Proceedings of the Second Workshop on Feature Selection for Data Mining:

Interfacing Machine Learning and Statistics

in conjunction with the

2006 SIAM International Conference on Data Mining

April 22, 2006

Bethesda, MA

Chairs **Huan Liu (Arizona State University)**
Robert Stine (University of Pennsylvania)
Leonardo Auslender (SAS Institute)

Sponsored By 

Workshop on Feature Selection for Data Mining:

Interfacing Machine Learning and Statistics

http://enpub.eas.asu.edu/workshop/2006/

April 22, 2006

Workshop Chairs: Huan Liu (Arizona State University)
Robert Stine (University of Pennsylvania)
Leonardo Auslender (SAS Institute)

Program Committee:

Constantin Aliferis, Vanderbilt-Ingram Cancer Center
Leonardo Auslender, SAS Institute
Stephen Bay, Stanford University
Hamparsum Bozdogan, University of Tennessee
Yidong Chen, National Center for Human Genome Research
Manoranjan Dash, Nanyang Technological University
Ed Dougherty, Texas A&M University
Jennifer Dy, Northeastern University
George Forman, HP Labs
Edward George, University of Pennsylvania
Mark Hall, University of Waikato
William H. Hsu, Kansas State University
Moon Yul Huh, Sungkyunkwan University
Igor Kononenko, University of Ljubljana
Huan Liu, Arizona State University
Stan Matwin, University of Ottawa
Kudo Mineichi, Hokkaido University
Hiroshi Motoda, Osaka University
Sankar K. Pal, Indian Statistical Institute
Robert Stine, University of Pennsylvania
Kari Torkkola, Arizona State University
Ioannis Tsamardinos, Vanderbilt University
Eugen Tuv, Intel
Bin Yu, University of California Berkeley
Lei Yu, Binghamton University
Jacob Zahavi, Tel Aviv University
Jianping Zhang, AOL

Proceedings Chair: Lei Yu (Binghamton University)

Message from the FSDM Workshop Chairs

Knowledge discovery and data mining (KDD) is a multidisciplinary effort to extract nuggets of information from data. Massive data sets have become common in many applications and pose novel challenges for KDD. Along with changes in size, the context of these data runs from the loose structure of text and images to designs of microarray experiments. Research in computer science, engineering, and statistics confront similar issues in feature selection, and we see a pressing need for and benefits in the interdisciplinary exchange and discussion of ideas. We anticipate that our collaborations will shed light on research directions and provide the stimulus for creative breakthroughs.

Huge data sets have grown increasingly large in terms of number of dimensions and number of instances. The models derived from these data sets are mostly empirical in nature. Reducing number of dimensions by selecting variables and features has proven to be efficient and effective in dealing with high-dimensional data. Variable and feature selection is one of those areas in data mining both computer scientists and statisticians have strong interest in and have done extensive research work on. Thus, it is beneficial to computer scientists and statisticians that a bridge of communication be established and maintained among researchers for the purpose of learning from one another in addressing challenges from massive data using variable and feature selection.

This workshop brings together researchers from different disciplines and encourages collaborative research in feature selection. Feature selection is an essential step in successful data mining applications. Feature selection has practical significance in many areas such as statistics, pattern recognition, machine learning, and data mining. The objectives of feature selection include: building simpler and more comprehensible models, improving data mining performance, and helping to prepare, clean, and understand data.

This collection contains a wide range of research work in feature selection. The research considers approaches that reduce the task by eliminating redundant features before optimizing the fit of a model (filter methods and dimension reduction through principal components) as well as the success of optimization strategies (such as genetic algorithms) that accommodate a large number of features and multimodal objective functions. Important application domains include text clustering, analysis of genetic sequences and biomarkers, and the detection of cyber attacks and other types of anomalies.

It has been an enjoyable process for us to work together in achieving the aims of this workshop. We would like to convey our immense gratitude to our PC members and authors who have contributed tremendously to make this workshop a success.

Huan Liu, Robert Stine, and Leonardo Auslender

April 22, 2006, Bethesda, Maryland

Table of Contents

Table of Contents

Knowledge, Data, and Search in Computational Discovery

Pat Langley

Institute for the Study of Learning and Expertise
Stanford University

Early research on machine learning, which had strong links to symbolic artificial intelligence, studied interactions among three factors: knowledge, data, and search. Over the past decade, machine learning and statistics have joined forces to develop powerful techniques that combine data and search but that disregard the role of knowledge. In this talk, I argue that computational learning and discovery systems would benefit from a return to explicit, symbolic representations of knowledge in both their inputs and their outputs. I illustrate this approach with some recent results on the construction and revision of scientific models that are cast as sets of explanatory processes. In closing, I outline some open research problems in machine learning and discovery that revolve around the reintegration of knowledge with data and search.

This talk describes joint work with Nima Asgharbeygi, Will Bridewell, Andrew Pohorille, Oren Shiran, Jeff Shrager, and Ljupco Todorovski.

Feature Extraction for Classification: an LVQ-based Approach

Claudia Diamantini, Domenico Potena
Dipartimento di Ingegneria Informatica, Gestionale e dell'Automazione,
Università Politecnica delle Marche - via Brecce Bianche, 60131 Ancona, Italy
{diamantini,potena}@diiga.univpm.it

Abstract

Feature extraction is the process of selecting a set of relevant features to effectively apply data mining techniques. For the classification task, the relevance of features should be measured on the basis of their discriminative power, as defined inside the Bayes decision theory. This paper presents a truly Bayesian approach to feature extraction for classification, based on Labeled Vector Quantizers (LVQ). The approach is said to be Bayesian, since the LVQ is designed according to an algorithm for the minimization of the average misclassification risk, that allows to get a locally optimal approximation of the Bayes decision border, from which the most relevant features should be derived. We show experimentally the advantages of the proposed method over competing methods.

Keywords: Decision Border, Feature Extraction, Labeled Vector Quantizer, Classification, BVQ.

I. Introduction

Different data mining tasks require different criteria to select the relevant features. For instance, in classification tasks, the relevance of features should be measured on the basis of their discriminative power, as defined inside the Bayes decision theory. In the literature, the approaches to the definition of the set of most relevant features can be classified in *feature selection* and *feature extraction* approaches. Feature selection is the process to find a subset of the original features, hence it is a simple space projection [14]. Feature extraction looks for (linear or non linear) mappings of the original features into more effective features, performing a space transformation before the projection.

In this paper, a truly Bayesian approach to feature extraction for classification is proposed. The approach chooses the linear transformation of the original space that minimize the loss in classification accuracy achieved by the original features, by exploiting the classification rule of an appropriately trained neural network. The approach is similar to the *Decision Boundary Feature Extraction* (DBFE) method of Lee and Landgrebe [13], differing in the neural architecture adopted and in the training algorithm. In particular, we adopt Labeled Vector Quantizer (LVQ) architectures, originally proposed by Kohonen [10], trained with the Bayes risk weighted Vector Quantization (BVQ) learning algorithm. This algorithm is, at the best of our knowledge, the only learning algorithm based on the minimization of the misclassification risk [4]. Under this truly classification-based algorithm, an LVQ moves towards a locally optimal linear approximation of the optimal classification rule. As a consequence, the vectors normal to the pieces of hyperplanes defining the decision border in the LVQ rule are a good approximation of true optimal (i.e. discriminative) features. The approach gives the advantage of a more robust, efficient and effective feature extraction method for the classification task, that scales well with respect to the size of the data set.

The rest of the paper is organized as follows: the remaining part of this section presents an overview of feature extraction methods in general and of the DBFE approach in particular. Then, in section II we describe the BVQ algorithm for the minimization of the error probability performed by a Vector Quantizer. The details of the BVQ-based Feature Extraction (BVQFE) method are given in subsection

II-B. In section III BVQFE is experimentally compared to the other feature extraction methods. Section IV ends the paper.

A. Overview of Feature Extraction Techniques

Data in the original space $\mathcal{R}^n$ can be represented by $\mathbf{x} = \sum_{i=1}^{n} x_i \cdot e_i$, where e_i are column vectors and $\{e_1 e_2 \dots e_n\}$ is a basis for $\mathcal{R}^n$. Feature extraction looks for (linear or non linear) mappings of the original features into more effective features, performing a space transformation. If the mapping is linear, then the feature extraction problem can be viewed as finding n' orthonormal vectors $\phi_1, \dots, \phi'_n$, $n' < n$, in order to transform the feature space $\mathcal{R}^n$ in the reduced space $\mathcal{R}^{n'}$: $\mathbf{x}' = \sum_{i=1}^{n'} x_i \cdot \phi_i$.

The problem is to find the vectors of the linear transformation that maximize or minimize a given criterion. Different criteria have been proposed. They can be roughly divided in data representation and data discrimination criteria. In the former case, the goal is to find the set of reduced features which best approximate the original data so the criteria are based on the minimization of a mean-squared error or distortion measure. One of the better known methods based on this criterion is the *Principal Component Analysis* (PCA) or Karhunen-Loeve expansion [7], that calculates eigenvalues and eigenvectors of the data covariance matrix, and defines as a transformation basis the set of eigenvectors corresponding to the highest eigenvalues. The squared error of the transformation is simply the sum of the leftover eigenvalues. PCA is an optimum method for data compression and signal representation however it presents several limitations for discriminating between data belonging to different classes. In particular, for data discrimination, criteria to evaluate the effectiveness of features should be a measure of the overlap or class separability among distributions, not a measure of fit such as the mean squared error. For this task, Bayes error is the best criterion to evaluate a feature set. Unfortunately, Bayes error is unknown in general. A family of methods that is frequently used in practice, but that is only indirectly related to Bayes error, is called *Discriminant Analysis* (DA), based on a family of functions of scatter matrices. In the simplest form, Linear DA (LDA), also known as Canonical Analysis (CA), considers a within-class scatter matrix for each class, measuring the scatter of samples around the respective class mean, the between-class scatter matrix, measuring the scatter of class means around the mixture mean, and finds a transformation that maximizes the between-class scatter and minimizes the within-class scatter, so that the class separability is maximized in the reduced dimensional space [7], [1]. Other approaches use upper bounds of Bayes error, like the Bhattacharyya distance [2]. Lee and his co-authors [12], [9] proposed a feature extraction algorithm based on the geometry of the Bayes decision decision border (DBFE) in order to predict the minimum number of features needed to achieve the same classification accuracy as in the original space. At the same time the algorithm finds the needed n' feature vectors. The DBFE algorithm is based on eliminating redundant features for classification, that is to find vectors $\beta_1, \beta_2, \dots, \beta_R$ such that make no contribution in discriminating classes. Geometrically, this means that moving along the direction of vector β_k, the classification result of each observation will remain unchanged. It is noted that the direction of a redundant feature and the decision border are parallel. In the same way, a normal vector to the decision border at a point represents an informative direction and its effectiveness is proportional to the area of decision border that has the same normal vector.

In [12], starting from the normal vectors to the decision border, authors define the Effective Decision Boundary Feature Matrix (EDBFM) as

$$\Sigma_{EDBFM} = \frac{1}{\int_{S'} p(x)dx} \int_{S'} N(x) N^t(x) p(x) dx$$

where $N(x)$ is the normal vector at a point x, S' is the portion of decision border containing most of the training data (the effective decision boundary). It is proved that:

- the rank of the EDBFM represents the minimum number of feature vectors needed to achieve the same bayes error probability as in the original space.
- the eigenvectors of EDBFM corresponding to nonzero eigenvalues are the necessary feature vectors.

