter-arrival time, idle and active time and their variations. Our system shows a superior performance where DPI method fails for recognition of encrypted traffic.

References

1. http://www.ipoque.com/en/resources/internet-studies
2. Dreger, H., Feldmann, A., Mai, M., Paxson, V., Sommer, R.: Dynamic Application-Layer Protocol Analysis for Network Intrusion Detection. In: USENIX Security Symposium, pp. 257–272 (2006)
3. Szabo, G., Szabo, I., Orincsay, D.: Accurate Traffic Classification. In: IEEE International Symposium on World of Wireless Mobile and Multimedia Networks, pp. 1–8 (2007)
4. Zhou, L., Wang, X., Tu, W., Mutean, G., Geller, B.: Distributed scheduling scheme for video streaming over multi-channel multi-radio multi-hop wireless networks. IEEE Journal on Selected Areas in Communications 28, 409–419 (2010)
5. Constantinou, F., Mavrommatis, P.: Identifying Known and Unknown Peer-to-Peer Traffic. In: Fifth IEEE International Symposium on Network Computing and Applications, pp. 93–102 (2006)
6. Moore, A.W., Papagiannaki, K.: Toward the Accurate Identification of Network Applications. In: Dovrolis, C. (ed.) PAM 2005. LNCS, vol. 3431, pp. 41–54. Springer, Heidelberg (2005)
7. Patrick, H., et al.: ACAS: Automated Construction of Application Signatures. In: ACM SIGCOMM Workshop on Mining Network Data, Philadelphia, Pennsylvania, pp. 197–202 (2005)
8. Sen, S., Spatscheck, O., Wang, D.: Accurate, Scalable In-Network Identification of p2p Traffic Using Application Signatures. In: 13th International Conference on World Wide Web, New York, USA, pp. 512–521 (2004)
9. Roughan, M., Sen, S., Spatscheck, O.: Class ofService Mapping for QoS: A Statistical Signature-Sased Approach to IP Traffic Classification. In: 4th ACM SIGCOMM Conference on Internet Measurement, Sicily, Italy, pp. 135–148 (2004)
10. Williams, N., Zander, S., Armitage, G.: A Preliminary Performance Comparison of Five Machine Learning Algorithms for IP Flow Classification. ACM SIGCOMM Computer Communication Review 36, 5–16 (2006)
11. Dusi, M., Gringoli, F., Salgarelli, L.: IP traffic classification for QoS Guarantees: the Independece of Packets. In: 17th International Conference on Computer Communications and Networks, pp. 1–8 (2008)

12. Mellia, M., Pescapè, A., Salgarelli, L.: Traffic Classification and Its Applications to Modern Networks. Computer Networks, 759–760 (2009)
13. Frank, J.: Machine Learning and Intrusion Detection: Current and Future Directions. In: 17th Computer Security Conference (1994)
14. McGregor, A., Hall, M., Lorier, P., Brunskill, J.: Flow Clustering Using Machine Learning Techniques. In: Barakat, C., Pratt, I. (eds.) PAM 2004. LNCS, vol. 3015, pp. 205–214. Springer, Heidelberg (2004)
15. Zander, S., Nguyen, T., Armitage, G.: Automated Traffic Classification and Application Identification Using Machine Learning. In: IEEE Conference on Local Computer Networks, pp. 250–257 (2005)
16. Arulampalam, G., Ramakonar, V., Bouzerdoum, A., Habibi, D.: Classification of Digital Modulation Schemes Using Neural Networks. In: Fifth International Symposium on Signal Processing and Its Applications, vol. 2, pp. 649–652 (1999)
17. Cristianini, N., Taylor, J.S.H.: An Introduction to Support Vector Machines and Other Kernel-based Methods. Cambridge University Press, United Kingdom (2000)
18. Nguyen, T., Armitage, G.: A survey of techniques for internet traffic classification using machine learning. In: IEEE Comm. Surv. & Tutor, pp. 56–76 (2008)
19. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Berlin (1995)

Application of Bagging, Boosting and Stacking to Intrusion Detection

Iwan Syarif [1,2], Ed Zaluska[1], Adam Prugel-Bennett[1], and Gary Wills[1]

[1] School of Electronics and Computer Science, University of Southampton, UK
{is1e08,ejz,apb,gbw}@ecs.soton.ac.uk
[2] Eletronics Engineering Polytechnics Institute of Surabaya, Indonesia
iwanarif@eepis-its.edu

Abstract. This paper investigates the possibility of using ensemble algorithms to improve the performance of network intrusion detection systems. We use an ensemble of three different methods, bagging, boosting and stacking, in order to improve the accuracy and reduce the false positive rate. We use four different data mining algorithms, naïve bayes, J48 (decision tree), JRip (rule induction) and iBK(nearest neighbour), as base classifiers for those ensemble methods. Our experiment shows that the prototype which implements four base classifiers and three ensemble algorithms achieves an accuracy of more than 99% in detecting known intrusions, but failed to detect novel intrusions with the accuracy rates of around just 60%. The use of bagging, boosting and stacking is unable to significantly improve the accuracy. Stacking is the only method that was able to reduce the false positive rate by a significantly high amount (46.84%); unfortunately, this method has the longest execution time and so is inefficient to implement in the intrusion detection field.

Keywords: Intrusion detection system, bagging, boosting, stacking, ensemble classifiers.

1 Intrusion Detection System

Intrusion detection is a process of gathering intrusion-related knowledge occurring in the process of monitoring events and analyzing them for signs of intrusion [1]. There are two basic IDS approaches: misuse detection (signature-based) and anomaly detection. The misuse detection system uses patterns of well-known attacks to match and identify known intrusions. It performs pattern matching between the captured network traffic and attack signatures. If a match is detected, the system generates an alarm. The main advantage of the signature detection paradigm is that it can accurately detect instances of known attacks. The main disadvantage is that it lacks the ability to detect new intrusions or zero-day attacks [16][17].

The anomaly detection model works by identifying an attack by looking for behaviour that is out of the normal. It establishes a baseline model of behaviour for users and components in a computer or network. Deviations from the baseline cause alerts that direct the attention of human operators to the anomalies [17][18]. This system searches for anomalies either in stored data or in the system activity. The main advantage of anomaly detection is that it does not require prior knowledge of an intrusion

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 593–602, 2012.

and thus can detect new intrusions. The main disadvantage is that it may not be able to describe what constitutes an attack and may have a high false positive rate [16][17][18]. We will develop a hybrid IDS which combines both misuse detection and anomaly detection system, but this paper focuses on the first technique.

2 Data Mining for IDS

Data mining studies automatic techniques for learning to make accurate predictions based on past observations [2]. In the intrusion detection case, data mining can be used to build a system that can distinguish intrusions or anomalies from normal network traffic. To build this kind of system, the first step is for the machine learning algorithms to learn the training dataset, which contains both normal traffic and intrusions. This learning phase results in a model that can be used to determine whether the network traffic is normal or an intrusion. There are many possible algorithms that can be used in the intrusion detection problem; their performance is measured using accuracy rate and false positive rate. In order to achieve a higher accuracy and lower false positive rate, many data mining researchers have proposed various ensemble learning approaches. It is well known in the data mining literature that the appropriate combination of a number of weak classifiers can yield a highly accurate global classifier [1].

3 Ensemble Classifier

An ensemble classifier is a method which uses or combines multiple classifiers to improve robustness as well as to achieve an improved classification performance from any of the constituent classifiers. Furthermore, this technique is more resilient to noise compared to the use of a single classifier. This method uses a 'divide and conquer approach' where a complex problem is decomposed into multiple sub-problems that are easier to understand and solve.

Ensemble approaches [2][15] have the advantage that they can be made to adapt to any changes in the monitored data stream more accurately than single model techniques. An ensemble classifier has better accuracy than single classification techniques. The success of the ensemble approach depends on the diversity in the individual classifiers with respect to misclassified instances [3]. According to Polikar [4], there are four ways to achieve this diversity, the first is to use different training data to train single classifiers, the second is to use different training parameters, the third is to use different features to train the classifiers and the final one is to combine different types of classifier.

Dietterich [5] reported that there are three main reasons why an ensemble classifier is usually significantly better than a single classifier. Firstly, the training data does not always provide sufficient information for selecting a single accurate hypothesis. Secondly, the learning processes of the weak classifier might be imperfect, and thirdly, the hypothesis space being searched might not contain the true target function while an ensemble classifier can provide a good approximation.

In this paper we evaluated and analyzed three different ensemble classifier techniques, called bagging, boosting and stacking, using various weak classifiers, such as nearest neighbour, decision tree, rule induction and naïve bayes; these were applied on a network intrusion dataset [11][12][13].

3.1 Bagging

Bagging, which means bootstrap aggregation, is one of the simplest but most successful ensemble methods for improving unstable classification problems. For example, weak classifiers, such as decision tree algorithms, can be unstable, especially when the position of a training point changes slightly and can lead to a very different tree. This method is usually applied to decision tree algorithms, but it also can be used with other classification algorithms such as naïve bayes, nearest neighbour, rule induction, etc. The bagging technique is very useful for large and high-dimensional data, such as intrusion data sets, where finding a good model or classifier that can work in one step is impossible because of the complexity and scale of the problem.

Bagging was first introduced by Leo Breiman [6] to reduce the variance of a predictor. It uses multiple versions of a training set which is generated by a random draw with the replacement of N examples where N is the size of original training set. Each of these data sets is used to train a different model. The outputs of the models are combined by voting to create a single output. Details of the bagging algorithm and its pseudo-code were given in [10].

3.2 Boosting

Boosting, which was introduced by Schapire et al.[7], is an ensemble method for boosting the performance of a set of weak classifiers into a strong classifier. This technique can be viewed as a model averaging method and it was originally designed for classification, but it can also be applied to regression. Boosting provides sequential learning of the predictors. The first one learns from the whole data set, while the following learns from training sets based on the performance of the previous one. The misclassified examples are marked and their weights increased so they will have a higher probability of appearing in the training set of the next predictor. It results in different machines being specialized in predicting different areas of the dataset [8].

In this paper, we select an AdaBoost algorithm, which is one of the most widely used boosting techniques for constructing a strong classifier as a linear combination of weak classifiers. The AdaBoost algorithm was first introduced by Freund and Schapire [9] and has been shown to solve many of the practical difficulties of earlier boosting algorithms, since it has solid theoretical foundation and produces very accurate predictions. Details of the boosting algorithm and its pseudo-code were given in [10].

3.3 Stacking

Stacking or *stacked generalization*, is a different technique of combining multiple classifiers. Unlike bagging and boosting, stacking is usually used to combine various different classifiers, e.g. decision tree, neural network, rule induction, naïve bayes, logistic regression, etc. Stacking consists of two levels which are base learner as level-0 and stacking model learner as level-1. Base learner (level-0) uses many different models to learn from a dataset. The outputs of each of the models are collected to create a new dataset. In the new dataset, each instance is related to the real value that

it is suppose to predict. Then that dataset is used by stacking model learner (level-1) to provide the final output [8]. For example, the predicted classifications from the three base classifiers, naïve bayes, decision tree and rule induction can be used as input variables into a nearest neighbour classifier as a stacking model learner, which will attempt to learn from the data how to combine the predictions from the different models to achieve the best classification accuracy. Details of the boosting algorithm and its pseudo-code were given in [10].

4 Experimental Settings

The following section describes the intrusion data sets used in the experiment, the performance metric used to evaluate the proposed system and the experimental settings and its results.

4.1 Intrusion Dataset

One of the most widely used data sets for evaluating intrusion detection systems (IDS) is the DARPA/Lincoln Laboratory off-line evaluation dataset or IDEVAL [11]. IDEVAL is the most comprehensive testset available today and it was used to develop the 1999 KDD Cup data mining competition [12]. In this experiment, we use the NSL-KDD intrusion data, which was provided to solve some problems in KDD'99, particularly that its training and test sets contained a huge number of redundant records with about 78% and 75% of the records being duplicated in the training and test sets, respectively. This may cause the classification algorithms to be biased towards these redundant records and thus prevent it from classifying other records [13].

Table 1. List of intrusions in training and testing data

Intrusions which exist in both training and testing data	Intrusions which only exist in testing data
back, buffer_overflow, ftp_write, guess_passwd, imap, ipsweep, land, loadmodule, multihop, neptune, nmap, phf, pod, portsweep, rootkit, satan, smurf, spy, teardrop, warezclient, warezmaster	apache2, httptunnel, mailbomb, mscan, named, perl, processtable, ps, saint, sendmail, snmpgetattack, snmpguess, sqlattack, udpstorm, worm, xlock, xsnoop, xterm

The intrusion data set consists of forty different intrusions classified into four main categories: DoS (Denial of Service), R2L (Remote to Local Attack), U2R (User to Root Attack) and Probing Attack. The training dataset consists of 25,191 instances and the testing dataset consists of 11,950 instances. The testing data set has many intrusions which do not exist in the training data, as shown in table 1.

4.2 Performance Metric

We use accuracy rate and false positive rate as the performance criteria based on the following metric shown in Table 2 below.

Table 2. Performance metric

		Actual Result	
		Intrusion	Normal
Predicted Result	Intrusion	True Positive (TP)	False Positive (FP)
	Normal	False Negative (FN)	True Negative (TN)

True Positive (TP) is a condition when an actual attack is successfully detected by the IDS and True Negative (TN) is a condition when no attack has taken place and no IDS alert is raised. False Positive (FP) is an alarm/alert that indicates that an attack is in progress when in fact there was no such attack. False Negative (FN) is a failure of IDS to detect an actual attack [19]. The accuracy rate and false positive rate are measured using these following formulas:

$$Accuracy\ rate = \frac{\text{TP+FN}}{\text{TP+TN+FP+FN}}(1), False\ Positive = \frac{FP}{\text{TP+FP}}(2)$$

4.3 Experimental Settings

We apply various data mining algorithms in the misuse detection module in order to find the best method for detecting intrusion based on accuracy, false positives and speed (computation time). We use four single algorithms from the Weka Data Mining Tools: Naïve Bayes, iBK, Jrip and J48, then apply these algorithms into three different ensemble classifiers, which are bagging, boosting and stacking, as shown in Figure 1 below.

These algorithms were executed on a PC with Intel Xeon quad core processors 2.67 GHz and 12 Gb RAM. In the first experiment, we use 10-fold cross validation as a performance measurement while in the second experiment we use testing data which contains many new intrusions.

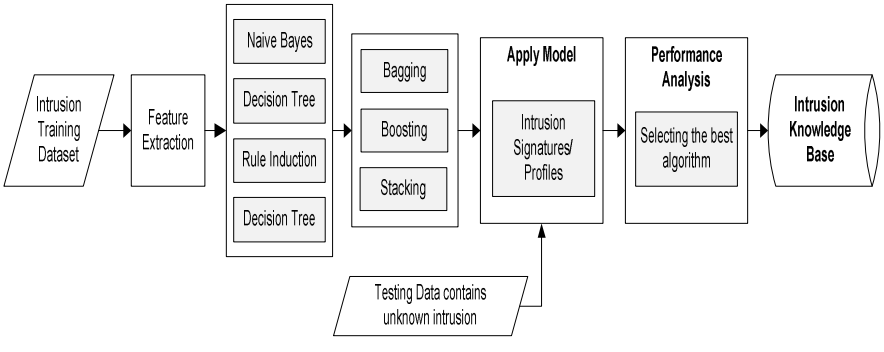


Fig. 1. Misuse detection model

4.3.1. Cross Validation

For performance measurement, we first use the 10-fold cross validation technique, which only needs training data. In 10-fold cross-validation, the original training data

is randomly partitioned into 10 subsamples. Of the 10 subsamples, a single subsample is retained as the validation data for testing the model, and the remaining 9 subsamples are used as training data. The cross-validation process is then repeated 10 times with each of the 10 subsamples used exactly once as the validation data. The 10 results from the folds then can be averaged to produce a single estimate. The results of the first experiment are given in Tables 3 and 4 below.

Table 3. The performance of ensemble classifiers using 10-fold cross validation

Algorithm	Accuracy			False Positive		
	Single	Bagging	Boosting	Single	Bagging	Boosting
Naïve Bayes	89.59%	89.57%	94.56%	10.60%	10.70%	5.30%
iBK	99.44%	99.44%	99.44%	0.60%	0.60%	0.60%
Jrip	99.58%	99.71%	99.73%	0.40%	0%	0.30%
J48	99.56%	99.67%	99.80%	0.40%	0.30%	0.20%

In the stacking method, we use three different algorithms as base learners and an algorithm as a stacking model learner. We use various combinations of naïve bayes, iBK, J48 and JRip. The classifications predicted by the base learners will be used as input variables into a stacking model learner. Each input classifier computes predicted classifications using cross validation from which overall performance characteristic can be computed. Then the stacking model learner will attempt to learn from the data how to combine the predictions from the different models to achieve maximum classification accuracy. The stacking algorithm experiment results are given in the Table 4.

Table 4. The performance of stacking algorithm using 10-fold cross validation

Base Learner	Stacking Model Learner	Accuracy (%)	False Positive (%)
Naïve Bayes	Jrip	99.64%	0.40%
iBK			
J48			
Jrip	Naïve Bayes	99.75%	0.30%
iBK			
J48			
Naïve Bayes	iBK	99.51%	0.50%
J48			
Jrip			
Naïve Bayes	J48	99.63%	0.40%
iBK			
Jrip			

4.3.1.1. Results. Overall, all the algorithms achieved good results, with the highest accuracy being 99.80% and the lowest being 89.59%. Tables 3 and 4 above show that Adaboost when implement with J48 as a weak classifier achieves the highest accuracy, which is 99.80%, with a false positive (FP) rate of 0.30%. On the other hand, the J48 Bagging algorithm achieves the lowest FP rate of 0%. Unfortunately the computation time of the three ensemble classifiers are all very high; the slowest one is stacking followed in turn by boosting and bagging.

Table 5. Accuracy improvement on 10 fold cross validation experiment

Algorithm	Single Classifier	Accuracy Improvement					
		Bagging	%	Boosting	%	Stacking	%
Naïve Bayes	89.59%	89.57%	-0.02%	94.56%	5.55%	99.75%	11.34%
iBK	99.44%	99.44%	0.00%	99.44%	0.00%	99.51%	0.07%
Jrip	99.58%	99.71%	0.13%	99.73%	0.15%	99.64%	0.06%
J48	99.56%	99.67%	0.11%	99.80%	0.24%	99.63%	0.07%

Table 5 and Table 6 show that the use of the bagging, boosting and stacking algorithms did not improve the accuracy significantly. Only the use of boosting and stacking on the Naïve Bayes algorithm were able to improve the accuracy, by 5.55% and 11.22% respectively, while the others showed a less than 1% improvement.

Table 6. False positive reduction on 10 fold cross validation experiment

Algorithm	Single Classifier	False Positive Improvement					
		Bagging	%	Boosting	%	Stacking	%
Naïve Bayes	10.60%	10.70%	-0.94%	5.30%	50.00%	0.30%	97.17%
iBK	0.60%	0.60%	0.00%	0.60%	0.00%	0.50%	16.67%
Jrip	0.40%	0.30%	25.00%	0.30%	25.00%	0.40%	0.00%
J48	0.40%	0.30%	25.00%	0.20%	50.00%	0.40%	0.00%

While the three ensemble algorithms failed to improve the accuracy, they succeed in reducing the false positive rates. Bagging was able to reduce the false positive rate by up to 25% when implemented with Jrip and J48, boosting by up to 50% for Naïve Bayes and J48, and stacking by up to 96.23% for Naïve Bayes.

4.3.2. Testing Data

In the second stage, we implement various single algorithms against the training data set to build an intrusion model then apply this model to the testing data which contains a lot of unknown attacks (see Table 1). The results are given in Tables 7 and 8 below.

4.3.2.1. Results. Overall none of the algorithms in the misuse detection module performed very well in detecting data with a lot of new intrusions. The best accuracy was only 67.90%, which was achieved by the stacking algorithm with iBK as a model learner and three other algorithms (Naïve Bayes, Jrip and J48) as base classifiers. Bagging was only able to improve it by less than 1% in three methods (Naïve Bayes, iBK, J48) while boosting failed to improve any method. The stacking method was able to improve the accuracy to 6.90% (Naïve Bayes) and 8.05% (iBK).

Table 7. Accuracy improvement using testing data experiment

Algorithm	Single Classifier	Accuracy Improvement					
		Bagging	%	Boosting	%	Stacking	%
Naïve Bayes	55.77%	56.10%	0.59%	37.60%	-32.58%	59.62%	6.90%
iBK	62.84%	62.95%	0.18%	20.90%	-66.74%	67.90%	8.05%
Jrip	63.69%	59.40%	-6.74%	18.40%	-71.11%	64.31%	0.97%
J48	63.97%	64.51%	0.84%	18.80%	-70.61%	61.23%	-4.28%

The bagging algorithm failed to reduce the false positive rates in three base classifiers (Naïve Bayes, iBK, JRip) and was only able to reduce it by 1.12% with J48 as a base classifier. Boosting is worse than bagging because it failed to reduce the false positive rates on all four base classifiers.

Table 8. False positive reduction using testing data experiment

Algorithm	Single Classifier	False Positive Improvement					
		Bagging	%	Boosting	%	Stacking	%
Naïve Bayes	34.80%	35.10%	-0.86%	37.60%	-8.05%	18.50%	46.84%
iBK	20.90%	20.90%	0.00%	20.90%	0.00%	17.40%	16.75%
Jrip	18.00%	19.00%	-5.56%	18.40%	-2.22%	16.90%	6.11%
J48	17.90%	17.70%	1.12%	18.80%	-5.03%	19.60%	-9.50%

Stacking algorithm is the only approach which was able to reduce the false positive rates significantly, with a 46.84% reduction on Naïve Bayes, a 16.75% reduction on iBK and a 6.11% reduction on JRip, even though it failed on J48 (-9.50%).

Figure 2 shows that the use of bagging, boosting and stacking significantly increases the execution time. The slowest is stacking followed in turn by bagging and boosting. The stacking method was able to reduce the false positive rate, but it would be too slow to implement in a misuse detection module. The bagging method, especially when applied to the iBK and Naïve Bayes algorithms, did not increase the execution time significantly and only improves the accuracy by 0.18% (iBK) and 0.59% (Naïve Bayes). Furthermore, bagging failed to reduce the false positive rate in either algorithm.

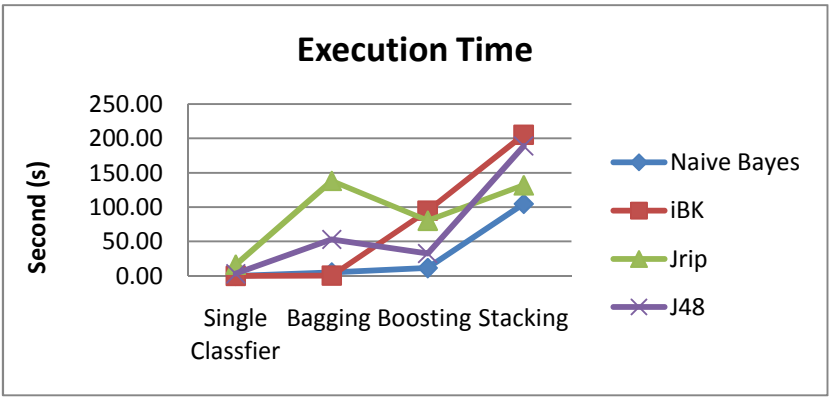


Fig. 2. Execution time comparison for single classifier bagging, boosting and stacking

5 Conclusions

We investigated the possibility of using ensemble algorithms (bagging, boosting and stacking) to improve the performance on network intrusion detection systems. Our experiment shows that a misuse detection module which implements four base classifiers and three ensemble algorithms achieves an accuracy of more than 99% in detecting known intrusions, but failed to detect novel intrusions with the accuracy rates of around just 60%. The use of bagging, boosting and stacking is unable to significantly improve the accuracy. Stacking is the only method that was able to reduce the false positive rate by a relatively high amount; unfortunately, this method has the longest execution time which is a serious disadvantage in the intrusion detection field. Of the four single classifiers used, J48 outperformed the three other methods by achieving the highest accuracy rates and the lowest false positive rate, with a relatively fast execution time. To improve the ability to detect new intrusions, we propose to develop an anomaly detection module and integrate both systems to produce a hybrid intrusion detection system.

References

1. Gudadhe, M., Prasad, P., Wankhade, K.: A new data mining based network intrusion detection model. In: International Conference on Computer & Communication Technology (ICCCT 2010), pp. 731–735 (2010)
2. Schapire, R.A.: The Boosting Approach to Machine Learning An Overview. In: Nonlinear Estimation and Classification. Springer (2003)
3. Lee, K.C., Cho, H.: Performance of Ensemble Classifier for Location Prediction Task: Emphasis on Markov Blanket Perspective. International Journal of u- and e- Service, Science and Technology 3(3) (September 2010)
4. Polikar, R.: Ensemble Based Systems in Decision Making. IEEE Circuits and Systems Magazine 6(3) (2006)

5. Dietterich, T.G.: Machine learning research: Four current directions. AI Magazine 18(4), 97–136 (1997)
6. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
7. Schapire, R.E., Freund, Y., Bartlett, P., Lee, W.S.: Boosting the margin: A new explanation for the effectiveness of voting methods. The Annals of Statistics 26(5), 1651–1686 (1998)
8. Graczyk, M., Lasota, T., Trawiński, B., Trawiński, K.: Comparison of Bagging, Boosting and Stacking Ensembles Applied to Real Estate Appraisal. In: Nguyen, N.T., Le, M.T., Świątek, J. (eds.) ACIIDS 2010, Part II. LNCS, vol. 5991, pp. 340–350. Springer, Heidelberg (2010)
9. Freund, Y., Schapire, R.E.: A Decision-Theoritic Generalization of on-line Learning and an Application to Boosting (1995)
10. Zhou, Z.-H.: Ensemble Learning. In: Encyclopedia of Biometrics, vol. 1, pp. 270–273. Springer, Berlin (2009) ISBN: 978-0-387-73002-8
11. DARPA Intrusion Detection Data Sets, http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html
12. KDD Cup 1999 Intrusion Data Sets, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
13. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.: A Detailed Analysis of the KDD CUP 99 Data Set. In: Second IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA (2009)
14. Dong, L., Yuan, Y., Cai, Y.: Using Bagging Classifiers to Predict Protein Domain Structural Class. Journal of Biomolecular Structure & Dynamics 24(3) (2006) ISSN 0739-1102
15. Dong, Y.S., Han, K.S.: A comparison of several ensemble methods for text categorization. In: The 2004 IEEE International Conference on Service Computing (SCC 2004), pp. 419–422. IEEE Computer Society, Washington DC (2004) ISBN:0-7695-2225-4
16. Panda, M., Patra, M.R.: Ensemble of Classifiers for Detecting Network Intrusion. In: International Conference on Advances in Computing, Communication and Control (ICAC3 2009), pp. 510–515 (2009)
17. Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G., Vazquez, E.: Anomaly-based network intrusion detection: Techniques, systems and challenges. Computer & Security 28(1-2), 18–28 (2009)
18. Davis, J.J., Clark, A.J.: Data preprocessing for anomaly based network intrusion detection: A review. Computer & Security 30(6-7), 353–375 (2011)
19. Whitman, M.E., Mattord, H.J.: Principles of Information Security, 4th edn. Course Technology (2011) ISBN: 1111138214

Classification of Elementary Stamp Shapes by Means of Reduced Point Distance Histogram Representation

Paweł Forczmański and Dariusz Frejlichowski

West Pomeranian University of Technology, Szczecin,
Faculty of Computer Science and Information Technology,
Żołnierska Str. 52, 71–210 Szczecin, Poland
{pforczmanski,dfrejlichowski}@wi.zut.edu.pl

Abstract. The paper presents a problem of stamp shape classification, where an input stamp is given as a bitmap containing binary values. While every stamp features a specific geometrical form coming from the *de facto* standards of stamping process, thus it can be classified as round, oval, square, rectangular or triangular. We assume to have a detected stamp and in this paper we deal with the stage of features extraction and reduction, by means of Point Distance Histogram (at the stage of features extraction) and Principal Component Analysis and Linear Discriminant Analysis (at the stage of dimensionality reduction). The final classification employs similarity evaluation involving hand-drawn templates, ideal shapes and average descriptors calculated for the entire database. Despite the fact that there are only several basic stamp shapes, the task is not trivial since there are many variations in size, silhouette and complexity of individual stamps. It should be emphasized that the scanned document may be degraded in quality thus extracted stamp can be distorted (the silhouette may be discontinuous and/or can be noised). The paper provides some experimental results on real documents with different types of stamps and a comparison with a classical Discrete Cosine Transform (DCT) and PCA applied on image matrix.

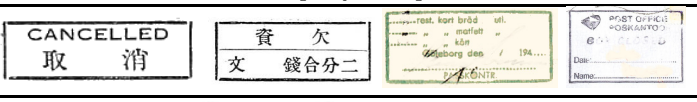
1 Introduction

Classification and recognition of two-dimensional shapes are classical Computer Vision problems with huge bibliography and very robust and efficient algorithms [1]. However not all algorithms can be applied to any specific task. The problem of stamp shapes classification and recognition has been addressed in the scientific literature in several works related to stamps or logo detection [2,3,4,5,6].It has emerged from the task of seal imprint identification on bank checks, envelopes, and transaction receipts in mid-1980s [2]. However, it should be noticed that the reliable recognition of stamps in the documents has not been solved in any satisfactory way till today. One of the most advanced method of stamps detection found in the scientific literature [4] scans for stamps considered as regions with analytically shaped contours, however in that work these regions are limited to

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 603–616, 2012.

oval (round) shapes only. Other methods look for stamps given as sets of "corners" and detect them using some fuzzy rules [6]. The other possible approach to stamp detection employs color segmentation and simple geometrical features to extract stamps from documents [7,8]. It is rather easy to define the features that can be employed to describe stamps that can be found in the documents. In this paper we focus on official stamps since they play a meaningful role in practical tasks. Typical stamps which can be found in paper documents have specific characteristics, which are derived from the process of stamping. These characteristics (e.g. shape, complexity, typical patterns) have evolved during many centuries into de facto standards. The analysis of the problem shows that the shape of stamp defines its category, thus the official stamps are in most cases round. There are four other classes that cover most of the shapes of stamps (given in the order of descending popularity): oval, rectangular, square and triangular. Sample members of these classes are presented in Tab.1. As it can be seen, they are often regular and without distinct decorations. Typical stamp, no matter of the shape, consists of regularly-shaped silhouette with clearly visible text and mere ornaments. Hence, the class can be determined by analysis of the general shape. While any method that uses analytically described shape (e.g. generalized Hough transform) can be applied to the task of stamp detection, the classification has to employ more task-oriented descriptors.

Table 1. Sample official stamps, divided into five classes

class	exemplary stamps
rectangular	
square	
triangular	
round	
oval	

The general motivation of the research presented in this paper is a shortage of algorithms that are able to classify visual objects such as stamps. The application area of this kind of an algorithm is broad, ranging form law-enforcement forces, law offices, official archives and any other institutions that utilize stamping technique. Instead of browsing large sets of images stored on digital media, the proposed software approach is able to retrieve images that contain objects recognized as certain stamps. This paper extends the research described in [7,8] in the area of classification. Moreover, the benchmark database has been extended and covers wider range of stamps variants and their distortions.

The algorithms of initial stamps processing have been presented in [7,8]. Both lead from stamp detection to the stamp localization and extraction. Further in this paper we describe a procedure of stamp classification, which is divided into several stages: calculation of Point Distance Histogram (PDH), reduction of features dimensionality by means of Principal Component Analysis/Karhunen Loeve Transform (PCA/KLT) and Linear Discriminant Analysis (LDA) followed by distance calculation by means of Euclidean Metrics. The flow of computations is presented in Fig. 1. Detailed description of each stage is presented in the following sections.

Fig. 1. Process of feature extraction, reduction, and classification

2 Features Extraction/Reduction

2.1 Point Distance Histogram Representation

We assume that an input image of a document is stored in a file with possibly lossless compression, high spatial resolution and full color range. The process of stamp detection, localization and extraction was previously described in [7,8]. Localized stamps are extracted from an image and passed to recognition stage in which a particular object is not precisely identified, but only it is assigned to one of few basic classes, corresponding with the most common stamp shapes. Those shapes indicate the type of a stamp.

Having the above assumptions in mind, for the experiments presented in this paper, a shape description algorithm was applied at the first stage of the stamp shape representation, preceded by its localization and extraction. Usually in this context the so-called shape descriptors are applied. They are supposed to describe a shape in a way that makes this representation invariant to shape transformations and robust to as many problems as possible. The most common planar contour shape deformations can be divided into three main groups. The first one includes the affine transformations, e.g. translation, scaling, rotation. The second group covers two problems connected with the varying number of

points on the outline. The first one is the noise that results in a local change of point's co-ordinates. The second is the occlusion. It results in more global deformations of a shape — lack or addition of part of an object. The third group of problems is strictly related to the contour representation of a shape and includes the direction of tracing the contour, starting point selection, and the number of points. Since the shape analysis is one of the most explored approaches to the problem of object recognition, there are many algorithms developed so far [1,9]. The problem was investigated in [7] which lead to the selection of a few state-of-art descriptors: *2D Fourier Descriptors*, *Point Distance Histogram*, *Roundness*, *Moment Invariants* and *UNL-Fourier*. As it turned out the best results in the task of general shape classification were obtained for *Point Distance Histogram* thus this method is chosen for further investigation. Selected algorithm has some important advantages fulfilling the requirements related to the scale and rotation invariance. It is a combination of the polar transform and the derivation of the histogram. Thanks to this property the descriptor is invariant to scaling, rotation and shifting. It is also invariant to starting point selection and direction of contour tracing. Moreover, by setting the number of the bins in the resultant histogram (the parameter r in the algorithm) we can influence on the generalization ability. Therefore, the PDH was applied to the problem of identification of particular types of stamps at the stage of their description. The PDH approach has already been employed in the problem of General Shape Analysis [7,8], which is close to similar shape retrieval. Here, it is used for classification, which seems to be more difficult since the results should be much more 'crisp' that 'fuzzy' answers given by GSA.

The Point Distance Histogram uses Cartesian-to-polar transformation. It starts with calculation of the origin of this transform. Although any point can be used for this purpose, here, the most popular one — centroid — is chosen [7]. The centroid is easy to calculate, and stable in the presence of affine transformations. Since the objects we are dealing with, consist of finite set of K points p: $p_1, p_2, \ldots, p_K$ in $\mathbb{N}^2$, it is calculated according to the following formula:

$$S = \frac{1}{K} \sum_{k=1}^{K} p_k. \tag{1}$$

In case of noise, the centroid calculation can suffer from some inaccuracies, yet this problem is diminished by further dimensionality reduction stage. Using the calculated centroid we achieve the polar coordinates of points describing the stamp. They will be stored in two vectors — Θ for angles (in degrees) and P for radii [7]. The values in the vector Θ are later converted into nearest integers [7]. The elements in the vectors Θ and P are rearranged in accordance with the increasing values in Θ and put into two new vectors denoted as $\hat{\Theta}$ and $\hat{P}$. Next we check if there are any equal values in $\hat{\Theta}$ and only the one with the highest corresponding value in $\hat{P}$ is left while the others are removed. Thus, we create a vector $\tilde{P}$ having only selected elements from $\hat{P}$ (at most 360 elements). This process is performed in order to eliminate repeating angular values by selecting only the most distant from the centroid. Thanks to this only the outer points

of the contour are selected. Moreover, the method gives a descriptor that is less sensitive to the internal structure of an object. In further processing we create a new vector $\acute{P}$, where elements $\acute{\rho}_k$ are created from $\tilde{\rho}_k$ normalized to the maximal value in $\tilde{P}$ [7]. Finally, the histogram is built — elements in $\acute{P}$ are assigned to r bins in a vector H.

2.2 Features Dimensionality Reduction by Means of PCA/LDA

The volume of information to process should be reduced considerably to make the system memory requirements lower. Since the dimensionality of resultant PDH descriptors can be rather large ($r > 25$) it is necessary to introduce preliminary reduction stage (PCA/KLT) in order to make it possible to employ LDA. Hence, our procedure is divided into two main stages: preliminary reduction by means of PCA/KLT and main reduction/clusterization using LDA.

In the first step we divide template descriptors, calculated according to the above presented procedure, into $K = 5$ classes having in mind five most popular shapes of stamps, i.e. round, oval, square, rectangular and triangular. For the purpose of classification we have generated each template in two variants: as ideal silhouettes and shapes drawn manually. They can be observed in Fig.2. The PDH descriptors calculated for above mentioned templates are presented in Fig.3 (first and second row). In order to increase recognition rate and match the characteristics of the data (which will be described later in the paper), average descriptors have been also calculated. They are the results of simple averaging of all PDH descriptors in each class independently. They can be observed in the last row of Fig.3.

Fig. 2. Templates used in the process of classification — 5 pairs consisting of ideal silhouette and manually drawn template

First Analysis Stage: Principal Component Analysis. A PDH descriptor of a template stamp $j = \{1, 2, \ldots, J\}$ denoted as H_j is a vector of dimensionality $1 \times r$. In the first step, in order to normalize data, mean descriptor is calculated as a proportion of sum of all database descriptors and their total number [10]:

$$\bar{H} = \frac{1}{J} \sum_{j=1}^{J} H_j \tag{2}$$

and removed from all templates in the database $\hat{H}_j = H_j - \bar{H}, \forall j = 1 \ldots J$. Next, once for the database templates, we build a covariance matrix C of $r \times r$ elements. It corresponds to descriptors variance within the set [11]:

$$C = \sum_{j=1}^{J} \hat{H}\hat{H}^T. \tag{3}$$

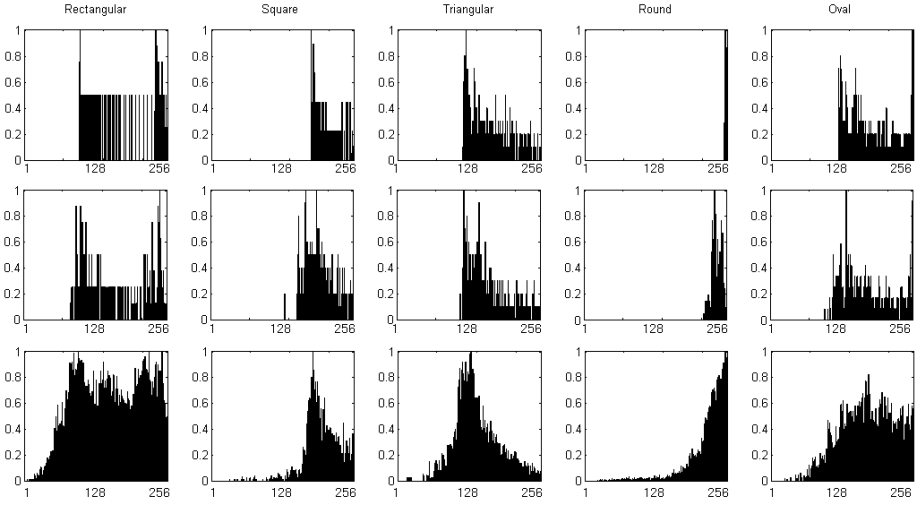


Fig. 3. The PDH descriptors for templates used in the process of classification — ideal silhouettes in the first row, manually drawn templates in the second and average PDH descriptors calculated over the entire database — in the third row

Next, two matrices: D and V are calculated, which are in the following relation with each other:

$$D = V^T C V. \tag{4}$$

Above matrices carry eigenvalues (D) and eigenvectors (V) of the covariance matrix C. It should be noted that this stage requires to solve the matrix equation and is performed using adequate numerical methods [11]. It should be remembered that proper diagonalization of above is possible only when V is orthogonal. In order to form the transformation matrix F_{PCA}, we sort elements on the diagonal of D in the descending order and select first p elements. From matrix V we select p columns which are related to respective elements in eigenvalues matrix ($p \leq r$) and form transformation matrix F_{PCA} of size $r \times p$.

First Reduction Stage: Karhunen-Loeve Transform. For each descriptor H_j from the database in order to get its reduced representation X_j we perform the projection:

$$X_j = F_{PCA}^T (H_j - \bar{H}), \ \forall j = 1 \ldots J; \tag{5}$$

The set consisting of $H_j, \forall j = 1, \ldots, J$ constitutes the intermediate template database used for further processing, divided into K classes.

In figures below one can observe the reduced feature-space after the PCA stage. Figure 4 presents a projection of stamp database onto first and second components of PCA while Fig. 5 presents a projection onto first and third components, respectively. The classes are denoted as follows: $\times$-rectangular, $\square$-square, $\triangle$-triangular, $\diamond$-oval, and $\bigcirc$-round. The objects in the space are distributed in

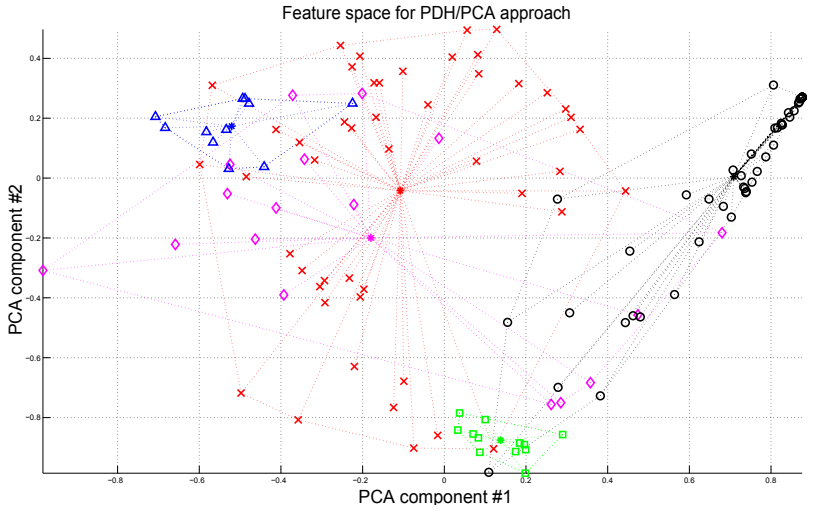


Fig. 4. The features space (the two first principal components) for PDH/PCA approach

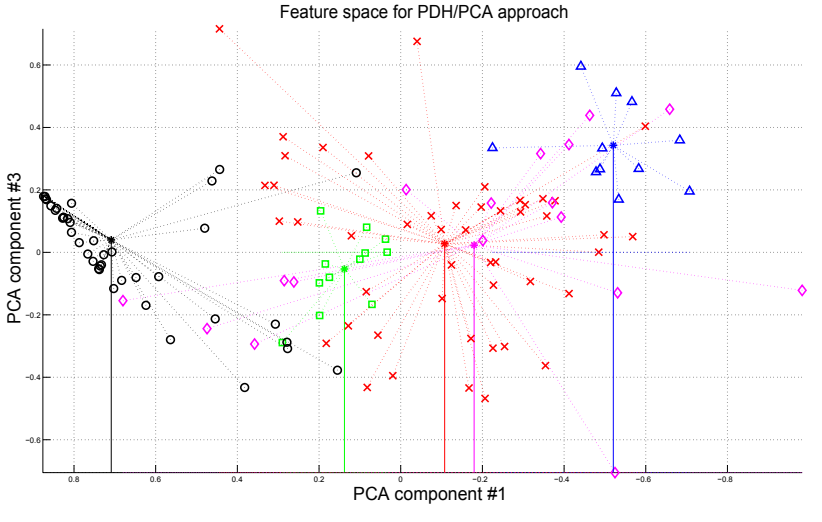


Fig. 5. The features space (the first and the third principal components) for PDH/PCA approach

a homogeneous manner and the centres of all classes are close to each other, make successful recognition very complicated.

Second Analysis Stage: Linear Discriminant Analysis. A KLT-reduced descriptor of a template stamp $j = \{1, 2, \ldots, J\}$ denoted as X_j is a vector of

dimensionality $1 \times p$. First, we build two covariance matrices [11] W and B related to within-class and between-class scatter in the set:

$$W = \sum_{j=1}^{J} (X_j - \bar{X}^k)(X_j - \bar{X}^k)^T, \tag{6}$$

where $\bar{X}^k$ is a mean descriptor in k-th class, while

$$B = \sum_{k=1}^{K} (\bar{X}^k - \bar{X})(\bar{X}^k - \bar{X})^T, \tag{7}$$

where $\bar{X}$ is an overall mean in the whole set.

Then, a total scatter covariance matrix is calculated:

$$Z = W^{-1}B \tag{8}$$

which is later decomposed into two matrices G and U, responsible for eigenvalues and eigenvectors, respectively:

$$G = U^T Z U. \tag{9}$$

Above matrices carry eigenvalues (G) and eigenvectors (U) of the covariance matrix Z. From a matrix U we create transformation matrix F_{LDA} in the same way as in the case of PCA.