The DBFE algorithm uses the knowledge of the true cumulative probability density function (cpdf) to define the Bayes decision border. However, true cpdf are generally unknown in real classification problems. In order to overcome this limitation, [13] and [8] proposed the use of Multi-Layer Perceptron (MLP) to estimate the decision border. These approaches have two main limitations: (1) their computational complexity depends quadratically on the training set size, and (2) the training of MLP is based on a squared error criterion, that is not directly related to classification error. In practice, MLP estimates class a posteriori probability distributions, and it is well known that, although a perfect estimation of these distributions leads to the minimum classification error, an accurate estimation does not necessarily lead to good classification performance (see e.g. [6]). In this paper, a truly Bayesian approach to decision border feature extraction is proposed. It exploits Labeled Vector Quantizer architectures, originally proposed by Kohonen [10], trained with the Bayes risk weighted Vector Quantization (BVQ) learning algorithm. This algorithm is, at the best of our knowledge, the only learning algorithm based on the minimization of the Bayes Risk criterion, which transforms to the error probability in the special case of 0/1 loss [4]. An LVQ is shown to define a classification rule in the original feature space that, under this truly classification-based algorithm, moves towards a locally optimal linear approximation of the Bayes classification rule. As a consequence, the vectors normal to the pieces of hyperplanes defining the decision border in the LVQ rule are a good approximation of true optimal (i.e. discriminative) features. The approach gives the advantage of a more robust, efficient and effective feature extraction method for the classification task, that scales well with respect to the size of the data set.

II. Feature Extraction Based on Vector Quantizers

The goal of this section is to present the novel approach to decision border feature extraction. The approach makes use of geometrical properties of Labeled Vector Quantizers (LVQ) to define a cheap representation of decision borders, and of the Bayes Risk weighted Vector Quantization algorithm (BVQ) to obtain an accurate approximation of the true decision border. For this reason, we call the method *BVQ-based Feature Extraction* (BVQFE).

A. The Bayes Risk Weighted Vector Quantization Algorithm

LVQ has been defined by Kohonen and his coworkers [11], [10] as a family of Learning algorithms to adapt a labeled VQ towards the optimal decision rule[1]. The original training algorithms have been inspired by the aim to approximate Bayes decision borders, thus overcoming the limit of training algorithms based on probability density function approximation, hence on the minimum squared distortion principle. However, these algorithms don't have a strong mathematical foundation, and have demonstrated some problems. The original aim of Kohonen has been accomplished by the development of the so-called Bayes Risk weighted Vector Quantization algorithm (BVQ), a gradient-based algorithm for the minimization of the Bayes Risk [4]. This algorithm is the unique algorithm for the optimization of the basic figure of merit in classification tasks, error probability or more generally classification risk, and has demonstrated the capability to accurately approximate the Bayes decision border with a robust, cheap and scalable procedure [3]. Here we briefly describe a simple version of the algorithm for the minimization of error probability.

[1]In his work, Kohonen used LVQ as an acronyms of Learning Vector Quantizers instead of Labeled Vector Quantizer. We believed that the labeling more than the learning has to be enlighten.

Let $\mathcal{TS} = \{(t_1, u_1), \ldots, (t_T, u_T)\}$ be a set of T labeled samples drawn from the original population (the training set), where $t_i \in \mathcal{R}^n$ denotes the feature vector and $u_i \in \mathcal{C}$ is the class the sample belongs to. Furthermore, let $\mathcal{LM} = \{(m_1, l_1), \ldots, (m_M, l_M)\}$ be an LVQ with euclidean distance, where $m_i \in \mathcal{R}^n$ is called code vector and $l_i \in \mathcal{C}$ is the associated class label. The LVQ can be used to define a nearest neighbor classification rule, where the class of a sample x is defined to be the label l^* associated to the code vector m^* at minimum euclidean distance from x. Hence, the hyperplane equidistant from m_i and m_j, denoted by $\mathcal{S}_{i,j} = \{x \in \mathcal{R}^n | \parallel x - m_i \parallel^2 = \parallel x - m_j \parallel^2\}$, represents a piece of the decision border when m_i and m_j have different labels. The BVQ algorithm is an interactive punishing-rewarding adaptation schema. At each iteration, the algorithm considers a training sample randomly picked from $\mathcal{TS}$. If the training sample turns out to fall "on" the decision border, then the position of the two code vectors determining the border is updated, moving the code vector with the same label of the sample towards the sample itself and moving away that with a different label. Since the decision border is a null measure subspace of the feature space, we have zero probability to get samples falling exactly on it. Thus, an approximation of the decision border is made, considering those samples falling close to it (at a maximum distance of $\Delta/2$). A more detailed description of BVQ algorithm is given in the following:

BVQ Algorithm - k-th iteration

1. randomly pick a training pair $(t^{(k)}, u^{(k)})$ from $\mathcal{TS}$;
2. find the code vectors $m_i^{(k)}$ and $m_j^{(k)}$ nearest to $t^{(k)}$;
3. $m_t^{(k+1)} = m_t^{(k)} \quad for \quad t \neq i, j$;
4. if $t^{(k)}$ fall at a distance $d \leq \Delta/2$ from the border $\mathcal{S}_{i,j}^{(k)}$, then

$$m_i^{(k+1)} = m_i^{(k)} - \gamma^{(k)} \frac{\delta(u^{(k)}, l_i^{(k)})}{\parallel m_i - m_j \parallel} (m_i^{(k)} - t_{i,j}^{(k)})$$

$$m_j^{(k+1)} = m_j^{(k)} + \gamma^{(k)} \frac{\delta(u^{(k)}, l_j^{(k)})}{\parallel m_i - m_j \parallel} (m_j^{(k)} - t_{i,j}^{(k)})$$

else $m_t^{(k+1)} = m_t^{(k)} \quad for \quad t = i, j$.

where $\delta(a, b)$ is the delta function: $\delta(a, b) =$ if $a = b$ then 1 else 0.

It is clear that the value of the parameter Δ should be kept as small as possible, in order to have a good approximation, however small values of Δ reduces the number of samples that are actually exploited in the training procedure. Then the optimal value of Δ depends on the number of training samples, and should be chosen as a tradeoff between the speed of the learning process and the accuracy of the final result, in a similar way as for the learning rate parameter $\gamma^{(0)}$. Converging properties of the algorithm suggest to adopt a decreasing schedule for $\gamma^{(k)}$, according to the Robbins-Monro method, while the number of iterations can be set as large as needed, since the algorithm does not suffer from overfitting problems.

B. Feature Extraction by Analytical Border Definition

Having a trained LVQ, the extraction of the most discriminating features from it is straightforward. As a matter of fact, the pairs of code vectors defining the decision hyperplanes can be exploited to calculate the normal vectors to the hyperplanes, and these normal vectors are combined together to form the EDBFM as in the Landgrebe and Lee approach. In order to enhance the result, the normal vectors should be appropriately weighted to take into account the extent of the portion of hyperplane actually forming the decision border. To better explain this point, let us consider the example of decision border shown in Figure 1. For this example, we get four normal vectors to the piecewise linear decision border: [1 0] and [0 1], each repeated two times. Eigenvalues and eigenvectors of the EDBFM matrix turn out

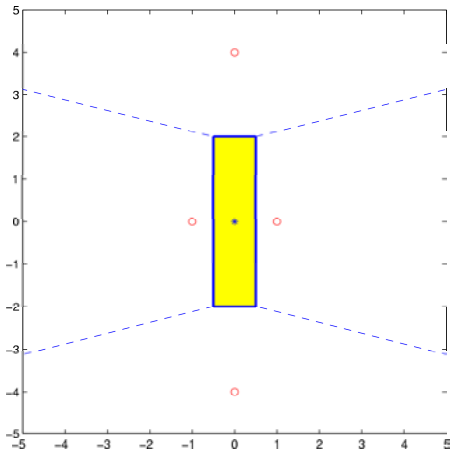


Fig. 1. An example of uneven contribution of normal vectors. White dots: class A code vectors. Black dot: class B code vector.

to be $\lambda_1 = \lambda_2 = 0.5$, $u_1 = [1\ 0]$, $u_2 = [0\ 1]$, suggesting that the two dimensions have the same discriminative power, while it is clear that projecting on the first dimension results in a minor accuracy loss than projecting on the second dimension. By defining the EDBFM as a weighted sum of normal vectors, where each normal vector is weighted by the length of the relative segment of decision border over the total length of the decision border, we get $\lambda_1 = 0.8$, $\lambda_2 = 0.2$, $u_1 = [1\ 0]$, $u_2 = [0\ 1]$, hence the first dimension correctly results four times more important than the second one. In section II-C, a discussion about the calculus of the volume of piecewise linear decision border is presented.

In the following, it is shown the algorithm of our proposed method:

Let $\mathcal{TS} = \{(t_1, u_1), \ldots, (t_T, u_T)\}$, $t_i \in \mathcal{R}^n$, $u_i \in \mathcal{C}$, be a set of T labeled samples (the training set),

Feature Extraction based on LVQ classifier

1. Train the LVQ $\{(m_1, l_1), \ldots, (m_M, l_M)\}, m_i \in \mathcal{R}^n,\ l_i \in \mathcal{C}$ on $\mathcal{TS}$ by using the BVQ algorithm;
2. set the elements of the matrix Σ_{EDBFM} to 0;
3. set w_{tot} to 0;
4. for each pair $\{c_i,\ c_j\}$, where $i \neq j$ do
   1. set the elements of the matrix $\Sigma_{EDBFM_{ij}}$ to 0;
   2. for each pair $\{m_k,\ m_z\}$, where $l_k = i$ and $l_z = j$ do
      1. calculate the vector normal to the decision border as: $N_{kz} = \frac{(m_k - m_z)}{\|m_k - m_z\|}$;
      2. calculate the decision border weight (volume) w_{kz};
      3. $w_{tot} = w_{tot} + w_{kz}$;
      4. $\Sigma_{EDBFM_{ij}} = \Sigma_{EDBFM_{ij}} + w_{kz} N_{kz} N_{kz}^t$;
   3. $\Sigma_{EDBFM} = \Sigma_{EDBFM} + p(c_i)p(c_j)\Sigma_{EDBFM_{ij}}$;
5. Set $\Sigma_{EDBFM} = \frac{\Sigma_{EDBFM}}{w_{tot}}$;

The eigenvectors ϕ_i of the Σ_{EDBFM} define the matrix $\Phi = [\phi_1, \phi_2, \ldots, \phi_n]$, that is exploited to transform the original space in a new space such that $x' = \Phi \cdot x$. The eigenvectors corresponding to the most large eigenvalues represent the most discriminant features. So, the matrix Φ' built with only the first n' most discriminant features, defines the transformation of the original space $\mathcal{R}^n$ in the reduced space $\mathcal{R}^{n'}$.

C. Volume of piecewise linear decision border

The geometric properties of an LVQ simplifies the calculus of the normal vectors, that are simply the difference between the two code vectors defining a piece of decision border. The problems are to identify which pairs define the decision border and to calculate the volumes of the pieces of hyperplane. The proposed solution to the first problem implies the calculus of M distances (where M is the number of code vectors) for each train vector, in order to define the first two nearest code vectors. Since the optimal number of code vectors depends on the classification problem, but it does not depend on the size T of the training set [3], the complexity of the this solution is linear in the size of the training set.