Second Reduction Stage: Karhunen-Loeve Transform. Having only $s \times p$ elements in the transformation matrix F_{LDA} ($s < p$) we obtain final reduced descriptors Y_j by a projection:

$$Y_j = F_{LDA}^T X_j, \ \forall j = 1 \ldots J; \tag{10}$$

The set consisting of Y_j constitutes final template database used for further processing, divided into K classes.

In figures below one can observe final reduced feature-space after the LDA stage. Figure 6 presents a projection of stamp database onto first and second components of LDA while Fig. 7 presents a projection onto first and third components, respectively. The classes are denoted as follows: $\times$-rectangular, $\square$-square, $\triangle$-triangular, $\diamond$-oval, and $\bigcirc$-round. The feature space is much more "friendly" in a comparison to the PCA stage. The centres of all classes are close to each other, make successful recognition employing any distance metric much more probable.

2.3 Classification

The total number of (test) objects in the database is denoted as L. An i-th object (stamp) presented for recognition is processed in the following manner.

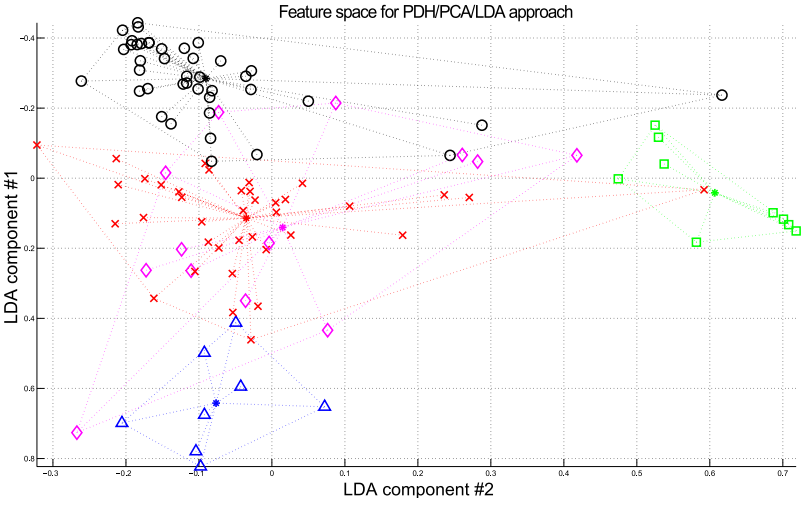


Fig. 6. The features space (the two first LDA components) for PDH/PCA/LDA approach

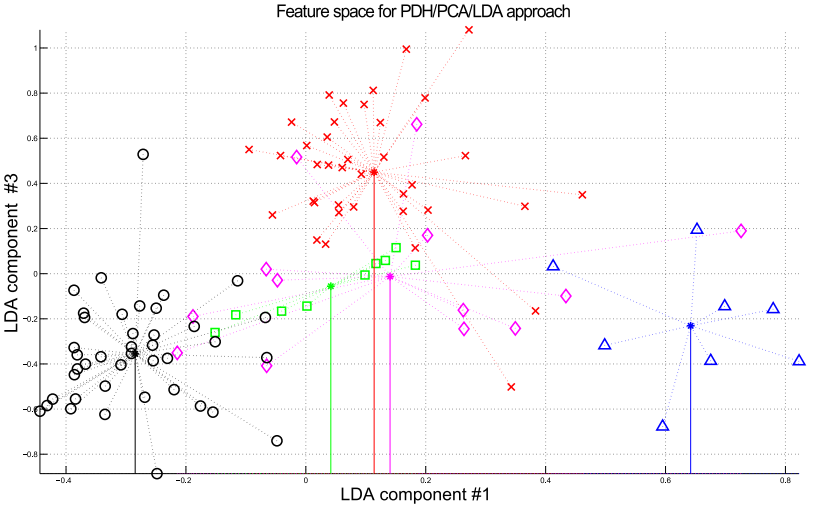


Fig. 7. The features space (the first and third LDA components) for PDH/PCA/LDA approach

First the PDH descriptor H_i is calculated, then it is centered and projected into PCA feature-space, then it is projected into LDA feature-space. It should be stressed that the average element $\bar{H}$ and transformation matrix F_{PCA} in the PCA stage and F_{LDA} in the LDA stage are calculated over J database templates only, not involving L test images. In order to do the recognition we perform

classification using distance calculation, namely we calculate the distance in the reduced feature space using Euclidean metrics. The distance is equal to the dissimilarity measure between two reduced shape descriptions (denoted as X_i and X_j):

$$L_2(X_i, X_j) = \sqrt{\sum_{n=1}^{p} ((x_1(n) - x_2(n))^2}. \tag{11}$$

The classification uses a simple distance criterion. We choose the closest object (or mean object), which implies a recognized class. Further in the paper we show the results of the experiments performed on our own benchmark database using three variants of distance calculation:

1. to the ideal templates (PCA components in a reference database are calculated for descriptors of ideal shapes only) — see first row of Fig.3,
2. to the drawn templates (PCA components in a reference database are calculated for descriptors of hand drawn images only) — see second row of Fig.3,
3. to the average descriptor in each class (PCA/LDA components of objects constituting a reference database) — see third row of Fig.3, which presents reconstructed descriptors for each class.

3 Experimental Results

The experiments were performed on a database consisting of bitmaps containing scanned stamps, which were extracted from freely available images gathered from the Internet. It should be stressed, that it is difficult to find a large dataset, especially a benchmark one, containing some thousands of stamp images. The stamps in our own dataset were extracted form scanned images in a semi-automatic way using the algorithms described in [7,8]. They were thresholded and stored in a binary form. The size of bitmaps ranged from about 50×50 to 256×256 pixels, depending of the source of the original image. All 140 stamps were divided into 5 classes: rectangular (48 elements), square (12 elements), triangular (11 elements), round (52 elements), and oval (17 elements). All stamps in the database were divided into two non-overlapping subsets: learning set of J elements and testing set of L elements in the random manner. First we wanted to evaluate the performance of simple PDH/PCA approach, involving shape descriptor reduced with help of global PCA, which means common covariance matrix for all elements in all classes. The classification was performed according to the approach described in the above section, with three variants of distance calculation. The results of the recognition are presented in Tab.2, where column 'ideal' stands for ideal templates, 'hand drawn' — for manually drawn images, and 'average' stands for average descriptors calculated for all learning images in each class (they have been chosen using random manner from the whole set).

Then, we performed dimensionality reduction of PDH descriptors using two-stage approach (PCA/LDA), with PDH/PCA descriptors calculated in the above

Table 2. Recognition performance of PDH/PCA for different templates

PC	ideal						hand drawn						average					
	PDH size (number of bins)						PDH size (number of bins)						PDH size (number of bins)					
	8	16	32	64	128	256	8	16	32	64	128	256	8	16	32	64	128	256
1	0.56	0.40	0.41	0.42	0.33	0.40	0.65	0.56	0.42	0.32	0.31	0.31	0.49	0.56	0.48	0.47	0.56	0.47
2	0.56	0.57	0.58	0.58	0.55	0.49	0.66	0.68	0.65	0.64	0.61	0.61	0.71	0.71	0.69	0.69	0.70	0.70
4	**0.64**	0.59	0.51	0.57	0.59	0.50	**0.72**	0.66	0.66	0.68	0.67	0.59	0.76	0.76	0.76	0.76	0.77	**0.79**
8	0.64	0.59	0.51	0.57	0.59	0.50	0.72	0.66	0.66	0.68	0.67	0.59	0.76	0.76	0.76	0.76	0.77	0.79
16	-	0.59	0.51	0.57	0.59	0.50	-	0.66	0.66	0.68	0.67	0.59	-	0.76	0.76	0.76	0.77	0.79
32	-	-	0.51	0.57	0.59	0.50	-	-	0.66	0.68	0.67	0.59	-	-	0.76	0.76	0.77	0.79
64	-	-	-	0.57	0.59	0.50	-	-	-	0.68	0.67	0.59	-	-	-	0.76	0.77	0.79
128	-	-	-	-	0.59	0.50	-	-	-	-	0.67	0.59	-	-	-	-	0.77	0.79

Table 3. Recognition performance of PDH/PCA/LDA for different number of Principal Components (PC), PDH size and different hold-off schemes (output feature-space is equal to 4)

PC	a) $J = 48, L = 92$					b) $J = 69, L = 71$					c) $J = 109, L = 31$				
	PDH size					PDH size					PDH size				
	16	32	64	128	256	16	32	64	128	256	16	32	64	128	256
12	0.69	0.73	0.76	0.82	0.77	0.80	0.80	0.80	0.83	0.77	0.77	0.74	0.77	0.77	0.71
16	-	0.76	0.80	**0.81**	0.75	-	0.83	0.87	**0.89**	0.75	-	0.74	0.77	**0.84**	0.77
20	-	0.71	0.78	0.79	0.78	-	0.76	0.80	0.85	0.82	-	0.71	0.81	0.74	0.77
24	-	0.60	0.73	0.79	0.77	-	0.77	0.79	0.85	0.83	-	0.77	0.81	0.74	0.81
32	-	-	0.63	0.69	0.69	-	-	0.79	0.83	0.79	-	-	0.84	0.74	0.81
40	-	-	0.41	0.54	0.55	-	-	0.68	0.80	0.82	-	-	0.84	0.84	0.77
48	-	-	0.23	0.17	0.25	-	-	0.69	0.75	0.79	-	-	0.77	0.77	0.77
50	-	-	0.33	0.18	0.22	-	-	0.62	0.76	0.80	-	-	0.74	0.77	0.77

presented manner. The final dimensionality, after second reduction, was $s = 4$, since there are 5 classes of stamps. There were three types of set hold-off schemes: a)$J = 48, L = 92$, b)$J = 69, L = 71$, and c)$J = 109, L = 31$. The detailed results are presented in the following table (Tab. 3).

The results confirm the superiority of two-stage dimensionality reduction involving PCA/LDA over the traditional PCA approach. With the same output dimensionality, the performance of the first one is lower by 2%-10%, depending on the learning set size. The closer analysis of the results shows that failed recognition refers to ambiguous objects in the testing database. Some of them were shown in Fig.8. As it can be seen, most of them have discontinuous edges or are not conforming with strict class properties. The other possible reason for the unsuccessful recognition may come from mixing squares and rectangles in one object (being one stamp actually).

oval round square rectangular rectangular

Fig. 8. Sample ambiguous objects lowering the recognition rate

The detailed, 5-class confusion matrix for PDH/PCA/LDA with parameters $L = 69$, $J = 71, r = 128, p = 16, s = 4$ (which actually give the best overall recognition rate) is presented in table below (see Tab 4).

Table 4. Confusion matrix for specific parameters selection

	rectangular	square	triangular	round	oval
rectangular	24	0	0	0	0
square	0	6	0	0	0
triangular	2	0	4	0	0
round	0	0	0	26	0
oval	2	0	2	2	3

As it can be seen, some classes are much more distinguishable than the others, e.g. rectangular, square and round shapes give 100% successful recognition, while the other shapes gave some errors. It is probably because of the fuzzy borders between shape definitions (i.e. round and ova or triangular and oval).

In order to evaluate the performance of the presented method in a comparison to several previously proposed algorithms we performed some experiments. They involved reducing the dimensionality of input images by means of classical DCT2, applying PCA on the bitmaps, joining PDH representation with DCT and PCA. All these algorithm were using the most optimal parameters, namely the parameters giving the best recognition rate. The results of the experiments are presented in Tab.5, where column 'highest rate' represents the best recognition rate, 'mean rate' shows mean recognition rate within one algorithm (and different parameter configurations). It can be clearly seen that the best results gives joint approach involving PDH, PCA and LDA calculated for all objects in the testing database.

Table 5. Recognition performance of different recognition algorithms

method	direct DCT	direct PCA	PDH/DCT	PDH/PCA	PDH/PCA/LDA
highest rate	0.57	0.75	0.76	0.79	0.89
mean rate	0.41	0.61	0.74	0.77	0.81

4 Summary

The proposed image recognition algorithm has, among others, the following useful features: robustness to the noise, high processing speed, and it can be easily adapted to low computing power devices. The algorithm presented in the paper was developed for the specific purpose, namely the identification of stamp types extracted from digital images. It is divided into two main stages. At the first stage, the processed stamp (binary matrix) is represented using a shape descriptor. In the paper the Point Distance Histogram was applied for this purpose. This choice was influenced by the advantages of this descriptor, particularly

useful in the considered problem (e.g. invariance to the affine transforms of planar shape, generalization property). As it turned out two important conclusions arose during the experiments. First of all, increasing the number of bins in the obtained histogram may lower the recognition performance. It comes from the fact that in case of stamp images, which are small in size, the number of points on the silhouette could be too low. At the second stage, Principal Component Analysis/ Karhunen-Loeve Transform was used with additional Linear Discriminant Analysis-based reduction. The dimensionality reduction performed on descriptors calculated by means of PDH have shown, that only four first output components are useful for classification purposes. Moreover, the intermediate dimensionality (after PCA) is crucial and should be not more than 16.

The algorithm PDH/PCA/LDA gives practically applicable approach which reaches high recognition rates. The possible area of applications of this algorithm is broad and was briefly described in the introductory part of the paper.

It should be noted, that the presented algorithm may be extended to a more general approach, giving a possibility to solve any other problem related to shape classification. Its main advantages are a low-dimensional feature vector and simple comparison stage, which is very desired property when it comes to hardware implementation.

Acknowledgements. This work was supported by National Science Centre (NCN) within the grant N N516 475540.

References

1. Zhang, D., Lu, G.: Review of shape representation and description techniques. Pattern Recognition 37, 1–19 (2004)
2. Ueda, K., Nakamura, Y.: Automatic verification of seal impression patterns. In: Proc. 7th. Int. Conf. on Pattern Recognition, pp. 1019–1021 (1984)
3. Pham, T.D.: Unconstrained logo detection in document images. Pattern Recognition 36, 3023–3025 (2003)
4. Zhu, G., Jaeger, S., Doermann, D.: A robust stamp detection framework on degraded documents. In: Proceedings — SPIE The International Society For Optical Engineering, vol. 6067 (2006)
5. Zhu, G., Doermann, D.: Automatic Document Logo Detection. In: The 9th International Conference on Document Analysis and Recognition (ICDAR 2007), pp. 864–868 (2007)
6. He, J., Downton, A.C.: Configurable Text Stamp Identification Tool with Application of Fuzzy Logic. In: Marinai, S., Dengel, A.R. (eds.) DAS 2004. LNCS, vol. 3163, pp. 201–212. Springer, Heidelberg (2004)
7. Frejlichowski, D., Forczmański, P.: General Shape Analysis Applied to Stamps Retrieval from Scanned Documents. In: Dicheva, D., Dochev, D. (eds.) AIMSA 2010. LNCS, vol. 6304, pp. 251–260. Springer, Heidelberg (2010)

8. Forczmański, P., Frejlichowski, D.: Robust Stamps Detection and Classification by Means of General Shape Analysis. In: Bolc, L., Tadeusiewicz, R., Chmielewski, L.J., Wojciechowski, K. (eds.) ICCVG 2010. LNCS, vol. 6374, pp. 360–367. Springer, Heidelberg (2010)
9. Wood, J.: Invariant Pattern Recognition: A Review. Pattern Recognition 29, 1–17 (1996)
10. Jolliffe, I.T.: Principal Component Analysis. Springer, NY (1986)
11. Kukharev, G., Forczmański, P.: Data Dimensionality Reduction for Face Recognition. Machine Graphics & Vision 13(1/2), 99–122 (2004)

A Multiclassifier Approach for Drill Wear Prediction

Alberto Diez and Alberto Carrascal

Tecnalia Research & Innovation, Paseo Mikeletegi 7, Parque Tecnológico,
20009 Donostia-San Sebastián (Guipúzcoa, España)

Abstract. Classification methods have been widely used during last years in order to predict patterns and trends of interest in data. In present paper, a multiclassifier approach that combines the output of some of the most popular data mining algorithms is shown. The approach is based on voting criteria, by estimating the confidence distributions of each algorithm individually and combining them according to three different methods: confidence voting, weighted voting and majority voting. To illustrate its applicability in a real problem, the drill wear detection in machine-tool sector is addressed. In this study, the accuracy obtained by each isolated classifier is compared with the performance of the multiclassifier when characterizing the patterns of interest involved in the drilling process and predicting the drill wear. Experimental results show that, in general, false positives obtained by the classifiers can be slightly reduced by using the multiclassifier approach.

Keywords: Classification, multiclassifier, drill wear prediction, pattern identification.

1 Introduction

In the machine-tool sector, emerging industrial processes and methodologies due to technological improvements in manufacturing lines require new valid solutions when detecting failures and scheduling maintenance operations [1]. Preventive and corrective maintenance procedures have been developed and widely used during the last years; nevertheless, there yet exist important maintenance gaps to be fulfilled [2]. The research activity done in this field is focused on how to predict that something unexpected is going to happen before it really occurs, with the aim of avoiding overhead costs derived from production line breakdowns and maintenance operations. Machinery builders and vendors usually provide preventive maintenance strategies, but they are focused on normal operating conditions [3]. Benefits derived from anomaly prediction are not only oriented to reduce costs and to optimize the machine lifecycle, but also to infer new relevant knowledge about the process and the most probable cause and propagation of the problem.

The motivation regarding drill wear detection and prediction in the machine-tool processes is mainly focused on the loss of quality of the resulting holes as the drill used is close to the decline stage of its lifecycle. Into this work, new diagnosis methods regarding drill wear characterization and classification by means of a

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 617–630, 2012.

multiclassifier approach is presented. Many authors have adopted approaches based on the combination of classifiers for resolving different problems and by using different combination rules and strategies [4]. The multiclassifier proposed in this study combines the outputs of different data mining techniques based on voting criteria and on the given label distribution, therefore more accurate predictions can be achieved. The goodness of this approach is compared with results obtained by each isolated classifier when addressing the drill wear problem, characterized by the presence of burr and roughness on holes drilled. Experiments performed consist on several drilling experiments, in which several process parameters are monitored and analyzed to extract the most relevant patterns associated to drill wear problem. It is demonstrated that the use of data mining algorithms provides a promising methodology and decision making support tool regarding drill substitution strategy.

The layout of the paper is as follows: the problem under study and its characterization is introduced in Section 2; the data mining algorithms used and the multiclassifier approach are shown in Section 3; in Section 4 experimental results are analyzed and discussed; finally, conclusions of this study are presented in Section 5.

2 The Drill Wear Problem

The quality of the resulting workpiece in drilling process can be estimated by means of two physic parameters: burr and roughness in the resulting holes drilled [5]. The more wear is the drill, the less quality is obtained. For this reason, over certain threshold values it is strongly recommended to replace the drill in order to assure enough quality and to avoid its breakage, which can provoke serious safety problems and maintenance costs.

The main motivation of this paper is to analyze how drill wear influences in the quality of the hole drilled and how to characterize it by means of most relevant patterns inferred from monitoring data. Several experiments regarding drilling process have been monitored in order to acquire the data needed to tackle with this study.

2.1 The Drilling Process

During drilling process, the intensity of the engine regulator of machine spindle and the intensity of the engine regulator of machine head are measured. The first one is related to force needed in each drilling operation, F, and the second one is the torque, T. The signals assessed show four significant points of interest that delimitate three cutting areas, as showed in Fig. 1. Those four points are the following:

1. When the drill head comes into contact with the workpiece and is introduced into the material; the intensity of the signal increases.
2. When all drill diameter is inside the workpiece.
3. When the drill head exits from the surface of the workpiece; the intensity of the signal decreases.
4. When the drill is totally outside the workpiece.

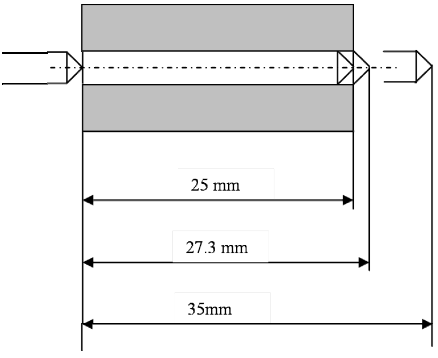


Fig. 1. Cutting areas of interest during a drilling process

Given intensity signals, F and T, obtained during each drilling operation for each critical cutting area, the variables of interest that characterize the drilling process and that can be affected by drill wear are calculated. Those variables are briefly described in Table 1.

Table 1. Variables measured during the drilling process

Attribute	Description
F23_MEAN	Advance force mean value when all the drilling tool diameter is drilling inside the workpiece
F23_STDE	Advance force standard deviation when all the drilling tool diameter is drilling inside the workpiece
F23_SKEW	Advance force bias when all the drilling tool diameter is drilling inside the workpiece
T23_MEAN	Torque mean value when all the drilling tool diameter is drilling inside the workpiece
T23_STDE	Torque standard deviation when all the drilling tool diameter is drilling inside the workpiece
T23_SKEW	Torque bias when all the drilling tool diameter is drilling inside the workpiece
F12_SLAN	Force slope at the entry point of the workpiece
F12_TIME	Time measured from the drilling tool makes contact with the workpiece until all the drilling tool diameter is drilling inside the workpiece, in terms of force
T12_SLAN	Torque slope at the entry point of the workpiece
T12_TIME	Time measured from the drilling tool makes contact with the workpiece until all the tool diameter is drilling inside the workpiece, in terms of torque
F23_A001	Force value area under the FFT in the following rank of frequency: 50 Hz – 250 Hz
T23_A001	Torque value area under the FFT in the following rank of frequency: 50 Hz – 250 Hz
Stde_Fin	Drilling force standard deviation when all the drilling tool diameter is drilling inside the workpiece, at the beginning of the hole
Fin_MEAN	Drilling force mean value when all the drilling tool diameter is drilling inside the workpiece, at the beginning of the hole
Stde_Fout	Drilling force standard deviation when all the drilling tool diameter is drilling inside the workpiece, at the end of the drilling process
Fout_MEAN	Drilling force mean value when all the drilling tool diameter is drilling inside the workpiece, at the end of the drilling process

Table 1. (*Continued*)

Delta_F	Difference between the drilling force mean value at the beginning of the hole and when finishing the drilling process
Stde_Tin	Torque standard deviation when all the drilling tool diameter is drilling inside the workpiece, at the beginning of the hole
Tin_MEAN	Torque mean value when all the drilling tool diameter is drilling inside the workpiece, at the beginning of the hole
Stde_Tout	Torque standard deviation when all the drilling tool diameter is drilling inside the workpiece, at the end of the drilling process
Tout_MEAN	Torque mean value when all the drilling tool diameter is drilling inside the workpiece, at the end of the drilling process
Delta_T	Difference between the torque mean value at the beginning of the hole and when finishing the drilling process
Roughness	The roughness of the workpiece, measured in microns
Burr	The burr of the workpiece, measured in microns

During different drilling experiments torque evolution, T, shows a slight variation as the number of holes drilled increases: torque measurement is hardly increased at the end of the drill. This evolution is shown in Fig. 2. At the moment the drill exits from the workpiece, force evolution, F, also shows an ascendant trend in relation with the number of holes drilled, as can be appreciated in Fig. 3.

It is assumed that drill wear effect can be easily estimated by measuring burr and roughness of holes drilled, as can be seen in Fig. 4, given the fact those measurements strongly and critically fluctuates as the number of holes drilled increases. Therefore, in this study the quality of hole drilled has been characterized in terms of burr and roughness.

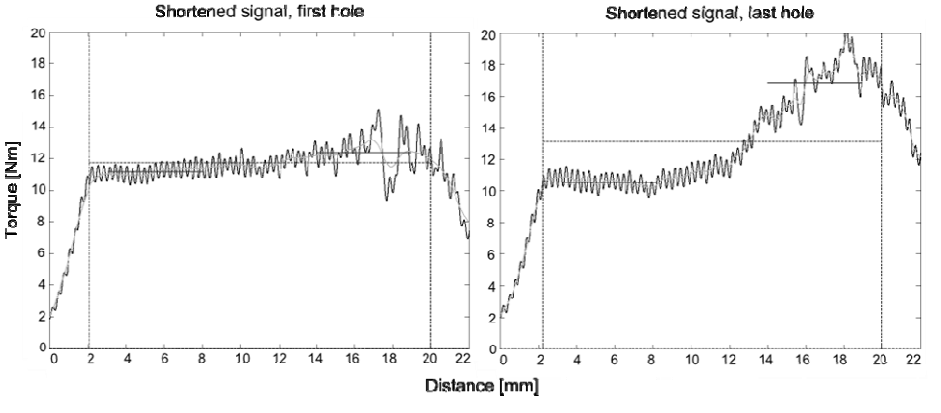


Fig. 2. Torque signal evolution among different drilling experiments: torque signal in the first hole drilled (left) and in the last the drilling operation performed (right)

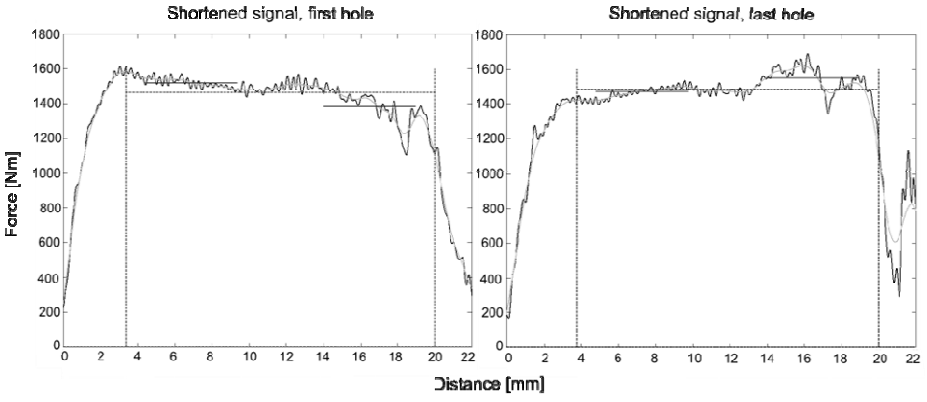


Fig. 3. Force signal evolution among different drilling experiments: force signal in the first hole drilled (left) and in the last drilling process performed (right)

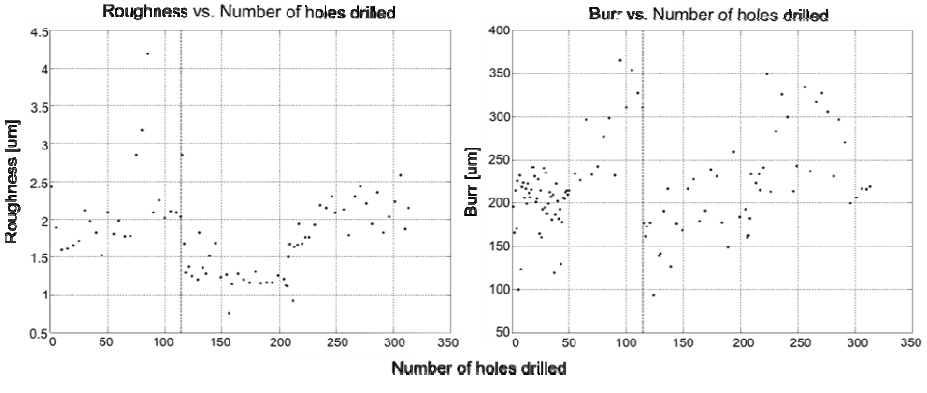


Fig. 4. Graphics of roughness (left) and burr (right) vs. number of drilling processes performed

3 Methodology

The proposed drill wear prediction methodology accomplished the following steps: 1) filtering samples that contained incorrectly measured signals when gathering data from the drilling processes performed, 2) applying feature selection algorithm, PCA, to try to reduce search space and to prioritize problem-related attributes, 3) performing supervised classification by means of different data mining algorithms and 4) combining the information from these classifiers using different strategies, based on proposed multiclassfier approach.

3.1 Feature Selection

The number of predictor variables is usually huge and the amount of information provided is different depending on the variable. It is fairly common to apply feature selection methods in order to identify patterns in data and, thus, to reduce the number

of dimensions of the problem. Due to its capability of establishing any correlation among the parameter values without much loss of information, a Principal Component Analysis (PCA [6]) has been applied. Therefore, dimensionality reduction of data can be accomplished by eliminating the principal components with low variance. This means that they are not relevant with regards to the problem.

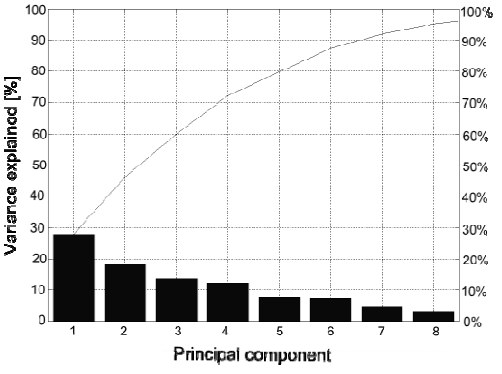


Fig. 5. PCA results: variance histogram

In case under study, since the distribution in the new system of reference shows a low correlation among variables as can be seen in Fig. 5, it is not finally used. All variables provide enough information to outline the drill wear problem.

3.2 Supervised Classification

Classification algorithms employed in this study have in common that all of them performs a supervised classification. This is due to the nature of the data set used, in which each experiment or sample is classified according to resulting burr and roughness values. There exist other classification methods, such as non-supervised or semi-supervised classification, whose applicability is more oriented to detect anomalies when knowledge of data behaviour is limited or even inexistent [7].

The supervised classification paradigm [8] consists of a set of N facts, each of them characterized by $n+1$ variables; first n variables, $X_1, X_2,\ldots, X_n$, would be predictor variables and the variable with index $n+1$, identified as C, would be the *class* variable. These data can be represented in table format using the following notation:

1. x_i^j is the value that the j-th fact takes in the i-th predictor variable, $i = 1, \ldots, n$ and $j = 1, \ldots, N$;
2. c^j is the class that the j-th fact belongs to.

Facts can be also named as cases or instances and the variables are the attributes of the supervised problem. The goal is to obtain a classification model that is able to predict the value of variable C when a new case is analyzed. This new case will be composed of n predictor variables and an unknown value of variable C. Consequently, the problem consists on correctly classifying a new fact based on previous evidences or cases.

Table 2. Supervised classification elements

	X_1	X_2	…	X_i	…	X_n	C
1	x_1^1	x_2^1	…	x_i^1	…	x_n^1	C^1
⋮	⋮	⋮	⋱	⋮	⋱	⋮	⋮
J	x_1^j	x_2^j	…	x_i^j	…	x_n^j	C^j
⋮	⋮	⋮	⋱	⋮	⋱	⋮	⋮
N	x_1^N	x_2^N	…	x_i^N	…	x_n^N	C^N

In order to accomplish this task properly, data set should be wide enough since the more cases are available about the problem the more accurate will be the classification model. Applications based on supervised classification methods are very common as, in general, any complex system can be monitored obtaining data that are characterized and used to train a model able to classify new data based on previous evidences.

3.3 Classification Algorithms

According to [9], six of the most popular supervised data mining algorithms have been tested in present study. They are the following:

1. Statistical learning methods. Naïve Bayes [10]: is a Bayesian algorithm built by the assumption of the conditional independence among the predictor variables, given the class value.
2. Instance-based learning methods. *k*-Nearest Neighbour [11]: lazy-based learning method that given a new case finds the closest group of *k* cases in the training set, classifying the new case based on the predominance of a particular class in this neighborhood of size *k*.
3. Support Vector Machines, SVM [12]. This algorithm builds linear functions, $f(x)$, or hyperplanes, which separate the different classes of the training data set; thus, classification of a new case is made by testing the sign of this function.
4. Artificial Neural Networks. RBF Network [13]: typically is a three-layer feedback network composed by one unique hidden layer with radial basis functions as activation functions and the input and output layers.
5. Logic based algorithms. C4.5 algorithm [14] is an extension of ID3 algorithm [15] that builds a classification tree based on gain ratio criteria, defined as $I(X_i, C)/H(X_i)$. The algorithm also includes a pruning of induced tree, based on a test of hypothesis.
6. Rule induction methods. RIPPER algorithm [16] (Repeated Incremental Pruning Produce Error Reduction), is a rule induction algorithm that is an extension of IREP algorithm [17] (Incremental Reduced Error Pruning) and that consists of learning rules obtained by performing a process of repeated growing and pruning.

3.4 Validation Method

The validation method used in present work is the 10-fold cross validation [18]. This method consists in dividing the data set $D = \{(x_1^1,...,x_n^1,c^1),...,(x_1^N,...,x_n^N,c^N)\}$ in f disjoint subsets of approximately the same size $S_1,\ldots,S_i,\ldots,S_f$. For each subset S_i the following procedure is repeated: S_i is considered the test sample, so the training sample will be composed of the $f-1$ other samples $S_1,\ldots,S_{i-1},S_{i+1},\ldots,S_f$, and is used to build the decision function d; classes from S_i experiments are predicted by using d. This procedure is repeated for $i = 1,\ldots,f$.

In our case, in order to validate the algorithms tested, a value of f = 10 is established; though there exists other values commonly used, i.e. f = n. When this occurs, the method is named *leave-one-out cross validation*. Compared with other error estimation methodologies, this method provides an estimator with little bias but a lot of variance.

3.5 Multiclassifier Approach

Having a data set $D = \{(x_1^1,...,x_n^1,c^1),...,(x_1^N,...,x_n^N,c^N)\}$, for each classifier mentioned in Section 3.3 a classification model is obtained and the label distribution for each test case $x = (x_1^{'},...,x_n^{'})$ is computed. Given these inputs, a multiclassifier approach is designed given three different kinds of experiments:

- First experiment: confidence voting [19]. The multiclassifier will predict the label, *y'*, of each test case, *x*, based on the sum of the label distributions obtained by each classifier:

$$\forall_x\, y' = \sum_{c_i \in C(i=1,...,6)} \left(dist(c_i) * I(v = y_i) \right) . \tag{1}$$

  In (1) v is a class label, y_i is the class label for the ith classifier tested $c_i \in C(i = 1,\ldots,6)$, $dist(c_i)$ is the distribution that c_i obtained for class label v and I $(\cdot)$ is an indicator function to modify the sign of the confidence that returns the value 1 if its argument is true and -1 otherwise; a positive sign implies a correct prediction, whereas a negative sign implies a wrong prediction.
- Second experiment: weighted voting [20]. Given the class distribution, a weighting of the classifiers based on its accuracy is performed. Prediction made by the multiclassifier will be based on the sum of such weights.

$$weight(c_i,c_j) = \begin{Bmatrix} 1 & if\ dist(c_i) > dist(c_j) \\ -1 & if\ dist(c_i) < dist(c_j) \end{Bmatrix} * I(v = y_i) \ ; \tag{2}$$

$$\forall_x\, y' = \sum_{i,j=1,...,6}^{i \neq j} weight(c_i,c_j) . \tag{3}$$

  In (2) v is a class label, y_i is the class label for the ith classifier tested $c_i \in C(i = 1,\ldots,6)$ and I $(\cdot)$ is an indicator function that returns the value 1 if its

argument is true and -1 otherwise; a positive sign implies a correct prediction, whereas a negative sign implies a wrong prediction.

- Third experiment: majority voting [21]. Once the list of labels for each test case is obtained by each classifier, the multiclassifier will predict the label, *y'*, of each test case, *x*, based on the majority label; the tie case is solved randomly, providing an arbitrary solution.

$$\forall_x y' = \arg\max_v \sum_{x_i, y_i \in D_i} I(v = y_i) \ . \tag{4}$$

In (4) v is a class label, y_i is the class label for the ith classifier tested $c_i \in C(i = 1,\ldots,6)$, and I $(\cdot)$ is an indicator function that returns the value 1 if its argument is true and 0 otherwise.

The idea that is beyond these experiments is based on *Voting* concept, which is one of the simplest procedures when combining different classifier outputs in a vote-based framework: having $C_1, \ldots, C_N$ the set of classification models induced by a total of N different learning algorithms $L_1, \ldots, L_N$ and a database D with characteristic vectors, in order to classify a new instance the classifiers $C_1, \ldots, C_N$ will be asked to get the class value they predicted.

Into present work voting criteria is enhanced by computing distributions of class probabilities and thus obtaining a vector of degrees of confidence for all considered class labels. Those degrees of confidence will be the input of the multiclassifier approach, which will estimate the class value considering the three different experiments mentioned above.

4 Experimental Results

Experiments accomplished in this study envisage a total of 313 drilling processes using two different workpieces made of the same material, aluminium 7075. They are distributed as follows: 115 drilling processes related to first workpiece and 198 drilling processes related to second workpiece. All holes have been drilled using a hard metal drill of 10 millimetres of diameter. Machining parameters considered have been the following: 200 m/min. of cutting speed and 0.3 millimetres of advance/revolution.

The same training data and the same features are used to obtain each classification model, applying the 10-fold cross-validation method. Results obtained from this analysis are illustrated in table format, containing the accuracy of each isolated classifier and the multiclassifier approach, regarding the three different distribution-based experiments performed. The meaning of each numeric value in the tables is as follows:

- *Correctly Classified Instances*: number of correctly classified instances from data.
- *Incorrectly Classified Instances*: number of incorrectly classified instances from data.
- *Precision:* estimation of the accuracy of the classifier provided that drill wear has been predicted.

$$precision = tp/(tp + fp) \ . \tag{5}$$

In (5) *tp* is the rate of true positives predictions and *fp* is the rate of false positives predictions.

- *Recall:* estimation of the ability of a classifier to select instances that are related to drill wear from the data set.

$$recall = tp/(tp + fn) \ . \tag{6}$$

In (6) *tp* is the rate of true positives predictions and *fn* is the rate of false negative predictions.

- *F-Measure:* the weighted harmonic mean of precision and recall.

$$F - Measure = \frac{2 \cdot precision \cdot recall}{(precision + recall)} \ . \tag{7}$$

4.1 Burr detection

A total of 104 holes have been analysed, after performing a simple sample selection process consisted in filtering samples that contained incorrectly measured signals. Discretization of continuous burr values measured has been established as follows, according to the values obtained and under the supervision of drilling process experts:

- If burr level was under 150 microns, discrete value has been set to "low"
- If burr level was between 150 and 300 microns, discrete value has been set to "medium"
- If burr level was over 300 microns, discrete value has been set to "high"

In the following tables results obtained by each isolated classifier tested are shown, based on the criteria mentioned above (see Table 3 and Table 4).

Table 3. Results obtained from Naïve Bayes, *k*-NN and SVM

	Naïve Bayes	*k-NN*	*SVM*
Correctly Classified Instances	71 (68.3 %)	77 (74 %)	78 (75 %)
Incorrectly Classified Instances	33 (31.7 %)	27 (26 %)	26 (25 %)
Precision	0.28	0.31	0.18
Recall	0.54	0.36	0.18
F-Measure	0.37	0.32	0.18

Table 4. Results obtained from RBF Network, C4.5 and RIPPER

	RBF Network	*C4.5*	*RIPPER*
Correctly Classified Instances	85 (81.6 %)	82 (78.9 %)	84 (80.8 %)
Incorrectly Classified Instances	19 (18.4 %)	22 (21.1 %)	20 (19.2 %)
Precision	0.66	0.33	0.43
Recall	0.36	0.27	0.27
F-Measure	0.46	0.3	0.32

RBF Network obtained the highest average correct classification rate: 81.6%. In the case of the multiclassifier approach, as can be seen in Table 5, the burr prediction accuracy is, in general, slightly improved. *Naïve Bayes* was the classifier that presented the lowest percentage of goodness, 68.3%, much lower than percentages obtained by the three voting experiments.

Table 5. Final results obtained by applying the multiclassifier approach

	Confidence voting method	*Weighted voting method*	*Majority voting method*
Correctly Classified Instances	85 (81.6 %)	86 (82.7 %)	87 (83.6 %)
Incorrectly Classified Instances	19 (18.4 %)	18 (17.3 %)	17 (16.4 %)
Precision	0.33	0.4	0.44
Recall	0.27	0.36	0.36
F-Measure	0.3	0.38	0.4

Regarding the experiment related to confidence voting method, it is important to point out the information gain that implies working with class label distributions. This is really interesting from the point of view of an expert on the application field, providing an estimation of reliability for each prediction made.

4.2 Roughness Detection

Regarding roughness detection problem, 66 holes were analysed after performing a sample selection process as it was made in the case of burr detection problem. Resulting continuous roughness values were discretized as follows, according to the values obtained and under the supervision of drilling process experts:

- If roughness values were under 2 microns, discrete value was set to “normal”
- If roughness values were equal or over 2 microns, discrete value was set to “high”

Results obtained from roughness analysis with regards to each isolated classification model are shown in the following tables taking into account previously defined criteria (see Table 6 and Table 7).

Table 6. Results obtained from Naïve Bayes, *k*-NN and SVM

	Naïve Bayes	*k-NN*	*SVM*
Correctly Classified Instances	47 (71.2 %)	45 (68.2 %)	41 (62 %)
Incorrectly Classified Instances	19 (28.8 %)	21 (31.8 %)	25 (38 %)
Precision	0.58	0.54	0.68
Recall	0.56	0.61	0.38
F-Measure	0.57	0.56	0.49

Table 7. Results obtained from RBF Network, C4.5 and RIPPER

	RBF Network	*C4.5*	*RIPPER*
Correctly Classified Instances	42 (63.6 %)	46 (69.7 %)	43 (65.2 %)
Incorrectly Classified Instances	24 (36.4 %)	20 (30.3 %)	23 (34.8 %)
Precision	0.46	0.63	0.5
Recall	0.35	0.3	0.3
F-Measure	0.40	0.41	0.37

As can be seen in tables showed above, the isolated classifier that best predicted the roughness levels was *Naïve Bayes*, obtaining 71.2% of well classified samples. This value coincides with performance obtained by the multiclassifier approach when using weighted voting method (see Table 8). Performance obtained from confidence and majority voting methods was slightly higher, 74.1%. The worst percentage of correctly classified instances was obtained by *SVM* algorithm: 62%.

Table 8. Final results obtained by applying the multiclassifier approach

	Confidence voting method	*Weighted voting method*	*Majority voting method*
Correctly Classified Instances	49 (74.1 %)	47 (71.2 %)	49 (74.1 %)
Incorrectly Classified Instances	17 (25.9 %)	19 (28.8 %)	17 (25.9 %)
Precision	0.69	0.63	0.69
Recall	0.48	0.38	0.48
F-Measure	0.56	0.46	0.56

5 Conclusions

From experimental results presented in this paper, it can be concluded that multiclassifer approach predictions are, in general, more accurate than predictions made by isolated classifiers. However, it is important to stress that in some cases an individual classifier is able to guarantee a good performance; therefore, a further similarity analysis among classifiers should be accomplished in order to study this interesting behaviour. A balance between the expected accuracy of the classifier model and the time and resources needed should be also established in advance, in order to estimate the usefulness of having a more accurate combination of several classification models or just one less accurate but faster and lighter model. This is strongly recommended when data sets are composed of a great number of instances or when the quality of data is poor.

Regarding the drill wear prediction problem, the following conclusions are drawn:

- Drill wear prediction can be characterized by the presence of burr and roughness on holes drilled with the support of data mining techniques
- When detecting burr, the multiclassifier approach experiment related to majority voting obtained the highest average correct classification rate: 83.6 %

- When detecting roughness, the multiclassifier approach experiments related to confidence values and majority vote of class labels obtained the best accuracy: 74.1 %
- In general, experimental results demonstrate that multiclassifier approach is able to slightly reduce the number of false positives obtained by most of the classifiers individually, which can lead to an incorrect drill wear substitution strategy
- By means of classification algorithms it is possible to avoid poor drilling quality due to drill wear in an efficient and automatic way

Future works in relation to this study will be oriented not only to predict burr and roughness more accurately, but also to infer relations between both parameters and how they make influence on drill wear over time, by considering different predictor variables and multiclass models. Different classification models can also be learnt for different feature recognition tasks in order to improve the accuracy of the multiclassifier, which will combine predictions made for each feature separately. The study must be also extended to different application sectors and domains.