In order to calculate the volume of the piece of linear decision border, we propose both an analytical and a numerical approach. In the former approach, the equations of hyperplanes and of their borders can be derived by code vectors, and the volumes calculated analytically. However, such calculus is unpractical, since it is based on the n-dimensional Voronoi region definition and its complexity grows exponentially with the number of dimensions of the vector spaces [5]. We are investigating cheaper approaches to analytical calculus of volumes. In the latter approach we resort to the numerical integration. In practice, this can be done both by calculating the number of training samples that are classified by the piece of decision border ; and by defining a grid of points in the $\mathcal{R}^{n-1}$ space, and evaluating how many points belong to the piece of hyperplane. Such grid is defined by the cartesian product of $\mathcal{P} = \{p_1, \ldots, p_k\} \in \mathcal{R}$ with itself $n-1$ times. The accuracy of both numerical methods, of course, increases with the number of used points. Unfortunately, the size of the training set is often inadequate to guarantee a good integration accuracy, while we can arbitrarily choose the number of points belonging to the grid, changing the size k of $\mathcal{P}$. On the other hand, the number of grid points, and the complexity of this solution, grows exponentially with the number of dimensions of the vector space; while the computational cost of using the training set is linear in the size of training set. Then, the choice of the most suitable approach to calculate the volume of piecewise linear decision border depends on the classification problem.

III. Experiments

In the present section we show the performance of our method, compared with those of other feature extraction approaches, especially the MLPFE [13] and ADBFE [8], by means of artificial data. The experiment consists of three equiprobable classes $w1$, $w2$, $w3$ distributed according to the following statistics:

$$\mu_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 9 \end{bmatrix}.$$

$$\mu_{21} = \begin{bmatrix} 5 \\ 0 \\ 0 \end{bmatrix}, \Sigma_{21} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 9 \end{bmatrix} \textit{ and } \mu_{22} = \begin{bmatrix} -5 \\ 0 \\ 0 \end{bmatrix}, \Sigma_{22} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 9 \end{bmatrix}.$$

$$\mu_{31} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, \Sigma_{31} = \begin{bmatrix} 9 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 9 \end{bmatrix} \textit{ and } \mu_{32} = \begin{bmatrix} 0 \\ -5 \\ 0 \end{bmatrix}, \Sigma_{32} = \begin{bmatrix} 9 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 9 \end{bmatrix}.$$

The experiment is taken from [13], [8], where it was used to compare the performance of PCA, CA, DBFE, MLPFE and ADBFE approaches. Similarly to [8], 2000 samples from each class were generated, of which 500 where used for the training and the remaining for the test. We initialized an LVQ of order 20 with the first 20 training vectors, and we set $\Delta = 0.5$, $\gamma^{(0)} = 0.3$. The LVQ is then trained until the training error settle down around the value 0.141. We did not stress the setting of the parameters deliberately, in order not to take advantage neither of the knowledge of the class statistics, nor of the results in [13], [8]. As a result, the net shows an error probability on the test set of 0.1738, which

is slightly worse than that in [13] (0.143) and [8] (0.152). For the volume calculus we use a grid of 100 points in 2 dimensional space. Nevertheless, the feature extraction algorithm produces comparable eigenvalues and eigenvectors:

$$\lambda_1 = 0.5249,\ \lambda_2 = 0.3663,\ \lambda_3 = 0.1088,$$

$$\phi_1 = \begin{bmatrix} -0.054 \\ -0.998 \\ -0.016 \end{bmatrix}, \phi_2 = \begin{bmatrix} -0.998 \\ 0.054 \\ 0.012 \end{bmatrix}, \phi_3 = \begin{bmatrix} 0.011 \\ -0.017 \\ 1.000 \end{bmatrix}.$$

Table I compares the accuracy of the proposed method with that of competing methods, namely of MLPFE, ADBFE, PCA, CA, by showing the error performed by a nearest neighbor classifier on the data transformed according to the above approaches. In particular, we present the error probability when the most important feature is considered, when the first two features are considered and on the whole transformed space. By using the same classifier for each approach, we eliminate the influence of the classifier characteristics (in particular, MLP vs LVQ) and we can better appreciate the performance of the feature extraction methods. Error probabilities in Table I are averaged over 10 different data sets, and the related variances are also shown in brackets.

TABLE I

NEAREST NEIGHBOR ERROR PROBABILITY VS DIMENSION OF THE TRANSFORMED SPACE FOR THE BVQFE, MLPFE, ADBFE, PCA AND CA APPROACHES. TEST DATA.

feature No.	Error Probability				
	PCA	CA	MLPFE	ADBFE	BVQFE
1	0.489 ($1.8{\cdot}10^{-4}$)	0.590 ($6.7{\cdot}10^{-3}$)	0.455 ($2.0{\cdot}10^{-3}$)	0.477 ($2.1{\cdot}10^{-3}$)	0.445 ($2.3{\cdot}10^{-3}$)
2	0.219 ($6.9{\cdot}10^{-5}$)	0.468 ($6.0{\cdot}10^{-3}$)	0.208 ($3.8{\cdot}10^{-5}$)	0.209 ($8.1{\cdot}10^{-5}$)	0.211 ($1.5{\cdot}10^{-4}$)
3	0.218 ($6.8{\cdot}10^{-5}$)	0.212 ($1.0{\cdot}10^{-4}$)	0.217 ($5.2{\cdot}10^{-5}$)	0.217 ($5.2{\cdot}10^{-5}$)	0.215 ($4.6{\cdot}10^{-5}$)

We can see that accuracies obtained by using the BVQFE method are substantially the same of the MLPFE and ADBFE methods, and they are definitely better than those of the methods that do not exploit information about the decision border.

It was noted that MLPFE and ADBFE indirectly define the decision border from the estimation of the a-posteriori class probability, while the BVQ approach is designed to directly move the decision border defined by an LVQ towards the Bayes border. The use of a direct information about the decision border is an advantage in many cases since it is well known that an accurate estimation of the a-posteriori class probabilities leads to an accurate estimation of the decision border, however if a-posteriori class probabilities are not well estimated, nothing can be said on the accuracy of the estimated decision border. This advantage can be experimentally observed if, for the same experiment, we consider a training set of reduced size. Table II reports the average error probability and variance of the error performed by MLPFE, ADBFE and BVQFE methods when only 50 training vectors and 150 test vectors are used for each class. The results are averaged over 10 different data sets.

Notice that BVQFE both finds better features and is more robust: the variance of the error in the case of the best pair of features is an order of magnitude lower than that of both MLPFE and ADBFE. Nevertheless, the MLPs used in this experiment have on average the same mean squared error measured on the training set as the MLPs used in the experiments reported in Table I.

TABLE II

AVERAGE NEAREST NEIGHBOR ERROR PROBABILITY VS DIMENSION OF THE TRANSFORMED SPACE FOR THE BVQFE, ADBFE AND MLPFE METHODS. 50 TRAINING DATA. THE VARIANCE IS SHOWN IN BRACKETS.

feature No.	Error Probability (Variance)		
	MLPFE	ADBFE	BVQFE
1	0.534 $(3.2 \cdot 10^{-3})$	0.515 $(5.6 \cdot 10^{-3})$	0.472 $(2.7 \cdot 10^{-3})$
2	0.256 $(6.7 \cdot 10^{-3})$	0.252 $(8.4 \cdot 10^{-3})$	0.232 $(2.9 \cdot 10^{-4})$
3	0.242 $(5.4 \cdot 10^{-4})$	0.242 $(5.4 \cdot 10^{-4})$	0.241 $(6.8 \cdot 10^{-4})$

IV. CONCLUSIONS

The paper presented an approach to feature extraction for classification based on Vector Quantization. It is shown that Labeled Vector Quantizers allows for a cheap representation of decision borders, from which the most discriminative features can be extracted. Furthermore, the use of the BVQ algorithm to train the LVQ allows to define a robust and effective procedure for the estimation of true decision borders, which is truly based on the minimization of error probability. The result is a robust and accurate method for feature extraction that scales well with the dimension of the training set. The method is presented considering the minimum error probability as the guiding criterion, however notice that BVQ is a general algorithm for the minimization of the average misclassification risk. Hence, one more advantage of the approach is that it can directly manage real problems where the misclassification costs differ from one class to another (e.g for a banker, the cost of evaluating an unreliable client reliable for a loan is greater than evaluating a reliable client unreliable). A limit of the proposed method is the computational cost due to the calculus of the weights (volumes) of the pieces of hyperplanes defining the decision border. We are studying a cheaper approaches to analytical calculus of volumes based on Voronoi region definition.

REFERENCES

[1] C. H. Park, H. Park and P. Pardalos. A Comparative Study of Linear and Nonlinear Feature Extraction Methods. In *Proceedings of the Fourth IEEE International Conference on Data Mining*. IEEE, 2004.

[2] E. Choi and C. Lee. Feature extraction based on the bhattacharyya distance. *Pattern Recognition*, 36(8):1703–1709, 2003.

[3] C. Diamantini and M. Panti. An efficient and scalable data compression approach to classification. *ACM SIGKDD Explorations*, 2(2):54–60, 2000.

[4] C. Diamantini and A. Spalvieri. Quantizing for Minimum Average Misclassification Risk. *IEEE Trans. on Neural Networks*, 9(1):174–182, Jan. 1998.

[5] Diamantini, C., Potena, D. and Panti, M. KDD Support Services Based on Data Semantics. In Stefano Spaccapietra, editor, *Journal of Data Semantics IV*, volume 3730 of *LNCS*, pages 280–303. Springer, December 2005.

[6] J. H. Friedman. On Bias, Variance, 0/1-Loss, and the Curse-of-Dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77, 1997.

[7] K. Fukunaga. *Introduction to Statistical Pattern Recognition (2nd Edition)*. Ac. Press, San Diego, 1990.

[8] J. Go and C. Lee. Analytical decision boundary feature extraction for neural networks. In *Proc. IEEE Int. Symposium on Geoscience and Remote Sensing*, volume 7, pages 3072–3074. IEEE, 2000.

[9] J. Go and C. Lee. Optimality of decision boundary feature extraction for multiclass problems. In *Proc. IEEE Int. Symposium on Geoscience and Remote Sensing*, volume 3, pages 2051–2053. IEEE, 2003.

[10] T. Kohonen. The self organizing map. *Proc. of the IEEE*, 78(9):1464–1480, Sept. 1990.

[11] T. Kohonen, G. Barna, and R. Chrisley. Statistical pattern recognition with neural networks: benchmarking studies. In *Proc. of the IEEE Int. Conf. on Neural Networks*, volume 1, pages 61–68, San Diego, CA, May 1988.

[12] C. Lee and D.A. Landgrebe. feature extraction based on decision boundaries. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(4):388–400, Apr. 1993.

[13] C. Lee and D.A. Landgrebe. Decision boundary feature extraction for neural networks. *IEEE Trans. on Neural Networks*, 8(1):75–83, Jan. 1997.

[14] H. Liu and L. Yu. Toward Integrating Feature Selection Algorithms for Classification and Clustering. *IEEE Trans. on Knowledge and Data Engineering*, 17(4):491–502, April 2005.