References

1. Emmanouilidis, C., et al.: Flexible software for condition monitoring, incorporating novelty detection and diagnostics. Computers in Industry 57, 516–527 (2006)
2. Cassady, C.R., Schneider, K., Yu, P.: Impact of Maintenance Resource Limitations on Manufacturing System Productivity. In: Proceedings of the Industrial Engineering Research 2002 Conference (2002)
3. Muller, A., et al.: Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial system. Reliability Engineering & System Safety 93, 234–253 (2008)
4. van Erp, M., Vuurpijl, L., Schomaker, L.: An overview and comparison of voting methods for pattern recognition. In: Proc. of the 8th IWFHR, pp. 195–200 (2002)
5. Ferreiro, S., Arana, R., Aizpurua, G., Aramendi, G., Arnaiz, A., Sierra, B.: Data Mining for Burr Detection (in the Drilling Process). In: Proceedings of the 10th International Work-Conference on Artificial Neural Network, Salamanca, Spain, June 10-12 (2009)
6. Jolliffe, I.T.: Principal Component Analysis. Springer Series in Statistics, 2nd edn., XXIX, 487 p. 28 illus. Springer, NY (2002) ISBN 978-0-387-95442-4
7. Carrascal, A., Díez, A., Azpeitia, A.: Unsupervised Methods for Anomalies Detection through Intelligent Monitoring Systems. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS, vol. 5572, pp. 137–144. Springer, Heidelberg (2009)
8. Kotsiantis, S.B.: Supervised Machine Learning. A Review of Classification Techniques, Informatics 31, 249–268 (2007)
9. Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G., Ng, A., Liu, B., Yu, P., Zhou, Z.-H., Steinbach, M., Hand, D., Steinberg, D.: Top 10 algorithms in data mining. Knowledge and Information Systems 14(1), 1–37 (2008)
10. Rish, I.: An empirical study of the naive Bayes classifier. In: IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence (2001)
11. Aha, D., Kibler, D.: Instance-based learning algorithms. Machine Learning 6, 37–66 (1991)

12. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines. Cambridge University Press (2000)
13. Powell, M.J.D.: Radial basis functions for multivariable interpolation: a review. In: Mason, J.C., Cox, M.G. (eds.) Algortithms for Approximation on Functions and Data, pp. 143–167. Oxford University Press, Oxford (1987)
14. Quinlan: C4.5. Programs for Machine Learning. Morgan Kaufmann (1993)
15. Quinlan, J.R.: Induction of decision trees. Machine Learning 1(1), 81–106 (1986)
16. Cohen, W.W.: Fast effective rule induction. In: Proceedings of the Twelfth Internacional Conference on Machine Learning (1995)
17. Fürnkranz, J., Widner, G.: Incremental Reduced Error Pruning. In: Proceedings of the Eleventh Internacional Conference on Machine Learning (1994)
18. Stone, M.: Cross-validatory choice and assessment of statistical predictions (with discussion). Journal of the Royal Statistical Society, Series B 36, 111–147 (1974)
19. Kittler, J., Hated, M., Duin, R.P.W., Matas, J.: On Combining Classifiers. IEEE Transactions PAMI 20, 226–239 (1998)
20. Verma, B., Gader, P., Chen, W.: Fusion of multiple handwritten word recognition techniques. Patt. Recog. Lett. 22, 991–998 (2001)
21. Dietterich, T.G.: Machine learning research: Four current directions. AI Magazine 18(4), 97–136 (1997)

Measuring the Dynamic Relatedness between Chinese Entities Orienting to News Corpus

Zhishu Wang, Jing Yang, and Xin Lin*

Department of Computer Science and Technology,
East China Normal University Shanghai, China
zhshwang@ica.stc.sh.cn
{jyang,xlin}@cs.ecnu.edu.cn

Abstract. The related applications are limited due to the static characteristics on existing relatedness calculation algorithms. We proposed a method aiming to efficiently compute the dynamic relatedness between Chinese entity-pairs, which changes over time. Our method consists of three components: using co-occurrence statistics method to mine the co-occurrence information of entities from the news texts, inducing the development law of dynamic relatedness between entity-pairs, taking the development law as basis and consulting the existing relatedness measures to design a dynamic relatedness measure algorithm. We evaluate the proposed method on the relatedness value and related entity ranking. Experimental results on a dynamic news corpus covering seven domains show a statistically significant improvement over the classical relatedness measure.

Keywords: Dynamic relatedness measure, Co-occurrence statistics, News corpus.

1 Introduction

With the rapid development of information technology and Internet, a great deal of information emerges and goes out of the date in Internet. In such a dynamic environment, the relatedness of some entities changes over time. For example, the entity-pair (*Japan*, *Earthquake*) are not similar or related on the semantic level, but they are strong related after a big earthquake happened in Japan in March 11, 2011. As time goes on, people pay less and less attention to the Japanese earthquake events, entity-pair (*Japan*, *earthquake*) slowly fade out of people's perspective, the degree of relatedness between them is also gradually decreased. We call this kind of relatedness the dynamic relatedness. In the Web entity retrieval (WER), it is very important to capture the dynamic relatedness between entity-pairs for improving users' search experience. However, the current relatedness calculation models only define the relatedness as a static value and usually calculate it based on the similarity measure, which can't meet that demand.

* Corresponding author.

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 631–644, 2012.

We propose a new method that specially measures dynamic relatedness between entity-pairs. Firstly, we use co-occurrence statistics method to mining the co-occurrence information of entities from the news texts. Secondly, we integrate time factor into a classical relatedness measure explained in section 3.3. Finally, given an entity-pair,(e_1, e_2), we design a function, $rel(e_1, e_2, t)$ that returns a relatedness score at time t. In this paper, entities include four types of Chinese words, including company names, personal names, locations and proper names.

Our contributions are summarized as follows:

1. We induce development law of dynamic relatedness between entity-pairs in news documents, and take this development law as the basis to continue our further works.
2. We present an efficient dynamic relatedness measure algorithm to compute the dynamic relatedness between entity-pairs. The proposed relatedness measure algorithm aims at the dynamic corpus of text, but not the closed and static text collection, thus the algorithm is more generic and the results are more precise.
3. We manually created a dataset of news texts covering seven domains (our dataset is explained in section 4.1) by crawling news from the Web, the total size of the dataset is1198M. We use this dataset to simulate a dynamic corpus of texts. We compare the proposed method against the classical relatedness measure (*Generalized*) [10] on this dataset. Experimental results show that the proposed method significantly outperforms the *Generalized* method.

The rest of this paper is organized as follows. Section 2 discusses some related work. Our proposal of dynamic relatedness measure between entity-pairs appears in section 3. Experimental results can be found in Section 4, and conclusions appear in Section 5.

2 Related Work

At present, the main tasks of word relatedness focus on the research of word semantic relatedness. There are two types of calculation models for word semantic relatedness, the first is to calculate degree of relatedness among words according to the language knowledge and classification system; the second is statistics method, which usually uses learning model acquiring law of word co-occurrence to calculate the degree of relatedness among words. Concept statistics, parameter estimation and feature acquirement are frequently used as the learning model.

On the classification system methods, Agrawal et al. [1] present a relatedness computation algorithm between words according to each word's category, which essentially is a kind of fuzzy collocation relationship of words. Silberschatz et al. [2] use semantic similarity and the semantic information from HowNet to calculate semantic relatedness. They take the similarity calculation as foundation, and integrate three kinds of influence factors into semantic relatedness calculation, the influence factors are hyponymy of HowNet, relationships between instances and vertical linkage between words. Liujun et al. [3] build Wikipedia category tree and present each word by a Wikipedia category vector, finally a Wikipedia dictionary which contains all

domains' knowledge is constructed. The Wikipedia dictionary is applied to semantic relatedness calculation. ZHU Jinwei et al. [4] leverage two kinds of language resources which are Tong YiCi CiLin and HowNet to measure the degree of relatedness between sentences. Though the relatedness measure is different from above methods, it is still based on the similarity and is not enough to consider the relationship between words, so there are still existing imperfect shortcomings on the relatedness measure. Except above mentioned problems, all methods mentioned above depend on a static and close knowledge base, it is difficult to update in time, as a result, the degree of relatedness are defined as a static value. Moreover, words are not well covered by manually created dictionaries such as WordNet [5], especially to entities.

Table 1. Well-known measures for co-occurrence

Measure name	Measure Formulation
The association strength	C_{ij}/S_iS_j
The cosine	$C_{ij}/\sqrt{S_iS_j}$
The Inclusion index	$C_{ij}/\min(S_iS_j)$
The Jaccard index	$C_{ij}/S_i + S_j - C_{ij}$

While employing statistics method to compute the degree of relatedness from a large-scale corpus can avoid above shortages. The most basic method is to measure the degree of relatedness through counting the relative co-occurrence frequency of words in the same text window. It is considered that the larger the frequency is, the closer the relatedness is. For example, CHEN Xiaoyu et al. [6] use the number of web pages which are returned by Web search engine to measure semantic relatedness between word-pair. It is supposed that the more the number of pages which contain word-pair appears, the closer the strength of relatedness is. Liu Jinpan et al. [7] summarized several the most frequently used relatedness measures which are based on co-occurrence statistics in computer science, as show in table 1. From the table 1, we can see that different methods which are based on co-occurrence statistics have been proposed in the past; however, current relatedness measures lack some desirable properties for the dynamic relatedness between entity-pair.

Besides, Google Insights developed by Google company can presents the dynamic phenomenon that varies with time between two words. As show in Figure 1, Google Insights is used for analyzing the queries of users and showing the attention for a certain phase via the search volume. The higher attention indicates that the terms contained in a query are searched by more users [8]. When we use word-pair (e.g. *Japan* and *Earthquake*) as a query to the Google Insights, the search volume of the

word-pair will be returned quickly by Google Insights. Let us suppose that larger frequency of word-pair means that the degree of relatedness between word-pair is larger. In this condition, we can use Google Insights for measuring the dynamic relatedness between word-pair. However, the applications of Google Insights have some limitations:

1. The data of Google Insights comes from a large number of query logs and these huge data must be analyzed. What's more, it is difficult to get these data by ordinary users.
2. The query logs require a long time to accumulation, thus Google Insights cannot return results for all the queries.

These above limitations hinder Google Insights from widely using. In this study, we explore a new relatedness measure which is more generic and the results are more precise.

Fig. 1. Attention of "Japan" and "Earthquake" changes with time. [9]

3 Methods

3.1 Outline

In this study, dynamic relatedness measure quantifies the degree of relatedness which changes over time between entity-pairs, considering not only similarity but also any possible relationship between them. Designing the dynamic relatedness measure algorithm consists of three steps. Firstly, count the total frequency of each entity-pair in news texts and induce the development law of dynamic relatedness. Secondly, decompose the development process of dynamic relatedness into two sub-processes, and then solve each sub-process respectively. Finally, define the formula of dynamic relatedness measure via merging the sub-processes appropriately. The subsequent sections will explain these steps in detail.

3.2 Development Law of Dynamic Relatedness between Entity-Pair

In order to compute the dynamic relatedness between entity-pair, it is very helpful to know the trend of dynamic relatedness between entity-pairs. Classical relatedness measure which is based on co-occurrence statistics supposes that two words occurrence in the same text window (e.g. a sentence), certain relatedness must exist them [6]. According to this hypothesis, if two entities occurrence more frequently in the same text window during a specified period of time, they are more related to each other. The co-occurrence frequency of two entities can get quickly by statistics. Based on this consideration, we count the co-occurrence frequencies of 500 entity-pairs (they are all related) on everyday news.

There are three observations on the development process of co-occurrence frequency for entity-pair, which can be concluded as:

1. The co-occurrence frequency of each entity-pair through the same development process from increasing phase to stable phase and then to decreasing phase. Although there are some ups and downs in the process, the general trends will not change.
2. When the co-occurrence frequency of entity-pair comes to the stable phase, it will start with decrease, and the rate of decrease is fast at the beginning and then become slow later (based on this theoretic, it produces the attenuation process, which will be explained in section 3.3).
3. The co-occurrence frequency of entity-pair may become increasing suddenly during the decreasing phase and when it goes to the stable phase, the trends will repeat the previous process (on this theory foundation, we also further discuss the impact process in section 3.3).

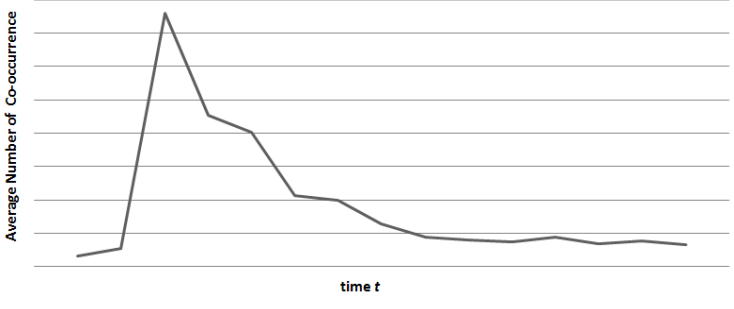


Fig. 2. Development curve of the average co-occurrence frequency for entity-pairs

As showed in Figure 2, the co-occurrence frequency of entity-pair experiences single development process which includes increasing phase, stable phase and decreasing phase. Since if entity-pair co-occur more times in the same text window during a specified phase of time, there will be larger degree of relatedness between them. The development trend of dynamic relatedness between entity-pair is consistent with the development trend of co-occurrence frequency of entity-pair. Naturally, we use the

development law of co-occurrence frequency of entity-pair as the development law of dynamic relatedness between entity-pair. Based on the basic principle of the development law, we will further discuss the development process of dynamic relatedness in section 3.

3.3 Development Process of Dynamic Relatedness

According to the Figure 2 and the development law of dynamic relatedness between entity-pairs, we can see that the changes of dynamic relatedness between entity-pairs mainly consist of the increase phrase and decrease phrase. Study when the dynamic relatedness is increasing and when it is decreasing, and the reasons for the increase and decrease are become the key parts of research. We decompose the development process of dynamic relatedness between entity-pair into two sub-processes, namely, the attenuation process and the impact process. Finally, we will use the two sub-processes for defining the degree of dynamic relatedness between entity-pair.

Attenuation Process. For a given entity-pair, even though the initial degree of relatedness between them is large, if their co-occurrence frequency decreases, the degree of relatedness between them will fall gradually. We call this process as attenuation process. In order to accurately express how much the degree of relatedness have decreased, based on the decrease part of curve in the Figure 2, we use exponential function to fit the real decrease curve, as show in Figure 3.

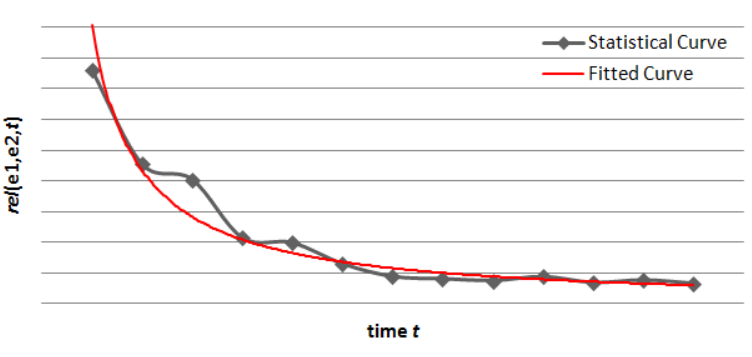


Fig. 3. Schematic diagram of a single attenuation process

If t' is initial time of the attenuation process, we denote the degree of relatedness between an entity-pair (e_1, e_2) by $rel_{remains}(e_1, e_2, t)$. Using exponential regression we can get:

$$\frac{rel_{remains}(e_1,e_2,t)}{dt} = -\beta rel_{remains}(e_1, e_2, t) \quad (1)$$

Here, β is the natural attenuation coefficient. Integrating the equation (1) , we can get as follows,

$$rel_{remains}(e_1, e_2, t) = rel(e_1, e_2, t')e^{-\beta(t-t')} \quad (2)$$

As shown in equation (2), the decreasing quantity of degree of relatedness between entity-pair is determined by the natural attenuation coefficient β and the long of attenuation time (i.e. $t - t'$).

Impact Process. If an entity-pair no longer co-occurs or their co-occurrence frequency decrease, which implies people's concern becomes less, so the degree of relatedness between entity-pair will goes along with the attenuation process. During this phase, if the entity-pair co-occurs again, then the new co-occurrence frequency of the entity-pair will be considered as an external impact source which can reverse the attenuation process and make the degree of relatedness between the entity-pair increasing. It is called impact process. During this process, an impact exists whenever the external impact source exists, however, both actual demand and the time-consuming of impact process are considered, we simplify this question via setting a circle T for the impact process, namely, we take an impact every other temporal distance T into account. Consequently, the schematic diagram of a single impact process generally expressed as in Figure 4.

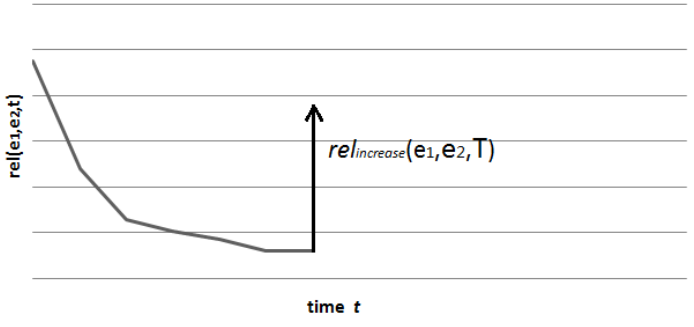


Fig. 4. Schematic diagram of a single impact process

As a statistical measure of co-occurrence, we compute the impact increment, $rel_{increase}(e_1, e_2, T)$, between an entity-pair (e_1, e_2) using a general co-occurrence measure formulations proposed by Ido Dagan [10]. So, the impact increment can be concluded as:

$$rel_{increase}(e_1, e_2, T) = \frac{2^{1/p} f(e_1,e_2,T)}{(f(e_1,T)^p + f(e_2,T)^p)^{1/p}} \quad (3)$$

Where, $f(e_1, T)$, $f(e_2, T)$ separately denote the frequency of entities e_1, e_2 appearing alone during the circle T. The value of $f(e_1, e_2, T)$ denotes the co-occurrence frequency of the entity-pair (e_1, e_2) during the circle T. p is the adjustable parameter and its value scope is the real set. We can get the co-occurrence measure formulations mentioned in section 2 through regulating the value of p [7], this is why we choose *Generalized* to compute the value of impact increment. Equation (3) shows that the

value of impact increment between an entity-pair is determined by the co-occurrence frequency and the single frequency.

Formula. During the development process of dynamic relatedness, we should take the increase into account because of the effect of the impact process and the decrease due to the attenuation process. Since our dynamic relatedness measure integrates an important time factor, we must update the former value of relatedness at all times. Actually, we take a simple strategy which updates former data every other circle T and consider once attenuation process and impact process at the same time to make our algorithm highly effective. In that way, the degree of dynamic relatedness, $rel(e_1, e_2, t)$, between an entity-pair (e_1, e_2) can be expressed by equations (2) and (3) for nth attenuation process and impact process.

$$\begin{aligned} rel(e_1, e_2, t) &= rel_{remains}(e_1, e_2, t) + rel_{increase}(e_1, e_2, t) \\ &= rel(e_1, e_2, t')e^{-\beta(t-t')} + \frac{2^{1/p} f(e_1, e_2, T)}{(f(e_1, T)^p + f(e_{21}, T)^p)^{1/p}} \\ &= rel(e_1, e_2, t - kT)e^{-\beta kT} + \frac{2^{1/p} f(e_1, e_2, T)}{(f(e_1, T)^p + f(e_2, T)^p)^{1/p}} \end{aligned} \tag{4}$$

Here, $k = \left\lfloor \frac{t-t'}{T} \right\rfloor$, is the number of attenuation process after last impact process.

Our dynamic relatedness measure algorithm has three parameters (ca. β, p and T). We set the values of these parameters experimentally, as explained later in section 4.2. It is noteworthy that the proposed dynamic relatedness measure algorithm considers both co-occurrence frequency of entity-pair and time factor, and is not limited to compute the relatedness degree between entity-pair depending on the single co-occurrence frequency. Moreover, the proposed algorithm does not need to update all the time, but set an optimal renewal cycle. The experiment shows that the speed of computation of algorithm can be improved effectively on the basis of ensuring high accurate rate for the measured relatedness.

4 Experiments

4.1 Dataset

In order to evaluate the proposed dynamic relatedness measure, we have created a dataset of text. Our dataset contains seven domains (News, Sports, Finance, Tech, Military, Auto and Entertainment). We selected these domains because they almost cover all aspects of life. We crawled news during March to September, 2011, from sina.com (http://rss.sina.com.cn/) and extracted the titles, texts and issue time from the crawled news and saved them in the order of time. The dataset is used for simulating a dynamic corpus. In Table 1, we show some key information of the dataset.

Table 2. Overview of the dynamic relatedness dataset

Domain	Number	Size(MB)	Relevant URL
News	43 331	166	http://rss.sina.com.cn/news/index.shtml
Sports	63 648	298	http://rss.sina.com.cn/sports/index.shtml
Finance	83 385	321	http://rss.sina.com.cn/finance/index.shtml
Tech	50 685	205	http://rss.sina.com.cn/tech/index.shtml
Military	10 681	46	http://rss.sina.com.cn/jczs/index.shtml
Auto	5 632	24	http://rss.sina.com.cn/auto/index.shtml
Entertainment	38075	138	http://rss.sina.com.cn/ent/index.shtml
Total	295 437	1198	

4.2 Evaluation of Metrics and Methods

In the evaluation of traditional semantic relatedness, the relatedness value and Spearman rank correlation coefficient are usually used for evaluating the quality of algorithms. However, the traditional semantic relatedness measures quantify the degree of relatedness as a static value, the proposed dynamic relatedness measure computes the degree of relatedness as a variable which changes with time, it is not acceptable for human judgment. Therefore, how to evaluate the dynamic relatedness measure has become a difficult problem.

We have introduced about Google Insights in the section 2. Although Google Insights have some limitations by generally applied, the advantages as follows:

1. The data of Google Insights derives from real users and directly reflects the demand of users.
2. The data of Google Insights can provide strong evidences for the degree of relatedness between an entity-pair.

So, let us intend to evaluate our proposed dynamic relatedness measure judging by the results returned by the Google Insights, namely, the differences between the results of the relatedness measures and Google Insights are compared and analyzed so as to determine whether our method is available. The smaller the value of difference is, demonstrating the more accuracy that the corresponding method is. Computing the value of difference can be defined as:

$$M_{difference} = \frac{\sum(M_r - M_g)^2}{N} \tag{5}$$

Here, M_r denotes the value of relatedness measure. M_g denotes the value from Google Insights. M_r and M_g are in symmetrical relationship. N denotes the total number of M_r or M_g.

In the accuracy evaluation, we first directly evaluate the relatedness value between single entity-pair during a specified period of time. A good relatedness measure must assign higher relatedness values to entity-pair with strong relevant relations. The former task does not evaluate the relative rankings of relatedness values. We use the equation (5) again to evaluate the top most related k entities to a given entity. In order to discuss how large of the difference value would demonstrate that our proposed method is available and high performance. We carried out two groups of experiments on the same dataset which explained in section 4.1. The first group used our proposed dynamic relatedness measure (Dynamic) for computing the relatedness value between the entity-pair and the other employed the classical relatedness calculation (Generalized) mentioned in [10].