Attribute Selection Methods for Filtered Attribute Subspace based Bagging with Injected Randomness (FASBIR)

I. M. Whittley[1], A. J. Bagnall[1], L. Bull[2], M. Pettipher[3], M. Studley[2] and F. Tekiner[3]

[1] School of Computing Sciences, University of East Anglia, Norwich, England
[2] School of Computer Science, University of West of England, England
[3] Department of Computer Science, University of Manchester, England

Abstract. Filtered Attribute Subspace based Bagging with Injected Randomness (FASBIR) is a recently proposed algorithm for ensembles of k-nn classifiers [28]. FASBIR works by first performing a global filtering of attributes using information gain, then randomising the bagged ensemble with random subsets of the remaining attributes and random distance metrics. In this paper we propose two refinements of FASBIR and evaluate them on several very large data sets.

keywords: ensemble; nearest neighbour classifier

1 Introduction

As part of an EPSRC project developing data mining tools for super computers, we are examining the best ways of employing emsembles of classifiers for data sets with a large number of attributes and many cases. *Filtered Attribute Subspace based Bagging with Injected Randomness* (FASBIR) is one of several recently proposed algorithms for ensembles of k-nn classifiers [28]. FASBIR works by first performing a global filtering of attributes using information gain, then constructing a bagged ensemble with random subsets of the remaining attributes and diversified distance measures.

The main contributions of this paper are, firstly, to collate and format a set of large attribute many case data sets used in related research and, secondly, to propose refinements for FASBIR that make it more suitable for use with these large datasets. Other algorithms proposed for ensembles of k-nn classifiers include Multiple Feature Subsets (MFS) [5] and Locally Adaptive Metric Nearest Neighbour (ADAMENN) [10]. We believe that FASBIR is the most promising approach for high dimensional data with a large number of observations. MFS will perform poorly with a large number of attributes. ADAMENN requires several scans of the data to classify a new case and hence is not appropriate for data set with many records. FASBIR reduces the dimensionality with a simple filter and is fast to classify new cases. However, three features of FASBIR (as

described in [28]) can be improved upon for dealing with large data. Firstly, the filter is performed on the whole data prior to ensembling. For large training sets, this can prohibitively expensive, particularly if there is little overlap in data cases between the ensembles. Secondly, FASBIR has a large number of parameters and traditional wrapper approaches for setting these values are not feasible for large data. Thirdly, FASBIR combines the output of the classifiers using a simple voting scheme. In this paper, for consistency with [28] we assume there is a large overlap in the data samples between classifiers, hence there is little to gain from localised filtering. Instead, we propose randomising parameters across the ensemble and using a probability based voting scheme.

The remainder of this paper is structured as follows. In Section 2 we briefly review related work. In Section 3 we describe the refinements introduced to improve FASBIR. In Section 4 we describe the data sets used in the experimentation reported in Section 5 and discuss our next objectives in the conclusions 6.

2 Background

Nearest neighbour (NN) classifiers, first proposed in 1951 by Fix and Hodge [16], are very simple forms of non-parametric, lazy classifiers that have remained a popular exploratory data analysis technique. A NN classifier consists of a training data set and a distance metric. New cases are assigned the class of the closest case in the training set. A common extension is k nearest neighbour (k-NN) [1], which involves finding the k nearest cases then assigning the most common class of these cases to the new case. NN search forms a core activity in many fields, such as time series data mining [24], pattern recognition [3] and spacial and multimedia databases [26]. A collection of the important papers on k-NN classifiers was published in 1990 [13].

The most commonly recognised problems with k-nn classifiers are that, firstly, they are sensitive to redundant features and secondly, classifying new cases is relatively time consuming for large data sets [2].

A recently popular research theme which partially addresses these problems is methods for combining several k-NN classifiers through *ensembles*. Ensemble techniques amalgamate the predictions of several classifiers through a voting scheme to form a single estimated class membership for each test case. Ensemble techniques can be categorised as two types: Sequential methods which iteratively build classifiers on the same data set with different weight functions (examples include ADABOOST [17], ARC-x4 [8], ECOC [25] and LogitBoost [18]); and parallel methods that construct classifiers concurrently on possibly different data sets, such as Bagging [7], Random Forest [9] and Randomization [14].

In the paper first introducing the Bagging technique [7] it is shown that bootstrapping k-NN classifiers does not result in better accuracy than that obtained with the whole model. If the ensemble samples are found by exhaustively sampling without replacement, and each ensemble still finds the closest k in the subset, then it is trivial to recreate the whole data k-NN from the ensemble. However, it is also well known that k-NN is sensitive to variation in the at-

tribute set. Since the objective of designing ensembles is to increase diversity while maintaining accuracy, these factors have meant that the major focus of research into ensembles of k-NN classifiers has been on methods to select subsets of attributes. For example, Bay [5] evaluates the effect using a random subset of attributes, called Multiple Feature Subsets (MFS), for each Nearest Neighbour member of the ensemble. MFS is evaluated when used with sampling with replacement and sampling without replacement on 25 small datasets from the UCI Repository of Machine Learning Databases [6]. MFS was compared to NN, k-NN and greedy feature selection algorithms (forward selection and backward elimination) described in [2]. MFS combines the votes of the individual classifiers. The method described in [23] also involves random attribute splits, but differs from MFS in that it combines the nearest neighbours of each ensemble rather than the votes. Random subspaces have also been used in [19]. Zhou's FASBIR [28] first measures the information gain of each attribute, then removes all the attributes with information gain less than some threshold. Bootstrapping samples are formed from the resulting dataset, and each classifier is assigned a random subset of attributes and a randomized distance metric (injected randomness). The algorithm is evaluated on 20 small data sets from the UCI repository. Domeniconi and Yann describe the Locally Adaptive Metric Nearest Neighbour (ADAMENN) classifier in [10] and how it could be used for ensembles in [15]. ADAMENN produces a probability distribution over the attribute space based on the Chi-squared statistic. It produces this distribution for each new case in the test data. For every classifier in the ensemble the attribute distribution is sampled (both with and without replacement) to form a subset of fixed size. The ADAMENN ensemble is evaluated on five small data sets from [6].

Filtering has been shown recently to be as effective as more complex wrapper methods on a range problems with a large number of attributes [21]. Since there are obvious speed benefits for filters over wrappers we believe that FASBIR is the most promising k-nn ensemble approach for the data with a large number of attributes and many cases.

3 FASBIR

The FASBIR [28] algorithm is summarised in Figure 3. A Minkowsky distance metric for ordinal attributes and the Value Difference Metric for nominal attributes is used. The set of distance functions for FASBIR is a set of possible values for the power p of the Minkowsky/VDM measure, which in [28] is restricted to the set $C = \{1, 2, 3\}$. The other parameters are listed in Table 1.

In this paper we test two refinements of FASBIR. The first is an improvement to the prediction mechanism. The algorithm as described in [28] uses simple majority voting. Each member of the ensemble predicts the class and these votes are collected to determine the predicted class. Our basic refinement is to make each classifier produce a probability estimate for class membership. These probabilities are then combined. This allows the ensemble to retain more of the

Algorithm 1 The FASBIR Algorithm

Given training data set D with n cases and m attributes, **train**

1. Filter the attributes
   (a) Measure IG on each attribute. Let a be the average information gain over the f attributes.
   (b) Discard any attribute with IG less than $f \cdot a$, giving m' attributes.
2. For each of the t classifiers in the ensemble
   (a) sample with replacement $n' = r \cdot n$ data
   (b) sample without replacement $s * m'$ of the filtered attributes
   (c) select a random distance measure from a set of candidates C

For each new case to classify **test**

1. For each of the t classifiers in the ensemble
   (a) Find the k nearest neighbours with selected distance metric
   (b) Return the majority class of the neighbours
2. Classify case as the majority class of all the ensemble votes

Table 1. FASBIR Parameters

Parameter	Meaning	Setting in [28]
f	proportion of attributes to filter	0.3
s	proportion of attributes to randomly select	0.25
r	proportion of data to sample	1
t	number of classifiers	20,40,60,80,100
k	number of neighbours	1,3,5,7,9

discriminatory power in the constituent classifiers. The second refinement is a generalisation of the parameter space. The parameter values for f and s are fairly arbitrary. For problems with redundant attributes, fixed cut off values may be sufficient to capture the important attributes. However, if there is multi-collinearity and deceptive/partially useful attributes, the filter may retain or remove fields of use. One of the driving forces in this algorithm's design is the need to diversify the classifiers. Hence, rather than have a fixed cut off value f and s, we randomised the filter value for each classifier in the ensemble. These refinements are assessed in Section 5.

4 Data Sets

We have collected 18 data sets, 9 from attribute and model selection competitions ([22, 11, 20, 12]), 2 standard sets from the UCI repository, 2 simulated sets and 5 new problems provided by contributors to our EPSRC project [27]. Summary information on the datasets is given in Table 2. Further information on all the data is available from [4]. We have included the very small Glass problem to provide validation that our results are comparable to those obtained in

the FASBIR paper. All continuous attributes are normalised using a standard normal transformation.

Table 2. Data Set Summary

Source	Name	Size(KB)	CASES	Atts	Data	Ordinal	Nominal	Classes
NIPS2003	Madelon	5,085	2600	500	1300000	500	0	2
PASCAL2004	Catalysis	2,660	1173	617	723741	617	0	2
WCCI2006	Hiva	12,155	3845	1617	6217365	1617	0	2
WCCI2006	Gina	7,554	3153	970	3058410	970	0	2
WCCI2006	Sylva	6,531	13086	216	2826576	216	0	2
WCCI2006	Ada	432	4147	48	199056	48	0	2
IJCNN2006	Temp	8,159	7177	106	760762	106	0	2
IJCNN2006	Rain	8,159	7031	106	745286	106	0	2
IJCNN2006	SO2	4,532	15304	27	413208	27	0	2
UCL	Adult	3,882	32561	14	455854	8	6	2
UCL	Glass	12	214	9	1926	9	0	7
Commercial	ProductW	71,873	715028	43	30746204	4	39	6
Commercial	ProductX	60,968	590943	44	26001492	4	40	6
Commercial	ProductY	34,283	339312	43	14590416	4	39	6
Commercial	ProductZ	23,304	224693	44	9886492	4	40	6
Hunt	Obesity	2,240	12429	89	1106181	49	40	4
Simulated	SGR	18,348	100000	100	10000000	100	0	2
Simulated	SGC	18,348	100000	100	10000000	100	0	2

5 Results

These experiments test the effectiveness of FASBIR with and without refinements on the 18 data sets described in Section 4. We compare FASBIR to a simple naive Bayes classifier and linear discriminant analysis, both of which use all the features, and to MFS. FASBIR Vote, is the algorithm proposed in [28]. FASBIR Prob uses probability distribution voting. FASBIR Random uses probability voting and has randomised parameters.

Table 3 gives the testing accuracy with a 10-fold cross validation on the 18 data sets. The experiments reported in Table 3 were performed with $k = 7$. Further testing not reported showed that a very similar pattern of results was produced with k values from 1 to 9. Looking at the balanced error rate rather than the accuracy also led to similar conclusions, so these statistics are omitted for brevity.

The first observation we can make from these results is that, with the exception of the two simulated data sets, FASBIR Random is always more accurate than both Naive Bayes (NB) and Discriminant Analysis (DA). The other FASBIR versions are generally more accurate than NB and DA. This is simply a demonstration the value of feature selection in high dimensional feature spaces, and serves as a basic sanity check.

Table 3. Test accuracy averaged over 10 folds. The highest figure is in bold

	Naive Bayes	DA	MFS	FAS Vote	FASBIR Prob	FASBIR Random
Madelon	59.15%	54.88%	59.38%	58.69%	57.88%	**59.81%**
Catalysis	67.26%	60.02%	69.39%	**70.84%**	69.56%	69.64%
Hiva	43.38%		96.70%	96.67%	96.67%	**96.70%**
Gina	75.33%	82.24%	83.45%	94.45%	**94.83%**	94.61%
Sylva	95.77%	98.67%	97.91%	99.30%	**99.34%**	99.29%
Ada	51.31%	84.35%	83.53%	84.50%	84.57%	**84.71%**
Temp	89.94%	92.82%	92.60%	93.26%	**93.34%**	93.23%
Rain	75.81%	78.27%	**79.21%**	79.12%	79.05%	79.04%
SO2	80.83%	87.11%	87.15%	87.49%	87.44%	**87.50%**
Adult	83.35%	81.00%	83.78%	84.60%	**85.35%**	85.06%
Glass	15.37%	45.26%	71.08%	72.08%	73.44%	**74.85%**
ProductW	41.57%	23.61%	43.25%	42.48%	**44.71%**	44.59%
ProductX	45.67%	27.25%	45.96%	48.16%	**49.99%**	49.27%
ProductY	51.85%	22.31%	55.13%	55.70%	55.63%	**55.70%**
ProductZ	48.82%	29.14%	51.34%	52.08%	52.49%	**52.51%**
Obesity	44.36%	54.40%	54.67%	54.59%	**54.93%**	54.45%
SGR	64.99%	**98.21%**	67.95%	70.92%	71.02%	71.19%
SGC	**86.73%**	86.69%	85.91%	86.16%	86.35%	86.29%

The second observation we can make from these results is that filtering provides a benefit for ensembles of k-nn. Each FASBIR implementation has greater accuracy than MFS on 15 of the 18 data sets. If we view the data as a paired sample, we can reject the hypothesis that the difference in accuracy is zero using a sign test, a Wilcoxon sign rank test and a t-test, even after removing the outlier Gina. With Rand Fasbir and MFS, there is a significant difference between the classifiers on 12 out of the 18 data, as measured by McNemar's test. There is no significant difference on the other 6 data sets. These results corroborate the findings of [28].

Thirdly, we observe that there is a definite advantage in combining the probability estimates of the individual classifiers in the ensemble rather than combining votes. If we compare majority voting FASBIR and FASBIR with probability voting (columns 5 and 6 in Table 3), we see that probability voting is better on 12 out of the 18 data sets. This difference is also observable if we look at the difference between MFS with and without voting and FASBIR random with and without voting. For all three pairs, there is a significant difference between the classifiers on 6 out of the 18 data, and no significant difference on the remaining sets. This indicates there is no penalty for using the probabilities, and on many occasions a significant improvement can be achieved.

Fourthly, randomising the filter and selection parameters has no significant any effect on accuracy. Randomisation significantly improves performance on Madelon and SGC, and this results demonstrates the potential benefit of avoiding a fixed filter value. Both Madelon and SGC have correlated attributes. This multicollinearity is exactly the type of situation where an arbitrary filter may

remove relevant attributes. The benefits are small and our experiments do not conclusively support the use of randomised parameters across the ensemble. However, because of the benefits of reducing the parameter space and increased robustness, we believe it is a sensible approach.

6 Conclusions and Future Directions

Very large, many attribute data sets offer a unique type of challenge that is being faced by data mining practitioners with increasing frequency. Many approaches to attribute selection are simply not feasible with such massive data. Attribute filters are a simple and effective method that have been effective with this kind of data. In this paper we have collated several disparate sources of very large, many attribute data sets, including five never used before. We have proposed some minor modifications of a recently published filtering algorithm using in conjunction with k-nn, and demonstrated that filtering improves performance on the majority of these large data.

The next stage of this work will be to collect more data, to experiment with alternative distance metrics and evaluate alternative greedy randomised atribute selection algorithms.

Acknowledgement

This research was funded by the EPSRC under grant GR/T18479/01, GR/T18455/01 and GR/T/18462/01.

References

1. D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.
2. D.W. Aha and R.L. Bankert. Feature selection for case-based classification of cloud types: an experimental comparison. In *Proc. AAAI 94*, pages 106–112, 1994.
3. S. N. Srihari B. Zhang. Fast k-nearest neighbor classification using cluster-based trees. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(4):525–528, 2002.
4. A. Bagnall. Large data sets for variable and feature selection. `http://www.cmp.uea.ac.uk/ ajb/SCDM/AttributeSelection.html`, 2006.
5. S. D. Bay. Combining nearest neighbor classifiers through multiple feature subsets. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 37–45, 1998.
6. C.L. Blake and C.J. Merz. UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 1998.
7. L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
8. L. Breiman. Arcing classifiers. *Annals of Statistics*, 26(3):801–849, 1998.
9. L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
10. D. Gunopulos C. Domeniconi, J. Peng. Locally adaptive metric nearest-neighbor classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(9):1281–1285, 2002.

11. J Q. Candela, C. Rasmussen, and Y. Bengio. Evaluating predictive uncertainty challenge, presented at NIPS 2004 workshop on calibration and probabilistic prediction in machine learning. `http://predict.kyb.tuebingen.mpg.de/`, 2004.
12. G. Cawley. Predictive uncertainty in environmental modelling competition, special session at ijcnn-2006. `http://clopinet.com/isabelle/Projects/modelselect/`, 2006.
13. B. Dasarathy. *Nearest Neighbor Pattern Classification Techniques*. IEEE Computer Society Press, 1990.
14. T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.
15. C. Domeniconi and B. Yan. On error correlation and accuracy of nearest neighbor ensemble classifiers. In *the SIAM International Conference on Data Mining (SDM 2005)*, 2005.
16. E. Fix and J. L Hodges. Discriminatory analysis, non-parametric discrimination: consistency properties. Technical report, USAF School of aviation and medicine, Randolph Field, 1951.
17. Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *2nd European Conference on Computational Learning Theory*, pages 23–37, 1995.
18. J. H. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28:337–407, 2000.
19. R. Sabourin G. Tremblay. Optimizing nearest neighbour in random subspaces using a multi-objective genetic algorithm. In *17th International Conference on Pattern Recognition (ICPR'04)*, 2004.
20. I. Guyon. Model selection workshop, part of the IEEE congress on computational intelligence. `http://clopinet.com/isabelle/Projects/modelselect/`, 2006.
21. I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003.
22. I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. NIPS 2003 workshop on feature extraction. `http://www.nipsfsc.ecs.soton.ac.uk/`, 2003.
23. T. K. Ho. Nearest neighbors in random subspaces. In *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*, pages 640–648, 1998.
24. E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Mining and Knowledge Discovery*, 7(4):349–371, 2003.
25. E. B. Kong and T. G. Dietterich. Error-correcting output coding corrects bias and variance. In *Machine Learning: Proceedings of the Twelfth International Conference*, pages 313–321, 1995.
26. T. Seidl and H. Kriegel. Efficient user-adaptable similarity search in large multimedia databases. In *VLDB '97: Proceedings of the 23rd International Conference on Very Large Data Bases*, pages 506–515, 1997.
27. F. Tekiner. Super computer data mining project. `http://www.mc.manchester.ac.uk/scdm/toolkit`, 2006.
28. Z.-H. Zhou and Y. Yu. Ensembling local learners through multi-modal perturbation. *IEEE Transactions on systems, man, and cybernetics - part B*, In press - 2005.

A Novel Effective Distributed Dimensionality Reduction Algorithm

Panagis Magdalinos, Christos Doulkeridis, Michalis Vazirgiannis
Department of Informatics, AUEB, Greece
{pmagdal,cdoulk,mvazirg}@aueb.gr

Abstract
Dimensionality reduction algorithms are extremely useful in various disciplines, especially related to data processing in high dimensional spaces. However, most algorithms proposed in the literature assume total knowledge of data usually residing in a centralized location. While this still suffices for several applications, there is an increasing need for management of vast data collections in a distributed way, since the assembly of data centrally is often infeasible. Towards this end, in this paper, a novel distributed dimensionality reduction (DDR) algorithm is proposed. The algorithm is compared with other effective centralized dimensionality reduction techniques and approximates the quality of FastMap, considered as one of the most effective algorithms, while its central execution outperforms FastMap. We prove our claims through experiments on a high dimensional synthetic dataset.

Keywords: Distributed dimensionality reduction, clustering, distributed knowledge discovery

1. Introduction

Dimensionality reduction algorithms are extremely useful in various disciplines, especially related to data processing in high dimensional spaces. The latter becomes a difficult task as dimensions increase, because of the two distinct problems: the “empty space phenomenon” and “the curse of dimensionality” [1],[2]. The first denotes the fact that in high dimensional spaces data is sparsely situated, appearing at equal distance from one another. The “curse of dimensionality” on the other hand refers to the fact that in the absence of simplifying assumptions, the sample needed to estimate a function of several variables to a given degree of accuracy grows exponentially with the number of variables. A thorough investigation considering both aforementioned facts from the perspective of the nearest neighbor retrieval can be found in [20],[19].

Besides high dimensionality, another problem encountered in the area of data processing is the large amount of data. Data is not always situated on a single machine, but is usually scattered in a network. The latter is more obvious nowadays with the emergence of several novel applications such as peer-to-peer, sensor networks, data streams, etc. The ability to collect, store, process and subsequently index huge amounts of data has necessitated the development of algorithms that can extract useful information from distributed data corpuses. The scientific field of distributed knowledge discovery (DKD) addresses this issue. Distributed knowledge discovery is divided into two distinct categories. *Homogeneous*, where resources queried are arbitrarily distributed among nodes although described by the same features and *heterogeneous*, where all participants share the same knowledge but described by different features. Another possible categorization is acquired when information is considered as a huge resources x features matrix. If the rows or the matrix are shared among peers then the distribution is called horizontal (which is analogous to homogeneous) while the division of columns denotes a case of vertical distribution (equal to heterogeneous).

This paper proposes a novel effective dimensionality reduction algorithm that enables the compression of data processed, while retaining information for subsequent clustering or classification purposes. The algorithm proposed however exhibits the ability of distributed execution tackling the issue of distributed dimensionality reduction (DDR) from the perspective of a distributed, homogeneous knowledge discovery problem. Despite the distributed nature of the

approach, the reduction and indexing performance produced approximates the one exhibited by a well-known centralized algorithm, namely FastMap ([17]).

The paper is organized as follows: in Section 2 we review the related work regarding dimensionality reduction techniques. In Section 3, we identify the requirements for a distributed dimensionality reduction algorithm, while in Section 4, we present the novel algorithm. In Section 5, the experimental results are presented, and finally in Section 6, we conclude the paper.

2. Related Work

Each dimensionality reduction algorithm must fulfill some requirements in order to be considered effective and efficient. Briefly stated, the prerequisites are: a) the discovery of the intrinsic dimensionality of the dataset, b) the preservation of correlation dimensions between data, while projecting to a lower dimensionality space, and c) the least possible loss of information.

One of the initial methods proposed is the multidimensional scaling technique (MDS) often referred today as classic MDS (http://www.statsoft.com, http://www.diap.polimi.it). MDS is an explorative technique of data analysis that provides a depiction of the processed dataset in a lower dimensionality space with the usage of correlation information. In general, MDS can be considered as a methodology for dimensionality reduction proposing the use of numerical analysis transformations on data until a certain criterion is maximized or minimized.

The best dimensionality reduction approach is Principal Components Analysis [1], [2]. PCA achieves high stress minimization and high level of mutual information preservation. The algorithm applies singular value decomposition on the correlation matrix and retains only the k greatest singular values and vectors. In general, all singular value decomposition based methods exhibit high quality of results. Latent Semantic Indexing (LSI - [18]) is a special case, because the process utilized for the projection also manages to capture and bring forward semantic information contained in data. If the Stress criterion of MDS is replaced by the level of mutual information preservation the method in question is Independent Component Analysis [1],[2]. In the case of PCA, the use of the negative entropy function, as defined by Shannon, produces the Projection Pursuit method [2].