Evaluation Method for Relatedness Value. In order to measure the quality of comparative methods, we directly evaluated the relatedness value between entity-pairs during a specified period of time. To be specific, firstly, the relatedness values are computed by the methods of Dynamic, Generalized and Google Insights respectively for the same entity-pairs during the same phase of time. Secondly, two difference values are calculated between the results of Dynamic and Google Insights and between the results of Generalized and Google Insights. Finally, the two difference values are compared, if the difference value of between Dynamic and Google Insights is smaller than that between Generalized and Google Insights, the Dynamic is more accurate than Generalized, and vice versa. We downloaded CSV documents for each entity-pair from Google Insights to compute the relatedness value. The CSV documents contain the attention of entity-pair and it ranges from 0 to 100. So, after shrinking it other 100 times, we can use it for computing the difference value.

Fig. 5. Sample of the results returned by Google Insights [9]

Evaluation Method for Entity Ranking. Entity ranking ranks related entities to a given entity according to their degree of relatedness. Evaluation of entity ranking aims to indirectly verify the accuracy of methods by comparing the relatedness values

between many entities and a given entity at some point. Similarly, we first get three groups of entity ranking results using Dynamic, Generalized and Google Insights respectively, and then, calculate two difference values by equation (5). Finally, the two difference values are compared, the more accurate method corresponds to the small difference value. For example, given a entity "*Kobe*", we select the top most related entities to construct queries (e.g. "*Kobe Lakers*", "*Kobe Jordon*", "*Kobe Basketball*", "*Kobe Wade*", "*Kobe James*"), and input these queries into Google Insights at the same time, the results will be returned as show in Figure 6. We rank these related entities by the values of attention which they have and get a set of ordinal. For example, the result of entity ranking is: "Lakers, Jordon, Basketball, James, Wade" by the point (a) in Figure 6, thus, the sets of order numbers are "1, 2 ,3, 4, 5". Similarity, we rank the related entities to the same given entity by Dynamic and Generalized respectively, and then acquire two set of numbers which correspond to the order number of each entity in the results of Google Insights according to the sort order. Here, the two set of numbers are "2, 1, 4, 5, 3" and "1, 2, 5, 4, 3". Finally, the difference values can be computed by equation (5).

Parameters Optimization. Before the experiments are carried out, we must choose suitable parameters p, T, and β to evaluate our work. The value of p in Liu Jinpan's method is 50, so we set $p = 50$. By using the special values for equation (2), we get the value of natural attenuation coefficient is 0.14943, so we set $\beta = 0.15$. Moreover, Figure 7 shows the relationship between T and average difference of entity ranking. As can be seen from the Figure 7, the average difference grows linearly when the value of T approximately than 11, and the smallest value of average difference at $T \approx 9$. In practice, if T is too small, method need to update the old data frequently, so it is time-consuming. If T is too large, method can decrease the times of renew but effects the timeliness of information. We set $T = 9$ to balance time-consuming and timeliness of information.

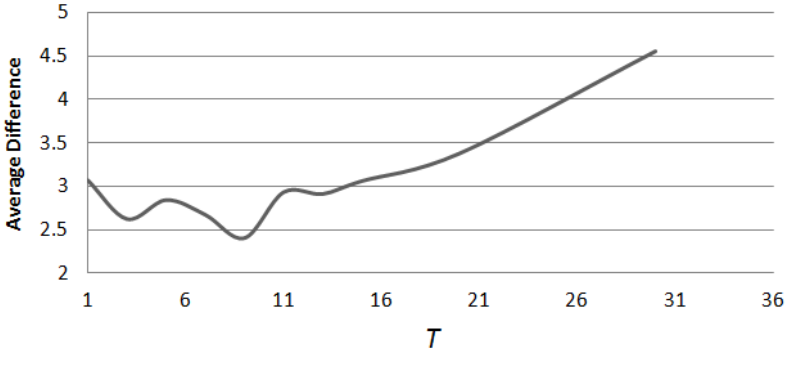


Fig. 6. The relationship between T and average difference

4.3 Results and Analysis

Relatedness Value Accuracy. Since considering Google Insights cold return corresponding comparing data, we selected some representative entities from person names, proper names and organization names as the subjects of our experiment. In this experiment, we selected entities (e.g. "*Junhui Ding*", "*Earthquake*", "*Google*" etc.) as queries, and then selected the top most related 10 entities to each query. The related entities and corresponding queries formed total of 100 entity-pairs. We show some comparing results between two different methods (Dynamic and Generalized) in Table 3. The proposed method reports smaller overall average difference than the classical method. Ordinarily, the generalized method is accepted in common usage. So, the average difference value of 0.12 (0.12<0.19) is small and which reveals that our proposed method is available.

Table 3. Simples of difference for relatedness values

Entity Pair		Time Phase	$M_{difference}$	
			Generalized	**Dynamic**
Junhui Ding	David Selby	April to May, 2011	0.163595	**0.118107**
Junhui Ding	Judd Trump	April 28 to May 4, 2011	0.219971	**0.219100**
Junhui Ding	Snooker	March to August, 2011	**0.103483**	0.115385
Junhui Ding	Billiard Ball	April to May, 2011	**0.152122**	0.170602
Junhui Ding	World Championship	April to May, 2011	0.344349	**0.122988**
Earthquake	Wenchuan	10 to 20 March, 2011	0.363885	**0.330599**
Earthquake	Yushu	March to April, 2011	0.375725	**0.340560**
Earthquake	Japan	10 to 20 March, 2011	0.157696	**0.131118**
Earthquake	Tsunami	10 to 20 March, 2011	0.285530	**0.194671**
Earthquake	Fukushima	10 to 20 March, 2011	0.047604	**0.028501**
Google	Microsoft	July to August, 2011	0.089791	**0.056991**

Table 3. (*Continued*)

Google	Apple	July to August, 2011	0.224577	**0.128638**
Google	Nokia	July to August, 2011	0.374441	**0.341046**
Google	Samsung	July to August, 2011	0.308716	0.254265
Google	Moto	14 to 20 August, 2011	0.223044	0.124710
Average Difference			0.193319	**0.121408**

Table 4. Queries for entity ranking

Query	Yao Ming, Google Earthquake, Junhui Ding, Bingbing Fan, Kobe, Obama, Escherichia Coli, Nokia

Entity Ranking Accuracy. We have selected some common entities as queries in Table 4. For each query, we selected the top most related 10 entities and formed totally of 90 entity-pairs. We used the evaluation method explained in section 4.2 to compare above mentioned two methods. In order to reduce the errors, we carried out the experiment for many times. The results of these experiments are showed in Figure 9. It can be seen from Figure 9, even for the same entity-pairs, the average differences are different at different time. However, from the overall trend of curve, the curve corresponding to Dynamic is always under the curve corresponding to Generalized and this trend appears stability. It reveals that our proposed method (Dynamic) superior to the other method (Generalized) and demonstrates our proposed method is higher accuracy on the computed relatedness values.

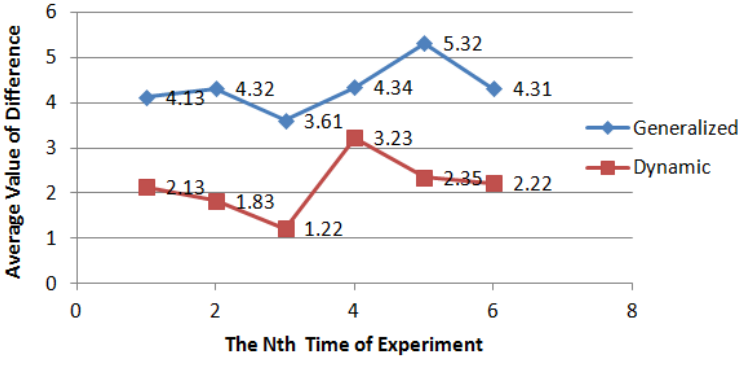


Fig. 7. Comparisons of average difference for entity ranking

5 Conclusions

We consulted the classical relatedness measure and took its principles as the basis of our works. We induced the development law of dynamic relatedness between entity-pair from everyday news. Given two entity-pair, a new dynamic relatedness measure was redefined to quantify the degree of relatedness between them. The proposed dynamic relatedness measure stresses on the dynamic for the degree of relatedness between entity-pair. Finally, we evaluated our propose algorithm directly (relatedness value) and indirectly (entity ranking), the experimental results show the new algorithm is effective.

Since our proposed method is synchronized to time, it needs to update the former computed relatedness values all the time. For huge data, this is time-consuming. We addressed this problem by setting an optimal renewal cycle T to balance time-consuming and timeliness of information. In our future work, the quicker algorithms will be studied to make our proposed algorithms higher efficient. Moreover, we intend to employ the proposed dynamic relatedness measure to directly retrieve a set of related entities for a given entity from the Web.

Acknowledgments. The authors wish to thank all members of our lab for discussing some issues about this paper, and our experiments for their helpful comments and suggestions. This work is supported by a grant from the Shanghai Science and Technology Foundation (No.10dz1500103, No.11530700300 and No.11511504000).

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Buneman, P., Jajodia, S. (eds.) Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C. (1993)
2. Silberschatz, A., Tuzhilin, A.: What makes patterns interesting in knowledge discovery systems. IEEE Transcations on Knowledge and Data Engineering 8 (1996)
3. Liu, J., Yao, T.-F.: Semantic Relevancy Computing Based on Wikipedia. Computer Engineering 36(19), 42–43 (2010)
4. Wettler, M., Rapp, R.: Computation of word associations based on the co-occurrences of words in large corpora, http://acl.ldc.upenn.edu/W/W93/W93-0310.pdf (accessed December 9, 2005)
5. Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Introduction to Wordnet: An on-line lexical database. International Journal of Lexicography 3, 238–244 (1990)
6. Chen, X.-Y., Guo, L., Fang, J.: Semantic relatedness based on searching engines. Computer Engineering and Applications 46(30), 128–130 (2010)
7. Liu, J., He, L., et al.: A specific word relatedness computation algorithm for news corpus. In: 2010 2nd International Workshop on Intelligent Systems and Applications, ISA (2010)
8. Baidu Wikipedia, http://baike.baidu.com/view/6306108.htm
9. Google Insights, http://www.google.com/insights/search
10. Dagan, I., Lee, L., Pereira, P.C.N.: Similarity-Based Models of Word Cooccurrence Probabilities. Maehine Learning (1999)

Prediction of Telephone User Attributes Based on Network Neighborhood Information

Carlos Herrera-Yagüe and Pedro J. Zufiria

Depto. Matemática Aplicada a las Tecnologías de la Información,
ETSI Telecomunicación, Universidad Politécnica de Madrid, Spain
carlos@hyague.es, pedro.zufiria@upm.es

Abstract This paper addresses the problem of predicting several attributes corresponding to telephone users, based on information gathered from the network which defines their communication patterns. Two approaches are compared which are grounded on machine learning techniques: the initial approach makes use of link information between two users, looking for the correlation between user attributes and communication patterns. The second approach exploits the network structure underlying the communication behavior of the user under study. Simulations show that the learning machines are able to extract network information to improve the attribute prediction capabilities.

1 Introduction

Communication records between human beings are probably one of the largest source of data generation nowadays. According to recent studies, every minute 370,000 Skype conversations, 198 million e-mails and over half million facebook comments are made worldwide[1]. In addition, only in the US, 2.4 billion calls and 6.1 billion SMSs take place daily[2].

This huge amount of information involves new challenges in the fields of data mining and machine learning. Classical approaches to the study of communication networks consist on aggregating information by user, such as the number of phone calls a user makes, or the time a user calls more often. However, recent works suggest that the aggregation of data may discard a crucial source of information and a new insight for a better characterization of systems: the network structure behind the connections. Many real world systems whose growth was not driven by any pre-existing blueprint have been proved to show non-trivial features when analyzed as a network. Examples include Internet routers [3], web hyperlinks, power grids, neurons connections on simple organisms [15], gene regulatory networks [2] and many others. Social interactions, which can be studied through the proxy of digital communication records, are no an exception for this. Social networks have been found to have a very high number of triangles (referred in literature as the clustering phenomenon), a short diameter and a very

[1] Source: http://www.go-globe.com
[2] Source: http://www.deadzones.com

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 645–659, 2012.

heterogeneous degree distribution: there are key nodes, called *hubs,* which have many edges adjacent to them [1]. Besides, it has been found that the structure has a very important influence on the events taking place in the network. For example, Granovetter's hypothesis made back in the 70s [5] turned out to be true according to a recent work by Onella et al. [10] using massive mobile phone data: most of social interaction events happen in small and dense zones of the graph.

The approach used in this work consists of aggregating information by links, and extracting features from both the dynamics of the events in each link and the network structure around the node (user) of interest. The precise problem which will be discussed in this article can be described easily looking at figure 1: given a number of nodes whose attributes are known (well-known nodes, white in the figure) and some links whose events are known, the problem consists of predicting the attributes of an opaque node (black in the figure).

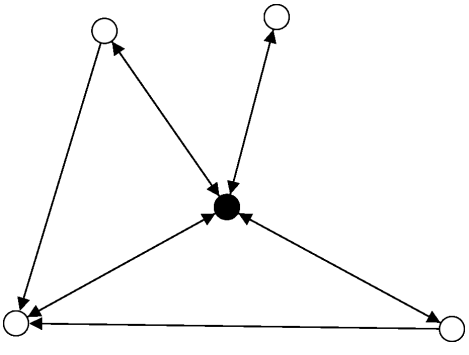


Fig. 1. Problem description: given a number of nodes whose attributes are known (well-known nodes, white) and some links whose events are also known, the attributes of an opaque node (black) must be estimated

2 Some Interesting Scenarios

The problem addressed here does model, in fact, a growing number of real world situations. In many communication services, gathering permission from a user allows the building of his full ego-centered network. This is so because a given link data set can be accessed provided either of the two users involved in the link has granted permission. Some of these situations are described here:

- Facebook applications: if an application is accepted by one user, this user becomes a well-known node, and all its neighbors become potential opaque nodes.
- Twitter private profiles: these profiles would be the opaque nodes, whose ego-networks can be built using their public profile neighbors.
- Mobile phone network: in mobile communications, carriers have information about their customers, and they also have communication records of any interaction between their customers and the rest of the users in the mobile

phone network. This way it is possible to build, using records from only one carrier, ego-networks where non-subscribers are the opaque nodes, and subscribers are the well-known nodes. This scenario will be the case test for this article.

3 Data Description and Preparation

Anonymized data for this article was provided by Orange, France Telecom Spain, consisting of two data-sets:

- Call Detail Records (CDR), contain information about customers interactions via phone calls and SMSs. For each interaction, two anonymized user identifiers and a time-stamp are provided. For phone calls, duration is also available. These data include interaction between subscribers during a continuous 14 week period. According to regulator data, for the observed period, the carrier owned 20% of mobile lines in the country, where mobile phone had already reached market saturation (1.16 mobile lines per person). The data contain records for 2.2 billion interactions among 11 million users.
- User Data (UD): provide age and gender for 8 million users, identified by consistent anonymized hashes.

In order to aggregate this information in a loseless way, records were grouped by relationship (link). For phone calls, along with first and last interaction timestamps, 3 vectors per relation were built:

- Duration vector: contains durations (in seconds) of all phone calls between the two users.
- Inter-event vector: contains inter-event time (in minutes). If the previous vector has length N, inter-event vector has length $N - 1$.
- Direction: binary vector providing caller and receiver roles for the interaction. If the relation is defined as "A-B", this vector has ones when the caller is A and zeroes otherwise.

Once aggregated, 168 million different relationships were found. The next step consisted in filtering meaningful relationships. Many CDR records belong to corporate phone lines for which it does not make sense to talk about user age or gender, since usually there are several persons behind those specific numbers. In this research, the criterion employed to filter out this kind of interactions was the criterion proposed in a seminal paper by Onella et al. [10]: only relationships with at least one call in each direction of the communication were considered. This way, 40 % of originally obtained relations were eliminated from the study.

After having chosen these mutual phone calls relationships, SMSs, inter-event and direction vectors for those relationships were built. The reason for using calls to define meaningful relationships instead of using SMS records, is that the mutual strategy does not work so well for SMS; this is so because of the increasing number of value added services which involve texting in both directions between the user and the corresponding automatic services.

4 Exploratory Analysis and Learning Approach

The resulting network metrics do agree with the metrics proposed in previous works on mobile phone networks [10,6]: high clustering coefficient, short diameter and a long-tail degree distribution. In our data, the degree distribution was found to fit a power-law with exponent $\gamma = -6.27$ which means that there is no scaling behaviour, but the decay does not fit an exponential model either. These features on the degree distribution were also found in previous works, where it was argued that the absence of scaling may be produced by the removal of non-mutual links.

From an exploratory perspective, an interesting behaviour was found in the inter-event time distribution. Basic network engineering does usually assume that the time between two calls (within a big enough population) can be described by a Poisson distribution, and that is the original assumption for many techniques used to properly dimension communication networks [12]. However, recent studies [8,11] have proved that this poissonian behaviour is not present in the individual level: certain events (for example an incoming call) make the user to immediately initiate a communication burst. In our research, once the inter-event vectors for links were compiled, the distribution at this link level was studied.

Inter-call vectors for 10000 randomly chosen links were concatenated, producing 105174 inter-call time samples. Figure 2 shows an histogram of those samples in the time interval from 6 hours to 4 days. One can easily identify that the distribution is peaked every 24 hours. This means that if two users A and B talk to each other by phone today at 10am, it is much more likely for their next conversation to happen the day after tomorrow at 10am than tomorrow at 6pm. This fact contradicts the Poisson distribution monotonic decay, proving that, at link level, there is no poissonian behaviour either.

After the exploratory analysis, a methodology to address the prediction problem was designed, consisting of two separate steps:

– Isolated link prediction: given a relationship A-B, extract features from link available data in order to predict age and gender for B.

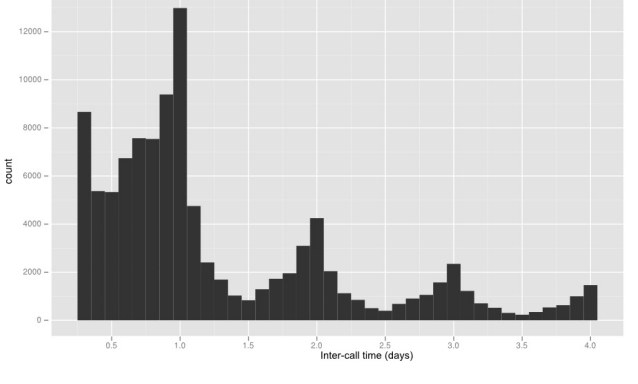


Fig. 2. Inter-call time distribution for a 10000 links sample

– Ego-network: use the results from the previous step to provide a prediction from each link leading to user B. These results, together with ego-network features, are then employed to predict attributes of B.

5 Isolated Link Approach

As previously stated, this section is aimed to perform a most accurate prediction of user attributes (gender, age) using only link related information. Link data gather three main sources of information: SMS records, Call records, and gender/age from the other user in the link.

As it will be shown later, a large part of prediction capability will come from demographic information pertaining to the other user. A reason for this to happen is the well-known phenomenon of homophily in social networks [7], formally defined as the tendency of connected nodes to be correlated. In our stage, this means that if user A and B are connected by a link, it is likely that they both have similar age and even same gender.

Apart from taking advantage of homophily, the manner people communicate also depends on their age and gender, as it was proved by Stoica et al. in [14]. That research found out that it is possible to cluster users into a number of groups according to their communication patterns (unsupervised learning). Then it was shown that some user attributes, such as age, were correlated with the group membership of the user. Although the phenomenon leading to this research is exactly the same, our approach will be grounded on supervised learning (precisely, a classification problem). On the other hand, a larger number of features will be used for machine learning, specially those related to communication dynamics whose relevance has been recently pointed out.

To run this experiment, 9860 links were randomly chosen among those whose both users data (gender and age) were available.

5.1 SMS and Call Metrics

As it has been already mentioned, SMS are one of our three sources of link related data. For each relationship 3 vectors are available which contain all the information associated with direction and communication times. Seminal research on mobile social networks [10] did commonly characterize the link using only quantitative information, such as the number of interactions or total conversation time. Later is was pointed out [8] that, due to the bursty pattern of individual communication, quantitative information may not be enough to describe the nature of the link: 50 messages a week in a relationship may not be more relevant than 10 messages during a month. Using these criteria, new selected metrics from SMS have been calculated:

– Number of SMS during the observation period.
– Mean time between messages and conversation length (from first message to last).

- Variation coefficient (average/standard deviation) for inter-event time.
- Reciprocity: in a link A-B, where B is the opaque node, fraction of messages sent by A. This is the only asymmetric feature for SMS data.
- Fraction of calls during weekend, during work hours, and peak hour of the conversation (0-23).

Concerning call records, the extracted metrics follow similar criteria:

- Number of calls, average call duration, and average inter event time.
- Inter-event variation coefficient and call reciprocity.
- Fraction of calls during weekend, during work hours, and peak hour of the conversation (0-23).

5.2 Gender Prediction

Figures 3 and 4 show the kernel density functions for the metrics described above, aggregated by gender. In general, few gender differences can be found by looking at those graphs; nevertheless, there are a couple of interesting facts to be pointed out. The average call length seems to be higher if user B is female. The median[3] of call duration if B is a female is 89.67 seconds and 75.06 seconds if B is a male. On the other hand, if there is a user sending 20% or less of the messages in the relationship, it is more likely this person to be a man.

Machine Learning Procedure. The problem is defined as a binary classification one. In order to gather an accurate idea of the data quality regarding gender prediction, several learning schemes are tested: Linear Discriminant Analysis (LDA), Decision Trees (Tree), Multilayer Perceptron (Nnet), Bagging and Support Vector Machines (SVM)[4]. In order to improve performance quality (mostly for LDA, which cannot perform non-linear transformations), long tail distributed metrics are logarithmized and, after that, all metrics are standardized (zero mean and unit variance).

Once the data are ready, predictions are obtained from a 10-fold cross validation scheme, ensuring every sample belongs to both training and test sets. Data are provided to the learning machinery in 3 different steps: first only SMS data are provided, then call data are also included, and finally user data are included as well. This whole procedure is summarized in table 1.

The results show an increasing performance, specially when user-data are included. This means that homophily definitely plays a role in this problem. However SMS and call metrics also help to reach a more accurate prediction of user's gender. On the other hand, the capability of splitting non linearly separable sets does not seem to help at all, since LDA performance is almost the same as Nnet or SVM.

[3] Due to the long tail distribution on call duration, it is more robust to use the median to characterize group differences, since the mean is severely biased by outliers.

[4] Implementations used in this article were the following R-packages: MASS (LDA), rpart (Tree), randomForest (Forest), nnet (Multilayer Perceptron), ipred (bagging) and e1071 (SVM).