One of the fastest methods available in this area is FastMap [17]. FastMap maps data from dimension n to n-1 by projection on a hyperplane perpendicular to the line defined by the two most distant points in the processed space. Recursive application of this procedure achieves the projection of N point from space R^n to subspace R^k in O(Nk) time while retaining distances among data. The Discrete Fourier Transform ([4] - DFT), is another method for fast projection and compression of data, which perceives each point as a series of randomly selected instances of a continuous signal and transforms it to a sum of basic signals. Afterwards, basic signals that do not add up to the final reconstruction are rejected; consequently, the corresponding coordinates are absconded thus resulting in the compression and reduction of data. PAA (Piecewise Aggregate Approximation) [4] is a close relative of DFT that projects each point independently from the rest. After fixing a window size f, all sets of f coordinates are replaced by their mean value. The main drawback of this fast approach (O(n)) is its dependence on the size of the initial window. If the latter is big, then sharp changes in data will be lost, as all will be smoothed to their mean value.

All previously presented algorithms except from MDS, are classified as linear, because they try to project data in a globally linear space of lower dimensionality. On the contrary, non-linear methods try to preserve linearity in the locality of each point. By adding up the local linear fractals of projection space one can achieve the formation of a non-linear projection space satisfying our

requirements. Prominent methods employed for non-linear dimensionality reduction are the spring models [8], self organizing (Kohonen) maps [5], neural networks [1][2] and non-linear PCA [2]. The general idea of non-linear projection has recently steered much research in the field of dimensionality reduction. Isomap [8], C-Isomap [12] and Local Linear Embedding [11],[10] are relatively new methods for non-linear reduction. The most novel approach presented in bibliography is Landmark MDS or shortly LMDS [3]. The major goal of LMDS is the provision of a dimensionality reduction approach adequate for large datasets that cannot be loaded on main memory. The cost of this approach is $O(2kbN + k^2N + b^3)$ (N being the cardinality of the projected set, b the number of landmark points and k the dimensionality of the projecting space) assuming that no heuristic is used. If a heuristic is employed for the selection of the initial points then a O(bN) factor is added to the aforementioned cost.

3. Requirements of a DDR Algorithm and Applicability of Centralized Algorithms

The aim of this section is the identification of some initial requirements that a dimensionality reduction method must fulfill, in order to be used in distributed environments, along with an evaluation of the applicability of the previously described centralized algorithms in this context. Before dwelling in further analysis, some assumptions are stated. It is assumed that all resources can be described as points in a high dimensional space, i.e. R^n, while the latter is common to all participating nodes that form a network. Moreover no node can have global knowledge of the data/corpus being processed, but only a small fraction. Both assumptions anagoge the problem to a horizontally distributed knowledge discovery problem.

Given a dimensionality reduction algorithm and a dataset of N resources, distributed arbitrarily among the nodes of a network, the following requirements must hold for the distributed execution of an algorithm: (1) Each point should be projected to the new subspace independently from the rest of the dataset.. (2) Distances between points should be preserved while projecting to a new subspace. The latter must hold true both locally and globally. Given two points A, B, their distance (d) in the high dimension space, and their distance (d') in the projection space, the algorithm must guarantee that these values will be preserved even when the points belong to different network nodes. (3) The algorithm should be fast to compute, and linear to the total number of points projected.

The vast majority of dimensionality reduction techniques attempt to map points in a low dimensional space by exploiting the correlations among them. This is not tolerable in our case, because no node can acquire full knowledge of the data shared by the network. As an example, one can imagine the use of LSI, PCA and in general all SVD based methods. In the case of LSI or PCA, the abruption of certain singular values and singular vectors retains only the dimensions that provide the most valuable information regarding the correlation of the data, while discarded information is regarded as noise. There is no way to ensure however the validity of the comparison of two models generated by two different corpuses. The reason is rather simple and straightforward. Correlation dimensions initially perceived as noise and thus discarded, could carry valuable information, if SVD would have been carried out on the union of the two corpuses. Furthermore, SVD based methods, especially LSI, appear to have low scaling ability because of their complexity (N^3, N being the size of the resources correlation matrix) and the fact that when vast amounts of data are processed it is not easy to distinguish noise from information.

One could argue however, that SVD is applicable in horizontally distributed data. Although this is the case, the cost of applying an SVD update algorithm is equal to the cost of re-calculating the decomposition [13], while the folding in technique (addition of data based on the assumption that the decomposition is not influenced by new information) deteriorates quickly [14]. The Discrete

Fourier Transform, although it satisfies the first and third requirement, discards dimensions in the depiction of the transformed signal, based on their significance. In our case, this would prevent even local comparison of data, because the discarded dimensions would differ among resources.

Only two of the presented methods can be applied in our case, LMDS and PAA. In the case of LMDS, a node can be arbitrarily chosen and assigned the task of reducing the initial points, which are provided by the rest. Afterwards, both projected and original data can be broadcasted across the network and each node may proceed independently. What the "adapted" LMDS achieves with high complexity and network traffic, PAA can achieve it with relatively no cost. The major drawback in this case is the size of the rolling window. If the latter is big (reduced dimensionality << original dimensionality) and the points are sparse then all variation will be lost.

4. The Proposed Algorithm

An algorithm with lower complexity than LMDS and lower network traffic would be an adequate solution to our problem. The DDR algorithm presented in this section is an attempt to reach these standards, while fulfilling the requirements stated in the previous section. The approach follows the general principles of the LMDS adaptation, while differentiating in the way each step is achieved and exhibiting lower complexity and network traffic. The setup of the problem is the same. Given N resources represented as points of R^n, distributed arbitrarily in a network of p nodes, we want to find a projection of the data in R^k, while retaining distances and the ability to perform clustering afterwards. Each node is assumed to possess $\lceil p/N \rceil$ resources. The algorithm is divided into four distinct steps.

Step 1*:* An aggregator node is selected. The selection can be made randomly or based on same kind of "built in" heuristic (i.e. a transformation of the IP address of nodes) as described in [16]. The aforementioned node is assigned all tasks that need to be executed centralized.

Step 2*:* Afterwards, k points must be sampled from the whole dataset and forwarded to the selected node. Each node selects and forwards $\lceil k/p \rceil$ points resulting in O(nk) network traffic load. The selection can be made with one of the following ways:

- Each node randomly selects from the resources owned $\lceil k/p \rceil$ points.
- Each node selects the $\lceil k/p \rceil$ most far off points of its collection trying to create a kernel of points with long connections among them. We refer to this heuristic as MaxDist. The cost of the selection is $O(\lceil k/p \rceil)$, when random selection is employed, and $O(\lceil N/p \rceil \lceil k/p \rceil)$, when MaxDist is used.

Step 3*:* Selected points are projected by the aggregator in the R^k space with the use of the FastMap algorithm and all data (original coordinates of resources and projections) are flooded to the rest of the peers. The initialization of the FastMap algorithm needs $O(k^2n)$ time and its execution $O(k^2)$, while the broadcasting of the result produces $O(nk + k^2)$ network traffic.

Step 4*:* In the final step of the procedure, each node is obliged to project the resources owned to the new subspace with the use of the provided points (hereafter referred as *landmark points*). During the projection, the algorithm attempts to preserve distances, meaning that the resource projected must have equal distance from the landmark points both in the original and in the projection space. If x is the projected point, L the set of k landmark points and l_i the landmark points then this requirement is stated as $\| x^{(k)} - l_i^{(k)} \| = \| x^{(n)} - l_i^{(n)} \|$ for i=1..k. The algorithm searches the common trace of all k hypersheres, which is in fact the projection of point x in the reduced space. The result can easily be obtained by solving the above system of nonlinear equations with the Newton method. If the approximation is precise, then the algorithm converges, otherwise the algorithm deviances

and produces a result after the completion of a certain amount of iterations. This step produces on each node a load analogous to $O((\lceil N/p \rceil - \lceil k/p \rceil)\, k^3/3)$.

For any set of points the algorithm will produce a solution if the triangular inequality is sustained in the original space. For any point S of the initial space and the landmark points A, B equation $\|\overrightarrow{AB}\| \leq \|\overrightarrow{SA}\| + \|\overrightarrow{SB}\|$ (1) holds true. The system defined for the projection ($\|\overrightarrow{SA}\| = \|\overrightarrow{S'A'}\|$, $\|\overrightarrow{SB}\| = \|\overrightarrow{S'B'}\|$) does not have a solution, if there exists no common trace between the created hyperspheres. This is translated as $\|\overrightarrow{A'B'}\| \geq \|\overrightarrow{S'A'}\| + \|\overrightarrow{S'B'}\|$ or equally $\|\overrightarrow{A'B'}\| \geq \|\overrightarrow{SA}\| + \|\overrightarrow{SB}\|$ (2) since $\|\overrightarrow{SA}\| = \|\overrightarrow{S'A'}\|$ and $\|\overrightarrow{SB}\| = \|\overrightarrow{S'B'}\|$ by default. After projecting A, B with FastMap the original and projected distances between these points are associated through inequality $\|\overrightarrow{A'B'}\| \leq \|\overrightarrow{AB}\|$ (3). Consequently based on (3), (1) we conclude that equation (2) is never true, meaning that the system in question always has a solution (there always exists a projection) provided that the triangular inequality is sustained in the original space. Moreover, the time needed to compute this solution depends only on the approximation vector provided initially to the Newton method and the accuracy factor ε.

To sum up, the proposed algorithm differs from other widely employed dimensionality reduction approaches for three distinct reasons. Initially, the projection of the vast majority of points is done independently from the rest, meaning that only the landmark points affect the projection. Moreover, landmark points remain unaffected by subsequent projections while the projection itself is independent of the sampled data. Finally the minimization criterion employed by the algorithm is $\sum_{|L|}\{|distance_{orig} - distance_{new}|\}$, applied to each point independently, contrary to the widely employed Stress function that is applied to the whole set of data.

Compared to the distributed LMDS adaptation - also proposed in this paper- our algorithm exhibits lower network load and computational complexity. Indeed, distributed application of LMDS produces $O(2bn + bk)$ network traffic and requires $O(k^2\lceil N/p \rceil + bk\lceil N/p \rceil)$ time for all nodes, while for the aggregator the load is $O(k^2\lceil N/p \rceil + bk\lceil N/p \rceil + b^3)$. Note that b is larger than k in all cases and signifies the number of points selected for the execution of LMDS. On the other hand our algorithm produces $O(2nk + k^2)$ traffic load and requires $O((\lceil N/p \rceil - \lceil k/p \rceil)\, k^3/3)$ time. This value is augmented at the aggregator node for an amount of $O(k^2)$ due to the execution of FastMap.

Apart from the lower complexity, the proposed algorithm comes with one more advantage against the distributed application of LMDS. The sampling procedure can be carried out once in the lifetime of a network and the result can be forwarded to all nodes entering the network at any time. Projection is independent of the sample, because each resource is projected to a point abstaining analogously far or close in the reduced space. On the contrary, since LMDS employs classic MDS that requires the solution of a generalized eigenvector problem, updates have to take place periodically, since content changes affect the projection.

5. Experimental Results

In an attempt to evaluate the proposed algorithm, a series of experiments on a synthetic dataset was carried out. The goal was to prove the validity of the approach while exhibiting results of quality close to well-known centralized approaches. In all experiments, we arbitrarily created a set of high dimensional vectors, which constructed a set of ten well separated clusters, so as to ensure that the applicability of clustering is unaffected by the high dimensionality of the processed space. The clustering algorithm employed was K-Means.

The data generator takes as input the number of vectors (s), and the number of clusters (c) to be created. All vectors coordinates are initialized by values belonging to [0,1]. At the second step the

generator produces a set of c different integers (p_i, i=1…10). Finally, each set of $\lceil s/c \rceil$ vectors changes the p_i coordinate of the elements contained to 5. This value ensures that each set of $\lceil s/c \rceil$ vectors is well separated from the rest, meaning that no overlapping occurs between clusters.

The points were subsequently projected to a predefined lower dimensionality space through the usage of four different algorithms. The first algorithm, which has been used as a point of reference, was FastMap. Afterwards, two different setups of our new algorithm were employed. The first (named DDR-R) used a random sample of initial data, while the other (named DDR-H) employed the MaxDist heuristic. PAA was also tested in order to evaluate its stability and quality in large-scale reduction processes. Finally, K-Means was employed in order to evaluate the clustering quality after the reduction. The Newton method employed by our algorithm utilizes as an approximation vector the perpendicular projection of the point (referred to as x) to the new subspace with every coordinate augmented by a factor $(a^2-1)||x||$ (a=0.7)

Results presented in this paper come from the projection of 1000 vectors of dimensionality 2000 to dimensions 10, 20, 40, 80, 100. Due to space limitations, three more sets of experiments are omitted, but can be found in the extended version of this paper [21].

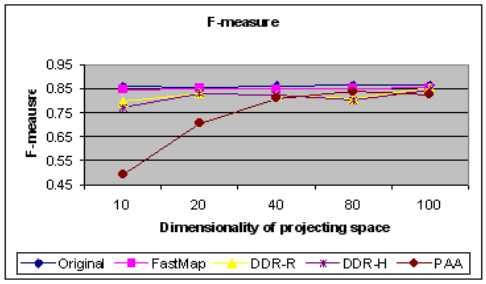


Figure 1: Deviation of clustering quality

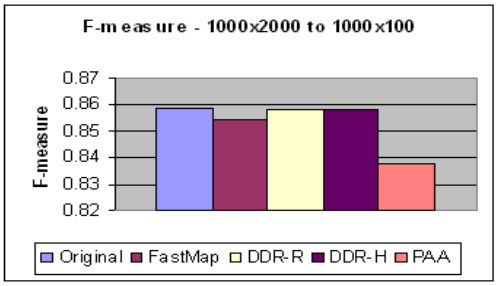


Figure 2: Outperforming FastMap in clustering quality maintenance

As far as clustering is concerned, figure 1 gives valuable insight and allows us to draw some initial conclusions. With a sampling of only 2%-4% of data, high quality projection and clustering is achieved. F-measure is in fact less than 5% lower than the one achieved with centralized projection of the data (FastMap). Moreover, when projecting from initial dimensionality 2000 to 100 dimensions, both DDR-R and DDR-H outperform FastMap, as exhibited in figure 2.

Another interesting result is that the method is not influenced by the way the initial set of points are selected, allowing in fact the usage of random sampling and thus lowering the complexity of the process. The projection quality is measured by computing the stress value. All experiments exhibited the same behavior, producing a very low stress value, almost equal to the one exhibited by FastMap. Moreover, the mathematically proven fact that the stress value decreases as projection dimension increases was also demonstrated. Finally, the projection was unaffected by the way initial points were sampled. Figure 3 demonstrates these facts, while Figure 4 demonstrates the time requirements of all four approaches.

In all above experiments, our algorithm was executed in a distributed way, as described in the previous sections. However, one can also employ this algorithm in a centralized way. In this case, the best way to choose the initial points is the execution of a clustering algorithm, namely K-Means with the usage of HAC as initialization process. After the latter's completion, the number of clusters generated is supplied to our algorithm as the projection dimension and the centroids calculated as the landmark points. When our algorithm was evaluated with the aforementioned

setup, it outperformed FastMap, reaching even more than 10% better clustering quality, together with an extremely low stress value. Figures 5, 6 demonstrate this fact in the projection of 300 points from an initial dimension of 1000 to 10.

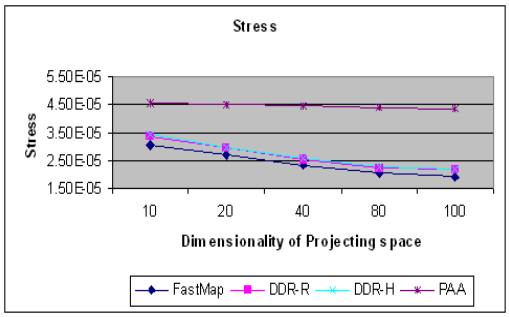


Figure 3: Projection quality

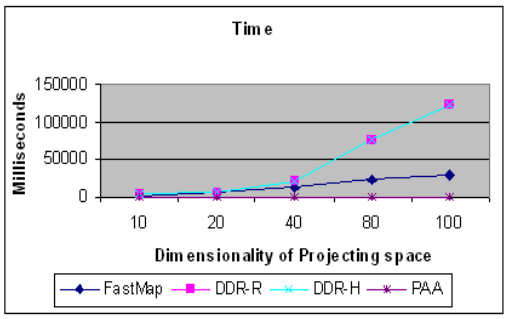


Figure 4: Time requirements

To sum up, the experimental evaluation presented in this section, leads to a primary conclusion stating that the proposed algorithm offers the possibility of distributed dimensionality reduction for large datasets providing projection quality equal to a centralized approach, namely FastMap. Furthermore, clustering the reduced data projected by our algorithm, retains high quality, marginally equal to the one achieved, when performing clustering in the original space (note that the initial clusters were well separated). Results obtained from clustering on the projections of the centrally executed FastMap, and our distributed executed algorithm exhibit the same quality. On the other hand, the use of the MaxDist heuristic does not ameliorate results. Finally, when our algorithm was used as a centralized dimensionality reduction approach and was evaluated against FastMap, it produced better quality results both in terms of F-Measure and Stress values.

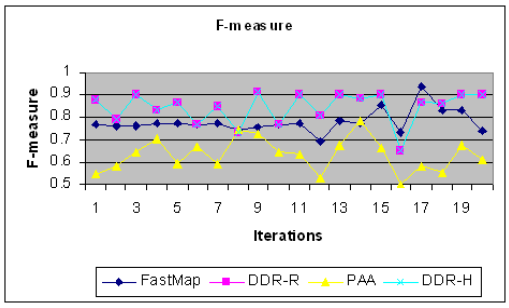


Figure 5: Clustering quality in centralized execution

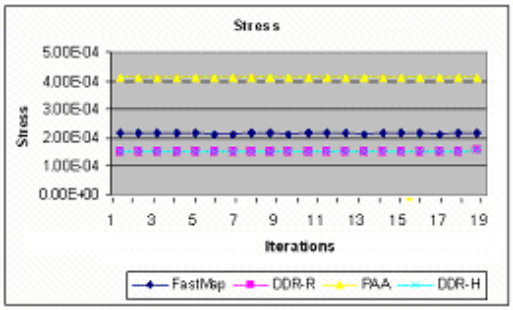


Figure 6: Stress in centralized execution

6. Conclusions and Future Work

This paper tackled the issue of distributed dimensionality reduction from the perspective of a distributed, homogeneous knowledge discovery problem. The bibliographic research indicated the absence of any appropriate solution to this problem. Furthermore, only one of the centralized approaches could be adjusted to fit our requirements. To the best of our knowledge, our approach and the distributed LMDS adaptation, both presented in this paper, are the first to provide a solution to this problem. However, our algorithm is the first approach that directly targets the problem of distributed dimensionality reduction. The quality of our results is almost equal to FastMap, measured in terms of Stress and F-measure values, while our algorithm's central execution outperforms FastMap. Future work will primarily concentrate on evaluating our algorithm with real

datasets against LMDS and PCA. The last comparison will demonstrate the viability of our approach against the best dimensionality reduction algorithm in the bibliography.

7. References

[1]. "A Survey of Dimension Reduction Techniques", *I.K. Fodor*, US Department Of Energy, 2002

[2]. "A Review of Dimension Reduction Techniques", *M.Carreira*, Technical Report, University of Stanford, 1997

[3]. "Sparse Multidimensional Scaling Using landmark points", *Vin de Silva, Joshua B. Tenenbaum*, 2004

[4]. "Dimensionality Reduction of Fast Similarity Search in Large Time Series Databases", *E.Keogh, K. Chakrabarti, M.Pazzani, S.Mehrotra*, Knowledge and Information Systems – Springer-Verlag, 2001

[5]. "Self-Organization of Very Large Document Collection: State of the Art", *Teuvo Cohonen*, ICANN 1998

[6]. "A Linear Iteration Time Layout Algorithm for Visualising High-Dimensional Data", *Matthew Chalmers*, 7th IEEE Visualization Conference, 1996.

[7]. "Nonlinear Dimensionality Reduction by Locally Linear Embedding", *Sam T.Roweis, Lawrence K.Saul*, Science Magazine (www.science.org) 2000.

[8]. "Unsupervised Learning Of Curved Manifolds", *Vin de Silva, Joshua B. Tenenbaum*, In Proceedings of the MSRI workshop on nonlinear estimation and classification. Springer Verlag, 2002.

[9]. "Fast Multidimensional Scaling through Sampling, Springs and Interpolation", *Alistair Morrison, Greg Ross, Mathew Chalmers*, Information Visualization, Vol.2, Issue 1, 2003.

[10]. "Think Globally, Fit Locally, Unsupervised Learning of Nonlinear Manifolds", *T.Roweis, Lawrence K.Saul*, Technical Report, 2002

[11]. "A Global Geometric Framework for Nonlinear Dimensionality Reduction", *Vin de Silva, Joshua B. Tenenbaum, John C.Langford*, Science Magazine (www.science.org) 2000

[12]. "Global versus local methods in nonlinear dimensionality reduction", *Vin de Silva, Joshua B. Tenenbaum.*, NIPS 2003

[13]. "Information Management Tools for Updating SVD-encoded indexing schemes", *O'Brien*, Master Thesis, University of Tennessee, 1994

[14]. "A Divide-and-Conquer Approach to the Singular Value Decomposition", *Jane Tougas*, Seminar on Machine Learning and Networked Information Spaces, University of Dajhousie, 2004

[15]. "A Semidescrete Matrix Decomposition for Latent Semantic Indexing in Information Retrieval", *Tamara G.Kolda, Dianne O'Leary*, ACM Transactions on Information Systems, Vol.16, No.4 October 1998.

[16]. "DESENT: Decentralized and Distributed Semantic Overlay Generation in P2P Networks", *C.Doulkeridis, K.Noervaag, M.Vazirgiannis*. Technical Report, DB-NET, AUEB, 2006 (submitted for publication).

[17]. "FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets", *Christos Faloutsos, David Lin*, ACM SIGMOD 1995.

[18]. "Indexing by Latent Semantic Analysis", *S.Deerwester, S.Dumais, T.Landauer, G.Fumas, R.Harshman*, Journal of the Society for Information Science, 1990

[19]. "When is 'Nearest Neighbor' Meaningful?", *K.Beyer, J.Goldstein, R.Ramakrishan, U.Shaft*, ICDT 1999

[20]. "What is nearest neighbor in high dimensional spaces?", *A.Hinneburg, C. Aggarwal, D.Keim*, VLDB 2000.