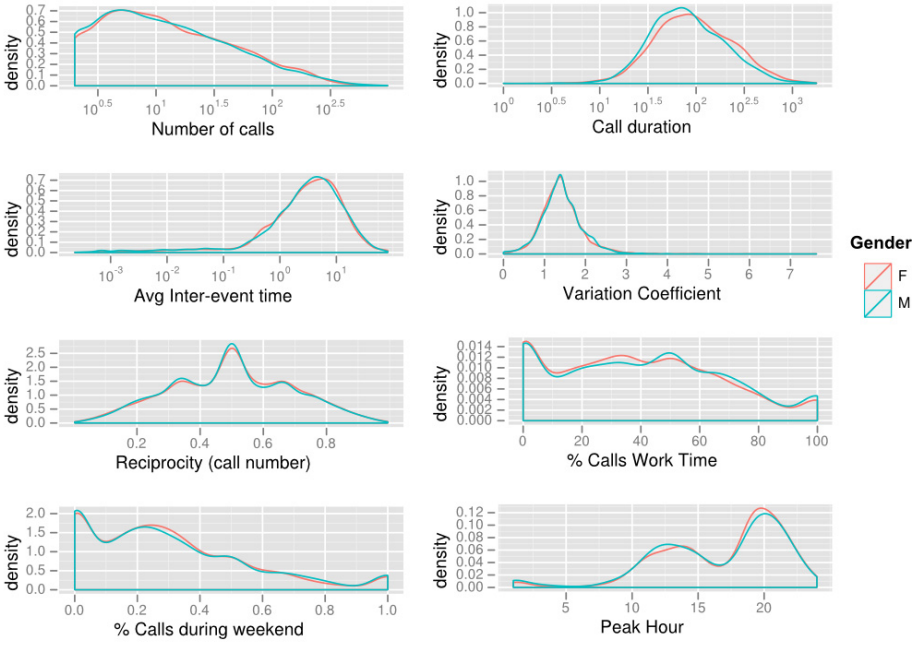


Fig. 3. Density functions for call metrics by gender

Table 1. Gender prediction accuracy using isolated link information

	Random	LDA	Tree	Nnet	Bagging	SVM
SMS	0.5	0.5272	0.5206	0.5334	0.5195	0.5321
SMS +Calls	0.5	0.5403	0.54429	0.5502	0.5343	0.5496
SMS + Calls + User-data	0.5	0.5914	0.60439	0.5945	0.5787	0.6050

5.3 Age Prediction

For prediction purposes, the ages of the users were binned into 6 different age segments. These segments were chosen according to the age distribution, so that every segment has the same number of users in a random sample. This way the age regression problem is transformed in a multi-class classification problem with balanced classes.

After redefining the problem as a classification one, the same methodology discussed in section 5.2 can be applied. Figures 5 and 6 show the density functions for different age groups. An exploratory study shows that people over 30 years old usually call more often during work time. Concerning the amount of SMS in the relationship, it is interesting to note that density functions are sorted, meaning that the elder a person is the fewer texts he/she sends. This behavior was also observed in [14]. However, the statement just made (younger implies more SMS) has an exception in our data which does not show up in any previous

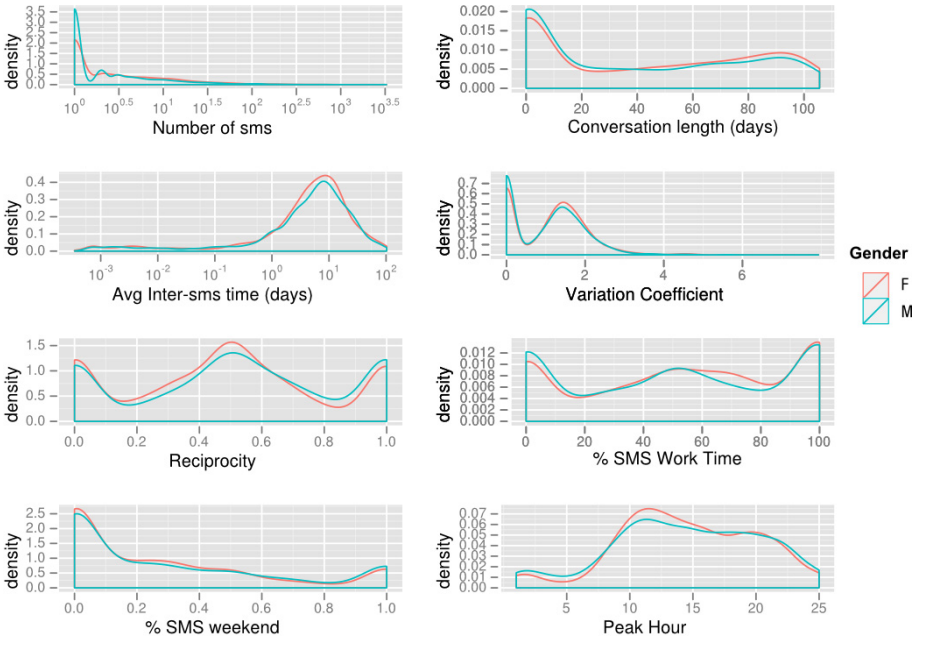


Fig. 4. Density functions for SMS metrics by gender

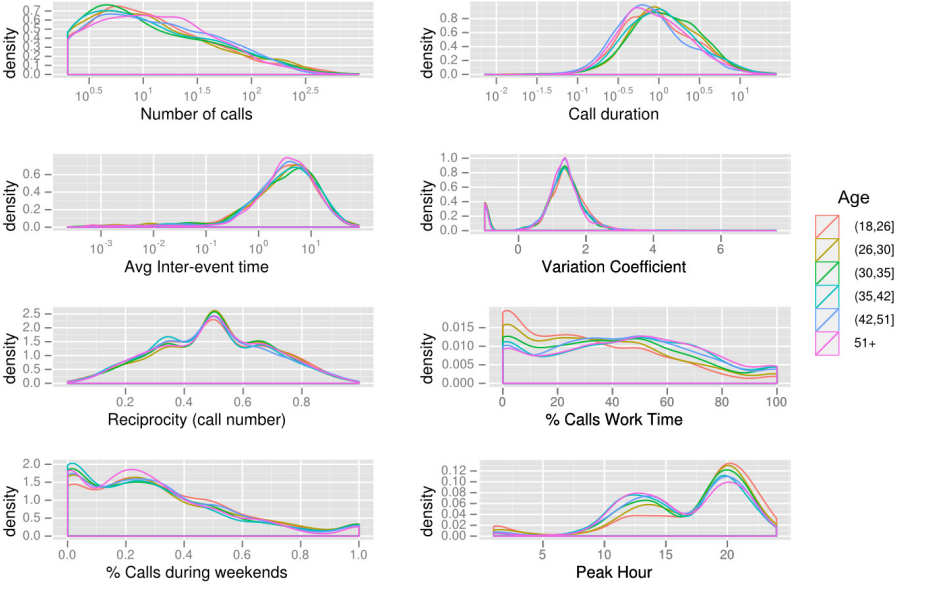


Fig. 5. Density functions for call metrics by age group

Table 2. Age prediction accuracy in using isolated link information

	Random	LDA	Forest[5]	Nnet	Bagging	SVM
Calls	0.1667	0.1997	0.1849	0.2194	0.2020	0.2156
SMS+Calls	0.1667	0.2340	0.2273	0.2410	0.2192	0.2395
SMS + Calls + User-data	0.1667	0.3907	0.4095	0.4027	0.3904	0.4021

study: youngest people (18-26) text a little bit less than people in the next segment. We propose an explanation for that fact: according to recent reports [4], the youngest people (18-25) acquisition of mobile Internet flat rates is higher than in any other age segment. In the same report it is stated that the increase of mobile data plans is severely correlated with the decrease of SMS usage. Hence, we conclude that the observed "lack of messages" among youngest users is probably masked by the replacement of IP-based messaging services whose traces are not recorded by the carrier. The impact of these new technologies on mobile network data should be taken into account in future studies.

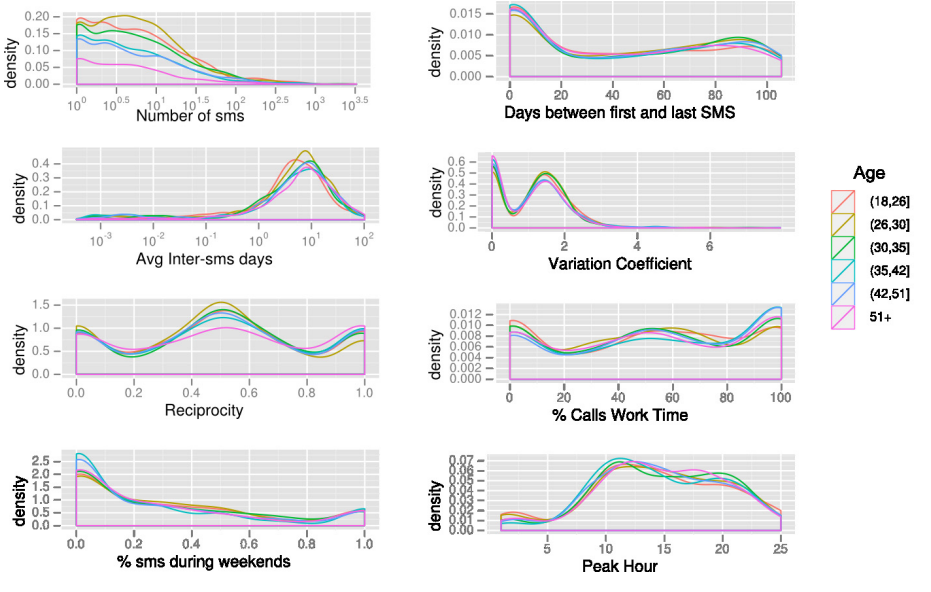


Fig. 6. Density functions for SMS metrics by age group

Table 2 shows the accuracy results for this classification problem. The results show that there is a prediction capability on communication metrics specially if SMS data are included. However, this prediction capability is outperformed if user data (precisely, user age) are included. Age homophily in mobile phone communications is so intense that all five classification techniques mimic the identity function on user age like the best possible classification scheme. The reason for this fact can be observed in figure 7, which represents a scatter density

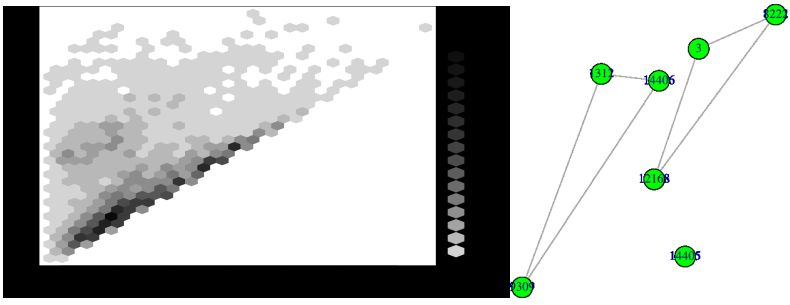


Fig. 7. Left: Age homophily in communication links. Right: ego network: the removal of the opaque node may lead to a not connected graph.

plot for ages in the same link. It is straightforward to check that the probability for a user A to be in the same ages than the user B is extremely high (dark colors in the diagonal of the plot).

6 Ego-Network Approach

The importance of network analysis has been raised over the last decade, where seminal papers by Watts-Strogatz [15], Barabási-Albert [1] and Newman [9] pointed out that many real world systems can be modeled as a network, and the study of the resulting graph usually provides non-obvious and relevant system features. For this reason, this research on multi-link prediction is not only about combining results from single link results, but also including information of the network structure around the opaque node.

6.1 Network Metrics

Due to the size and inherent complexity of their analysis, real-world big networks (mobile phones, social connections online...) are usually analyzed using global information. A very common approach to network analysis is the extraction of a certain N-grade neighborhood around a set of nodes of interest. Among these local neighborhoods, the one which has been more commonly studied is the ego-centered network. This graph includes, for a certain node, all its neighbors and the connections among them. It does not include the center node itself, neither the connections from it to the neighbors, so it is possible a ego-network not being a connected graph, as we can see in figure 7.

Once the subject under study is defined as the ego, a number of features are defined. Apart from quantitative information, such as the number of nodes and edges, some small structures are analyzed. It has been proved that some subgraphs show up much more often than in a purely random network. These subgraphs are called motifs, and their importance in biological networks has been already stressed: the appearance of some kind of motifs is related to some specific

Table 3. Gender prediction accuracy using ego-network data

	Random	LDA	Forest[6]	Nnet	Bagging	SVM
Network	0.5	0.5324	0.5295	0.5362	0.5198	0.5373
Net + Gender Score	0.5	0.6304	0.6323	0.6512	0.6420	0.6532

function within the cell. In social communications networks, the appearance of motifs has also been studied [13,14,16].

According to these guidelines the following metrics were used as network features:

- Node, edge and isolated node count.
- V-motifs (unclosed triangles), closed triangles and 4-star motifs (one node connected to 3).

For this experiment, data from 5670 subscribers were randomly chosen, and their neighborhood information was gathered. This way, a total number of 22098 users and 50377 relationships were analyzed for prediction purposes.

6.2 Gender Prediction

Figure 8 shows that the main difference between gender, regarding network features, is that women seem more likely to have triangles (less stars) in the ego-network. In order to perform final learning for gender classification, results from link level prediction are grouped in a gender score, whose value is the rate of female predictions for the node under study. For example, if there were 3 links, 2 predictions were male and 1 female, the gender score is 0.33. Therefore, a total of seven metrics (six network metrics plus gender score) were analyzed.

Accuracy results are shown in table 3, which shows that the multi-link level increases the performance by about 5% compared to predictions using only one link. On the other hand, figure 9 shows the Receiver Operating Characteristic (ROC) which shows how the accuracy of the best techniques (SVM and Multi-Layer Perceptron) is robust to small variations of the selected threshold.

6.3 Age Prediction

Regarding age, there seems to be a larger diversity in ego-network structure. Figure 10 shows the correlation between age and the appearance of certain motifs, specially stars and triangles. For age prediction, isolated link results were included in the experiment by incorporating 6 variables which contain the number of link level predictions for each label. Prediction accuracy results using these 12 variables (6 age scores and 6 network metrics) are shown in table 4.

Classification results show that the use of network metrics improves link-level predictions by around 10 %, reaching a final performance three times higher than with a random predictor. In addition, prediction errors usually lead to either the

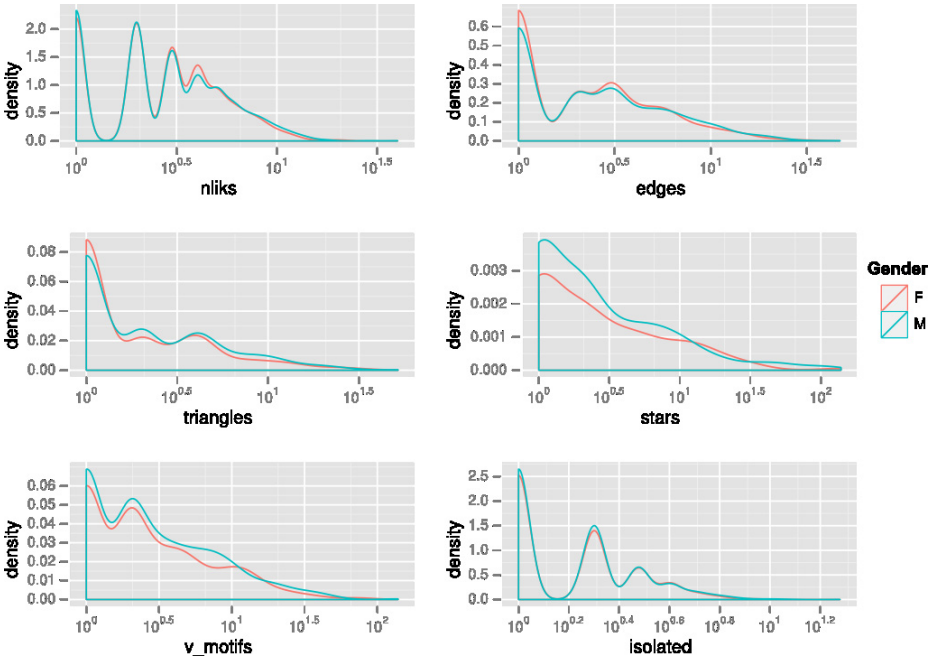


Fig. 8. Density functions for network metrics by gender

Table 4. Age prediction accuracy using ego-network data

	Random	LDA	Forest	Nnet	Bagging	SVM
Network	0.1666	0.2108	0.2022	0.2194	0.2030	0.2092
Net + Age Score	0.1666	0.5050	0.4904	0.5082	0.4931	0.5102

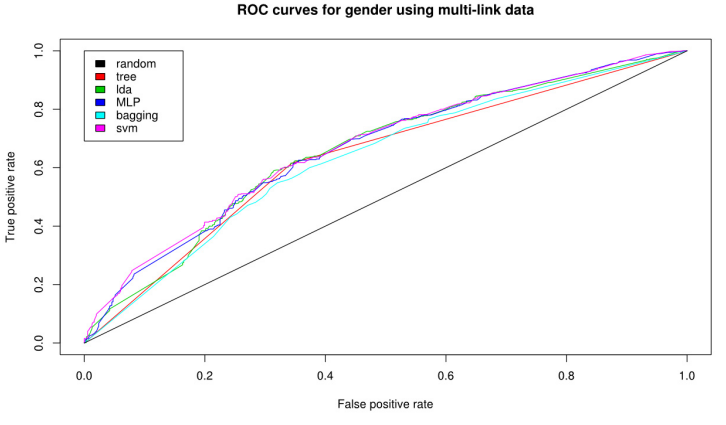


Fig. 9. ROC curve for different machine learning techniques

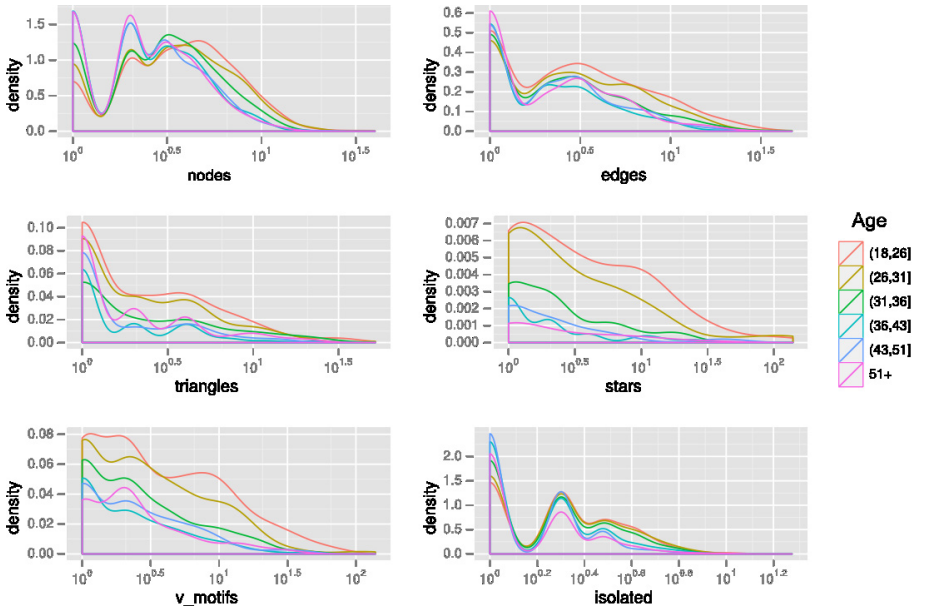


Fig. 10. Density functions for network metrics by age

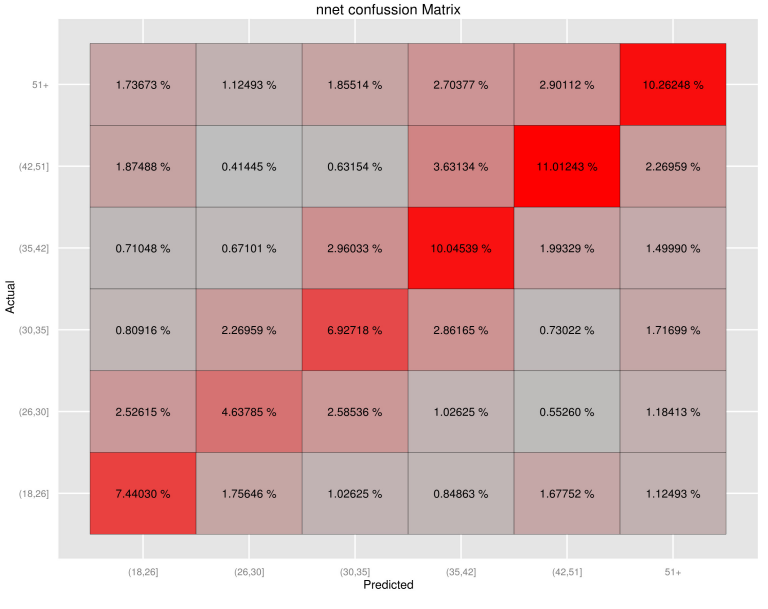


Fig. 11. Confusion matrix for neural network multi-link classifier

following or the previous age element, as it can be seen in the confusion matrix of figure 11. Note that when turning the regression problem into a classification one, the topology on the age variable was neglected; this could have propitiate that errors would lead to age segments non-contiguous with the correct one. Fortunately, the favorable results discarded this main concern.

7 Conclusion and Future Work

The paper has shown that machine learning tools can be a very useful for predicting several attributes corresponding to telephone users, using network data. Two approaches have been evaluated which make use of link information and network structure to improve the attribute prediction capabilities. Future research work on network science may furtherly benefit from machine learning paradigms to generalize the proposed feature extraction schemes when dealing with other different types of networks.

Acknowledgements. The authors want to acknowledge the financial support of Orange (Spain and France), in the framework of Cátedra Orange at the ETSI Telecomunicación in the Universidad Politécnica de Madrid (UPM). The work has been also partially supported by projects MTM2010-15102 of Ministerio de Ciencia e Innovación, and Q10 0930-144 of the UPM, Spain.

References

1. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Reviews of Modern Physics 74(1), 47 (2002)
2. Davidson, E., Levin, M.: Gene regulatory networks. Proceedings of the National Academy of Sciences of the United States of America 102(14), 4935 (2005)
3. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology. ACM SIGCOMM Computer Communication Review 29, 251–262 (1999)
4. Gimeno, M., Villamia, B., Suarez, V.: eEspaña, Informe anual sobre el desarrollo de la sociedad de la información en España. Fundación Orange (2011)
5. Granovetter, M.S.: The strength of weak ties. American Journal of Sociology, 1360–1380 (1973)
6. Hidalgo, C.: The dynamics of a mobile phone network. Physica A: Statistical Mechanics and its Applications 387(12), 3017–3024 (2008)
7. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: Homophily in social networks. Annual Review of Sociology, 415–444 (2001)
8. Miritello, G., Moro, E., Lara, R.: Dynamical strength of social ties in information spreading. Physical Review E 83(4), 3–6 (2011)
9. Newman, M.E.J.: The structure and function of complex networks. SIAM Review 45(2), 58 (2003)
10. Onnela, J.-P., Saramaki, J., Hyvonen, J., Szabo, G., Menezes, A.D., Kaski, K., Barabasi, A.-L., Kertesz, J.: Analysis of a large-scale weighted network of one-to-one human communication. New Journal of Physics 9(6), 25 (2007)

11. Rybski, D., Buldyrev, S.V., Havlin, S., Liljeros, F., Makse, H.: Scaling laws of human interaction activity. Proceedings of the National Academy of Sciences of the United States of America 106(31), 12640–12645 (2009)
12. Schwartz, M.: Telecommunication networks: protocols, modeling and analysis. Addison-Wesley Longman Publishing Co., Inc. (1986)
13. Stoica, A., Couronne, T., Beuscart, J.S.: To be a star is not only metaphoric: from popularity to social linkage. In: Proc. ICWSM 2010 4th. Intl. Conf. Weblogs & Social Media (2010)
14. Stoica, A., Smoreda, Z., Prieurb, C., Guillaumec, J.L.: Age, gender and communication networks. In: Proceedings of the Workshop on the Analysis of Mobile Phone Networks, Satellite Workshop to NetSci. 2010 (2010)
15. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature 393(6684), 440–442 (1998)
16. Zhao, Q., Tian, Y., He, Q., Oliver, N., Jin, R., Lee, W.C.: Communication motifs: a tool to characterize social communications. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 1645–1648. ACM (2010)

A Hybrid Approach to Increase the Performance of Protein Folding Recognition Using Support Vector Machines

Lavneet Singh, Girija Chetty, and Dharmendra Sharma

Faculty of Information Sciences & Engineering,
University of Canberra, Australia
{Lavneet.singh,girija.chetty,dharmendra.sharma}@canberra.edu.au

Abstract. In area of bioinformatics, large amount of data is being harvested with functional and genetic features of proteins. The data is being generated consists of thousands of features with least observations instances. In such case, we need computational tools to analyze and extract useful information from vast amount of raw data which help in predicting the major biological functions of genes and proteins with respect to their structural behavior. Thus, in this study, we use a new hybrid approach for features selection and classifying data using Support Vector Machine (SVM) classifiers with Quadratic Discriminant Analysis (QDA) as generative classifiers to increase more performance and accuracy. We compare our results with previous results and seem to be much promising. The proposed method provides the higher recognition ratio rather than other method used in previous studies. The obtained results are also compared with other different classifiers and our hybrid classifiers give more accuracy and achieve better results than any other classifiers.

Keywords: Support Vector Machines (SVM), Quadratic Discriminant Analysis (QDA), Feature Selection, Protein Folding.

1 Introduction

Protein Folding Recognition is one of the challenging and most important problems in area of bioinformatics. The structure of protein plays an important role in its biological and genetic function [1]. Thus to know about protein and genetic sequences which is basically sequences of various amino acids in protein molecules, firstly it structures is also to be known and how it folded. As with increase in computational power of computers, many genome sequencing research reach near to find its known amino acid sequencing but there are least proteins which 3-D structures has being find out. There are several machine learning method has being introduced in previous studies to predict protein folding structures and amino acids sequencing. Ding and Dubchak [2] uses the support vector machines (SVM) and Artificial Neural Network (ANN) classifiers to extract features and various properties on which certain defined protein folding is being predicted. Shen and Chou [3] proposed model based on nearest

P. Perner (Ed.): MLDM 2012, LNAI 7376, pp. 660–668, 2012.

neighbor algorithm and its modified nearest neighbor algorithm called K-local hyperplane (H-KNN) was implemented by Okun [4]. Nanni [5] also proposed model using Fishers linear classifier and H-KNN classifier. Eddy [6] uses Hidden Markov Models for protein folding recognition. There model predicted the most accurate rate of protein folding. However, the disadvantage in using Hidden Markov Models is that it needs high computational power working on large datasets of protein folding for training and testing but [7] and [8] uses the reduced state space Hidden Markov Model with small architecture.

Basically, in protein folding problem, classification is done on behalf of two types of classifiers: probabilistic approach use the training data to map the probability estimates of each class and find the posterior probability of test data. The second type of classifiers used the likely neighborhood or weights between different classes based on instances of training data. In our study, we use the Support Vector Machines (SVM) with specific parameters with generative Quadratic Discriminant Analysis (QDA) for classification of protein folding problem.

Many researchers have used the fusion of different classifiers to increase performance and accuracy in recognition rate in area of bioinformatics. Shen and Chou [9] proposed a fused classifier for large-scale human protein sub-cellular location prediction and Nanni et al. [10] used SVM classifiers fused with the max rule algorithms. The problem in classification and accuracy rate in protein folding problem is that after converting data into mxn matrices, it contains thousands of features with least training data which makes harder to implement any model in learning phase. Thus, before applying any classifiers, the first step is to reduce the high dimensionality features data so as to with n observations and instances, we have n number of features for learning phase in training data. Reducing features and to over fit noise can be achieved by using statistical or probabilistic approach. There are two ways of features reduction: feature selection and feature transformation to convert high dimensional data into new space with reduced dimensionality.

In this study, we uses the various classifiers as Support Vector Machines (SVM), Multiple Linear Regression, Neural Network (ANN) as Multi Layer Perceptron (MLP), and Random Forest and then finally all results are compared with our proposed hybrid classifiers fused using Quadratic Discriminant Analysis (QDA) and Support Vector Machines (SVM) which give more accurate results and recognition ratio than any other compared classifiers. SVM is a binary classifiers and protein folding recognition is a multi-class problem. Thus in this study, we use the strategy to give weights so as maximum same or near weights of each class is taken using the discriminant function of QDA classifiers. The rest of the paper is organized as follows: Section 2 discuss about the protein database and its features vectors. Section 3 defines the features selection and classification using SVM and fused hybrid classifiers, Section 4 presents the experiment results and Section 5 proposed the discussions based on experimental results including conclusions and future work.

2 Protein Database and Its Features

2.1 Protein Database

In this study, we have used the data sets derived from structural classification of proteins (SCOP) database [11]. The details of these protein sets have being clearly described in [12]. In our experiments, datasets consists of 698 protein sequences define but different features. These data sets included protein from 27 different folds representing all major structural classes: α, β, α/β and α + β. We used the cross validation with k folds for training and testing of data.

2.2 Features Vectors

In our experiments, we have uses the features described by Ding and Dubchak [2]. These feature vectors are based on six parameters: Amino acids composition (C), Predicted secondary structure (S), Hydrophobity (H), Normalized van der Waals volume (V), Polarity (P) and Polarizability (Z). Each parameter corresponds to 21 features except Amino acids composition (C), which corresponds to 20 features. We then compile all features into one data set corresponding to full feature vector (C, S, H, V, P, Z) counts 125 features. All feature vectors are standardize and normalize to the range of [-1; +1] before applying any classifiers. Standardizing and normalizing the features attributes reduces into small numeric ranges which is much easier for classifiers to classify with respective class.

3 Machine Learning Classifiers

3.1 Support Vector Machine Classifiers (SVM)

The support vector machine (SVM) is a well-known famous large margin classifier proposed by Vapnik [13]. The basic concept of the SVM classifier is to find an optimal separating hyper- plane, which separates two classes. The decision function of the binary SVM is

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

where b is a constant, yi ϵ {-1,1}, 0≤αi≤C, I = 1,2,……N are non negative Lagrange multipliers, C is a cost parameter, that controls the trade-off between allowing training errors and forcing rigid margins, xi are the support vectors and K(xi, x) is the kernel function.

After that we follow on SVM as multiclass problem using one against one method. It was first introduced in [14], and the first use of this strategy on SVM was in [15,16]. This method constructs k(k-1)/2 classifiers where each one trains data from two classes. For training data from the ith and the jth classes, we solve the following

binary classification problem. In this study, we use the max win voting strategy suggested in [17]. If sign ((wij) TØ (x) + bij)) says x is in the ith class, then the vote for the ith class is added by one. Otherwise, the jth is increased by one. Then the largest vote will be givrn to specific class on variable x.

We use the software LIBSVM library for experiments. LIBSVM is a general library for support vector classification and regression, which is available at http: //www.csie.ntu.edu.tw/~cjlin/libsvm. As mentioned above, that there are different functions to map data to higher dimensional spaces, practically we need to select the kernel function K(xi; xj) =Ø (xi)TØ(xj). There are several types of kernels in used with all kinds of problems. Each kernel has different parameters for different problems. For example, some well-known problems with large amount of features, such as text classification [18] and DNA problems [19], are reported to be classified more correctly with the linear kernel. In our study, we use the RBF kernel. A learner with the RBF kernel usually performs no worse than others do, in terms of the generalization ability. In this study, we did some simple comparisons and observed that using the RBF kernel the performance is a little better than the linear kernel K(xi; xj) = Ø (xi)TØ(xj) for all the problems we studied. Therefore, for the three data sets instead of staying in the original space a non-linear mapping to a higher dimensional space seems useful. Another important issue is the selection of parameters. For SVM training, few parameters such as the penalty parameter C and the kernel parameter of the RBF function must be determined in advance. Choosing optimal parameters for support vector machines is an important step in SVM design. We use the cross validation on different parameters for the model selection.

3.2 Quadratic Discriminant Analysis

Quadratic discriminant analysis (QDA) [20] describe the likelihood of a class as a Gaussian distribution and then uses the posterior distributions estimates to estimate the class for a given test vector. This approach leads to the function:

$$d_k(x) = (x - \mu_k)^T \sum_k^{-1} (x - \mu_k) + log \sum k - 2 \log p(k) \quad (2)$$

Where ∑k is the covariance matrix, x is the test vector, μk is the mean vector, and p(k) is the prior probability of the class k. The Gaussian parameters for each class can be estimated from the training dataset, so the values of ∑k and μk are replaced in above formula by its estimates ^∑k and ^ μk. However, when the number of training samples is small, compared to the number of dimensions of the training vector, the covariance estimation can be ill-posed. The approach to resolve the ill-posed estimation is to regularize the covariance matrix ∑k. It can be replaced by the average matrix i.e.

To apply QDA, our first goal is to reduce the dimension of the data by finding a small set of important features which can give good classification performance. Feature selection algorithms can be roughly grouped into two categories: filter methods and wrapper methods. Filter methods rely on general characteristics of the data to evaluate and to select the feature subsets without involving the chosen learning

algorithm (QDA in our case). Filters are usually used as a pre-processing step since they are simple and fast. A widely-used filter method for bioinformatics data is to apply a univariate criterion separately on each feature, assuming that there is no interaction between features. We apply the t-test on each feature and compare p-value (or the absolute values of t-statistics) for each feature as a measure of how effective it is at separating groups.

3.3 Feature Selection

As the dataset in our protein folding recognition is ill posed. We have greater number of features and sample is small which may be insufficient to solve the problem. Thus, in this case, we have to use feature selection algorithm to reduce the dimensionality of feature space. There are many feature selection algorithms used in the past literature review, but we try to approach efficient and faster feature selection algorithm to reduce the dimensionality space. Thus, in this study we use three different methods for feature selection on protein dataset which is clearly described in section 3.3.1.

3.4 Generalized Linear Model

In this model, we have used the generalized linear model for feature selection based on the probability value as (p-value) using t test statistics. We have used as a pre-processing step since they are simple and fast. We apply a univariate criterion separately on each feature, assuming that there is no interaction between features. We apply the t-test on each feature and compare p-value (or the absolute values of t-statistics) for each feature as a measure of how effective it is at separating groups. In order to get a general idea of how well-separated the two groups are by each feature, we plot the empirical cumulative distribution function (CDF) of the p-values:

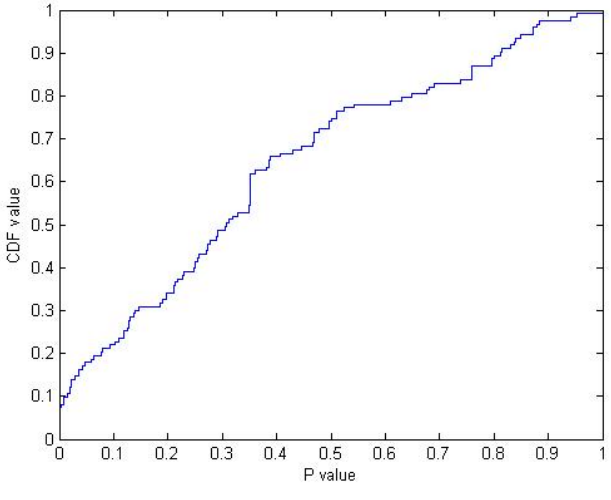


Fig. 1. CDF value with respect to p value of protein dataset

There are about 35% of features having p-values close to zero and over 50% of features having p-values smaller than 0.05, meaning there are more than 60 features among the original 125 features that have strong discrimination power. One can sort these features according to their p-values (or the absolute values of the t-statistic) and select some features from the sorted list. However, it is usually difficult to decide how many features are needed unless one has some domain knowledge or the maximum number of features that can be considered has been dictated in advance based on outside constraints. One quick way to decide the number of needed features is to plot the MCE (misclassification error, i.e., the number of misclassified observations divided by the number of observations) on the test set as a function of the number of features. The resubstitution MCE is over-optimistic. It consistently decreases when more features are used and drops to zero when more than 69 features are used. However, if the test error increases while the substitution error still decreases, then overfitting may have occurred. This simple filter feature selection method gets the smallest MCE on the test set when 28 features are used. The plot shows overfitting begins to occur when 28 or more features are used. The smallest MCE on the test set is 12.5%.

3.5 Sequential Feature Selection Algorithm

We have also extracted the features in our dataset usinf sequential feature selection algorithm. Sequential feature selection is one of the most widely used techniques. It selects a subset of features by sequentially adding (forward search) or removing (backward search) until certain stopping conditions are satisfied. In this study, we use forward sequential feature selection to find important features. More specifically, since the typical goal of classification is to minimize the MCE. The training set is used to select the features and to fit the QDA model, and the test set is used to evaluate the performance of the finally selected feature. During the feature selection procedure, to evaluate and to compare the performance of the each candidate feature subset, we apply stratified 7-fold cross-validation to the training set.

Experiments

Our experiments include implementation of SVM and other classifiers on the prescribed protein dataset and to do a comparative study to find the higher accuracy rate. To set on these experiments, firstly we choose certain parameters. We use the cross validation technique to avoid over fitting. We use k-fold cross validation with k=7, because there must be at least 7 samples of each class in the training dataset. Then the second parameter is to decide the kernel for SVM classifier. The RBF kernel is

$$K(x_i, x) = -\gamma(x - x_i)^2 \; \gamma > 0 \tag{3}$$

W uses other kernels also as linear, polynomial and Gaussian .The RBF kernel gave the best results in our experiments. The parameters C from Eq.(3) and g have certain parametric value. Both values has been experimentally chosen, which was done using a cross-validation procedure on the training dataset. The best recognition ratio was achieved using parameter values g = 0.7 and C = 300.

3.6 Feature Selection Algorithms

The best selection method for feature selection is to check all combinations. However, the number is too high so can't go ahead with combinations only. So, we combine all features based on parameters C, S, H, V, P, Z to make an subset. Then the various feature selection algorithms as previously described is being implemented to reduce the dimensionality space. Or first approach is to use simple Wrapper feature selection which reduces the features to 28 features. Our second approach is to use sequential forward selection algorithm which gives a combination 69 features matrix with observed instances. The combination of 69 features gives the best recognition ratio results using cross validation procedure. We also use the probabilistic feature selection algorithm which uses the generalized linear model as t- test statistics to find the significant features. Using this algorithm, we sort out the 48 features and give an average results using cross validation procedure.

4 Results

In this study we have use SVM and the proposed hybrid classifier for increasing the performance and accuracy. In this study, accuracy is measured in terms of percentage classified recognized ratio. Suppose there is N = n1 + n2 + n3…….+np test proteins, where ni is the number of proteins which belongs to the class i. suppose that ci of proteins from ni are correctly recognized (as belonging to class i). So the total number of C = c1 + c2 +c3…….+cp proteins is correctly recognized. Therefore the total accuracy is Q = C/N.

The SVM classifiers follow on a feature vector to each class with the minimum value of discriminant function. So for every instance, we calculate the function value of each class. In this way we create mxn matrices of two different feature vectors. The SVM classifiers were used with 126D feature vector. All binary classifiers are being trained and each instance is being assigned to each class. The recognition ratio using SVM classifiers is 63.89%.

Table 1. Comparison among different methods

Method	Recognition Ratio (%)
SVM	63.75
H-KNN	57.4
Random Forest	53.72
MLP	54.72
SVM-QDA (proposed method)	65.17

Table 2. Recognition Ratio obtained using various features selection algorithms

Selection Method (Number of Features)	Recognition Ratio	
	SVM (%)	SVM-QDA (%)
Generalized Linear Model (28)	55.36	62.04
SFS (69)	60.19	66.52

The final step was to use three feature selection algorithms matrices. The results are being described in table 1. There were three different voted table and weights for three different matrices with different selected features vectors. The recognition ratio was about 65.26% which is slightly higher than using SVM classifiers alone. Table 2 describes the comparative study of using various classifiers including SVM and using our proposed hybrid SVM classifiers. The result shows that our proposed hybrid classifiers show more performance and accuracy in terms of recognition ratio of proteins instances of various classes. The result seems to be very promising and can be increased by increasing more training data and number of folds with increasing in k-cross validation.

5 Conclusions and Future Work

In this study, we described and present various classifiers with their comparative approach and proposed an improved and more accurate hybrid classifiers based on Support Vector Machines (SVM and Quadratic Discriminant Analysis (QDA). The proposed classifier used the reduced data by using specific feature selection algorithms to improve the recognition and classification rate. The combined SVM-QDA classifiers achieve much better results (65.17%) rather than SVM (63.75%) and any other above mentioned classifiers. The results seem too much improve and accurate which is promising for classification for protein sequences with respect to past research work.

As in this specific protein dataset, we have larger features with fewer samples giving an high dimensionality space in which classification work to much harder. In such case, we use Generalized linear Models and Sequential Forward Selection Algorithm as feature selection which reduce the high dimensional space into less dimension new space which increases the more accuracy and recognition rate compared with alone classifiers implemented on whole high dimensionality data. So, in these experiments, it shows that feature selection algorithms influence the final results of classifiers which seems too much improvement and can be implemented on other high dimensional protein databases for future investigation. Our all experiments are done on the feature vector developed by Ding and Dubchak[2], further work can be done on implementing the same strategy on other high features protein sequences to reduce the computational power, more accuracy and recognition rate which will be much promising than previous results. Further work can be done using SVM with binary decision trees and other algorithms to make an hybrid classifiers.

References

1. Chan, H.S., Dill, K.: The protein folding problem. Physics Today February 24-32 (1993)
2. Ding, C.H., Dubchak, I.: Multi-class protein folds recognition using support vector machines and neural networks. Bioinformatics 17, 349–358 (2001)
3. Shen, H.B., Chou, K.C.: Ensemble classifiers for protein fold pattern recognition. Bioinformatics 22, 1717–1722 (2006)

4. Okun, O.: Protein fold recognition with k-local hyperplane distance nearest neighbor algorithm. In: Proceedings of the Second European Workshop on Data Mining and Text Mining in Bioinformatics, Pisa, Italy, pp. 51–57 (2004)
5. Nanni, L.: A novel ensemble of classifiers for protein folds recognition. Neuro Computing 69, 2434–2437 (2006)
6. Eddy, S.R.: Hidden Markov models. Current Opinion in Structural Biology 6, 361–365 (1995)
7. Madera, M., Gough, J.: A comparison of profile hidden Markov model procedures for remote homology detection. Nucleic Acids Research 30(19), 4321–4328 (2002)
8. Lampros, C., Papaloukas, C., Exarchos, T.P., Golectsis, Y., IFotiadis, D.: Sequence-based protein structure prediction using a reduced state-space hidden Markov model. Computers in Biology and Medicine 37, 1211–1224 (2007)
9. Lampros, C., Papaloukas, C., Exarchos, K., IFotiadis, D.: Improving the protein fold recognition accuracy of a reduced state-space hidden Markov model. Computers in Biology and Medicine 39, 907–914 (2009)
10. Shen, H.B., Chou, K.C.: Hum-mPLoc: an ensemble classifier for large-scale human protein subcellular location prediction by incorporating samples with multiple sites. Biochemical and Biophysical Research Communications 355, 1006–1011 (2007)
11. Nanni, L., Lumini, A.: MppS: an ensemble of support vector machine based on multiple physicochemical properties of amino acids. Neuro-computing 69, 1688–1690 (2006)
12. Zhang, C.X., Zhang, J.S.: RotBoost: a technique for combining rotation forest and adaboost. Pattern Recognition Letters 29, 1524–1536 (2008)
13. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
14. Knerr, S., Personnaz, L., Dreyfus, G.: Single-layer learning revisited: a step-wise procedure for building and training a neural network. In: Fogelman, J. (ed.) Neuro-computing: Algorithms, Architectures and Applications. Springer (1990)
15. Friedman, J.: Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University (1996)
16. Krebel, U.: Pair-wise classification and support vector machines. In: Scholkopf, B., Burges, C.J.C., Smola, A.J. (eds.) Advances in Kernel Methods —Support Vector Learning, pp. 255–268. MIT Press, Cambridge (1999)
17. Lin, C.-J.: Formulations of support vector machines: a note from an optimization point of view. Neural Computation 13(2), 307–317 (2001)
18. Joachims, T.: The Maximum-Margin Approach to Learning Text Classifiers: Methods, Theory, and Algorithms. PhD thesis, Universitaet Dortmund (200)
19. Yeang, C.-H., Ramaswamy, S., Tamayo, P., Mukherjee, S., Rifkin, R.M., Angelo, M., Reich, M., Lander, E., Mesirov, J., Golub, T.: Molecular classification of multiple tumor types. Bioinformatics: Discovery Note 1(1), 1–7 (2001)
20. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press, New York (1990)

Author Index