[21]. "A Novel Effective Distributed Dimensionality Reduction Algorithm", P.Magdalinos, C.Doulkeridis, M.Vazirgiannis. Technical report, 2006, available at: http://www.db-net.aueb.gr/index.php/publications/technical_reports

An Ensemble Method for Identifying Robust Features for Biomarker Identification

Diana Chan[1] Susan M. Bridges[1] Shane Burgess[2]
nc7@cse.msstate.edu, bridges@cse.msstate.edu, burgess@cvm.msstate.edu
[1]Department of Computer Science & Engineering
[2]Department of Basic Sciences, College of Veterinary Medicine
Mississippi State University
Mississippi State, MS 39759

Abstract

Identification of biomarkers for disease from mass spectrometry proteome profiles has recently become the subject of a great deal of research. These data sets are categorized as "wide data" because the number of features is much larger than the number of instances. Many different statistical and machine learning approaches have been applied to select a small number of features from the very large feature sets, but it has proven difficult to identify biomarkers that are robust in the sense that they provide reproducibly high accuracy with new datasets. We describe a framework for feature selection for wide data that is based on the intuition that features that are consistently selected under varying preprocessing steps, feature selection methods, and classification algorithms are more likely to be robust. An ensemble method is used that rewards features that are selected often and that occur in small feature sets that result in accurate classification. We demonstrate that, with Petricoin's ovarian cancer data set, the features selected by our method yield accurate classifiers, overlap the feature sets reported by other researchers, and, most importantly, can be used to build an accurate classifier for new data.

Keywords: Feature Selection, Ensemble, Machine Learning, Classification, Biomarkers

1 Introduction

Identification of biomarkers has recently become an active field of research for disease detection and monitoring. Biomarkers for cancer are of particular interest because early detection greatly improves the probability of successful treatment. A biomarker is a protein or set of proteins found in the blood, other body fluids, or tissues that has a distinct pattern of expression under certain conditions [15]. A seminal study by Petricoin et al. [17] reported the use of mass spectrometry to classify the serum proteome profiles of ovarian cancer patients. According to Coombes et al. [6], as of 2005, over 60 published studies have used similar techniques for profiling a number of different types of cancer and other diseases. Most of the studies have used relatively low resolution surface-enhanced laser desorption and ionization (SELDI) mass spectrometry although some of the more recent studies have used higher resolution matrix-assisted laser desorption and ionization and time-of-flight (MALDI-TOF) data. The resulting data sets consist of data instances with tens of thousands of features. The total number of data instances available is generally quite limited (fewer than 200). This type of data set where the number of features is much greater than the number of instances is often referred to as "wide" data. Researchers have applied a variety of statistical and machine learning approaches for analysis of mass spectrometry profiles. Classification accuracy with a single data set is typically high, but it has proven quite difficult to reproduce the results with new data sets. In addition, different data mining procedures will often select different sets of features from the same data set [3]. Our goal is to develop an ensemble-based feature selection method that will result in a robust set of features in the sense that they provide reproducibly high accuracy with new datasets.

Ensemble approaches are typically used to build robust classifiers where a number of different classifiers vote to provide the class for a new sample. However, in our case the ensemble of classifiers is used to vote for features rather than class labels. Because of the size of the data instances and the inherent noisiness of the data, data mining procedures for analysis of mass spectrometry profiles typically involve

a number of data preprocessing, feature selection, and model building steps. The choices used for different aspects of the data mining procedure have a substantial effect on the features selected and the accuracy of the resulting classifier. We describe a framework for feature selection for wide data that is based on the intuition that features that are consistently selected under varying preprocessing steps, feature selection methods, and classification algorithms are more likely to be robust. The ensemble method rewards features that are selected often and that result in accurate classification. The major steps in the process are shown in Figure 1.

1. Establish a general data mining process.
2. Generate an ensemble of classifiers by using different options at different stages in the data mining process.
3. Use a voting procedure for features that rewards features that occur in many accurate classifiers.
4. Build and test a classifier with the features accruing the most votes.

Figure 1 Ensemble method for feature selection

The main idea of ensemble methods is to allow better generalization among classifiers in order to achieve more accurate overall classification. Our work extends the idea of ensemble methods from classification to feature selection. Although it is critical to identify feature sets that maximize classification accuracy, some features seem to be chosen more frequently than other features no matter what feature selection or classification algorithm is used. We believe that these repeated features are more robust and are more likely to produce reproducible classification results.

We demonstrate that, with Petricoin's ovarian cancer data set [5], the features selected by our method yield accurate classifiers, overlap the feature sets reported by other researchers, and, most importantly, can be used to build an accurate classifier for new data.

2 Related Work

Petricoin et al. [17] first reported the use genetic algorithms with self-organizing maps to discriminate between non-cancer and ovarian cancer SELDI samples. Their methods were able to correctly classify all ovarian cancer samples and achieved a 95% of classification accuracy for non-cancer samples. Subsequent research has tried to locate biomarkers for various types of cancer from both SELDI and MALDI-TOF data including ovarian cancer [21], breast cancer [11], prostate cancer [1] and lung cancer [4]. Examples of computational methods that have been used include statistical feature selection with principal component analysis (PCA) and linear discriminate analysis (LDA) [12] and T-test with random forests [25].

Ensemble methods have been shown to generally be more effective than single classifiers and have become widely used for classification. For example, Tan and Gilbert used ensemble learning with classifiers of bagged and boosted decision trees with gene expression data. They found that an ensemble of decision trees always performed better than a single decision tree in classification [22]. Liu et al. [13] used a combinational feature selection and ensemble neural network method to classify gene expression data for cancer classification. The performance of classification was greatly improved using the outputs of several neural networks rather than a single neural network.

3 Data sets and data mining procedure

In this research, we use publicly available ovarian cancer datasets to demonstrate the capabilities our approach for biomarker identification. The SELDI 8-7-02 and 4-3-02 Ovarian Datasets were downloaded from the Clinical Proteomics Program Databank website [5].

The 8-7-02 dataset includes serum profiles of 91 non-cancer controls and 162 cancer subjects. This dataset was constructed using the Ciphergen WCX2 ProteinChip array. Each spectrum has two columns. The first column contains mass–to-charge ratios (m/z values) and the second column is the corresponding relative amplitude of the intensity. Each spectrum consists of 15,154 distinct m/z values with intensity values ranging from 0.0000786 and 19995.513.

Data mining for biomarker identification is critically dependent on a number of pre-processing steps including intensity normalization, statistical peak pre-selection and binning. The features that are selected and the accuracy of the resulting classifiers are dependent on the preprocessing steps used. Our overall data mining process is similar to that used by Sorace and Zhan for the same data [21]. The steps in this process are shown in Figure 2.

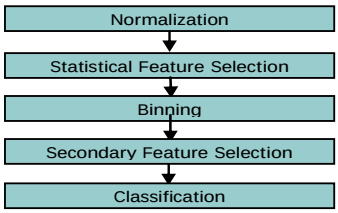


Figure 2 Data Mining Process for Biomarker Selection

Data-mining Steps	Options
1. Normalization	None, NV, Z-Score
2. Statistical Feature Pre-Selection	Wilcoxon Test
3. Binning	Minimum p-value, Maximum average intensity
4. Feature Selection Methods	None, CFS, Wrapper, PC
5. Classification	Neural Net, Naïve Bayes

Figure 3 Options for steps in data mining process

A number of different choices are available for each of these steps. In the following sections we describe the methods we have used to demonstrate the capabilities of our ensemble approach. Figure 3 summarizes the different options available for steps in the data mining process.

3.1 Data Normalization

Two normalization procedures were used for this study: z-score and normalized value (NV). Normalization is used to make feature values more comparable across all samples. A z-score relates individual intensity values to the population mean (M) and variance (S). The normalized value (NV) method places all values into the range 0 to 1. NV normalization has been used by a number of research groups working with biomarker discovery including Baggerly et al. [3]. We also used intensity values without normalization as a third alternative for this step.

3.2 Statistical Feature Selection

In situations where the number of features is huge compared to the number of samples, the major challenge is to select a few relevant and non-redundant features to distinguish cancer from non-cancer samples. The two-sided Wilcoxon test is used for the first level of feature selection to compare the intensity at each of the m/z values for all samples. The Wilcoxon test is a nonparametric test that is used to test the null hypothesis that the cancer (X) and control (Y) populations have the same continuous distribution [7]. We assume that we have independent random samples $x_1, x_2, \ldots, x_m$ and $y_1, y_2, \ldots, y_n$, where *m* is the number of samples from the cancer population and *n* is the number of samples from the control population. For each m/z value, the intensities for the cancer and control samples were merged and ranked in ascending order. For this research, the Wilcoxon test was used to identify the m/z values that are most discriminative between cancer and control samples. The lower the Wilcoxon test p-value, the better the m/z value distinguishes between the sick and healthy samples. The m/z values were first sorted according to p-value from lowest to highest. The 100 m/z values with the lowest p-values were selected as the initial feature subset for later feature selection.

3.3 Binning

Many of the m/z values that pass the Wilcoxon filter represent the same peak. A binning procedure similar to that used by Sorace and Zhan [21] was used to combine these values. The 100 m/z values with the lowest p-values and their corresponding intensities were sorted in ascending order by m/z values. Consecutive m/z values were combined if they were separated by less than 1 m/z. This resulted in 24, 23 and 18 bins for un-normalized data, NV normalized data and z-score normalized data respectively. Two different methods were used to choose a representative m/z value for each bin. For the first method, the

m/z value with the lowest p-value in each bin was selected. This m/z value indicates that it is the most discriminating value between the cancer and control samples for the bin. The second method selects the m/z value with the highest average intensity value across all samples as the representative m/z value.

3.4 Secondary feature selection

Pre-processing steps 1-3 provide six different combinations of features in terms of normalization and binning. Several different methods of secondary feature selection were considered. The goal of this feature selection step is to derive the best subset of features for classification in step 5. Feature selection at this stage is a search through the space of possible combinations of features and is driven by two procedures: attribute evaluation and search. The attribute evaluator is used to determine the quality of the individual feature for classification. The search procedure determines how the search space of possible features is explored. Different combinations of attribute evaluation and search procedures were used for feature selection to generate different feature subsets for the ensemble method.

Three attribute evaluators were used for secondary feature selection: correlation-based feature selection (CFS), wrapper-based evaluation (Wrapper) and principal components analysis (PCA). The CFS, Wrapper and PCA methods were selected because they are fast algorithms that have proven useful in many applications.

The CFS algorithm is a correlation-based heuristic that selects features that are highly correlated with the class, but are uncorrelated with other features [8]. A major advantage of the CFS algorithm is the generation of non-redundant feature subsets.

The wrapper based evaluation (Wrapper) approach uses the classification algorithm as part of the feature selection process. It selects attribute subsets based on the classification accuracy of classifiers trained with the feature subsets with training data. The advantage of this approach is that the bias of the classifier is considered during feature selection.

Principal components analysis (PCA) is a statistical, unsupervised learning technique that is commonly used for dimensionality reduction without information loss. PCA analysis has proven to be very effective in locating relevant features for classification of cancer data [27]. The original attributes are transformed by calculating their corresponding covariance matrix and extracting the eigenvectors as the principal components. The PCA algorithm ranks the eigenvectors based on the degree of variation in the original data that each accounts for. Those that account for 95% of the variance were selected in this case. This ranking is the search procedure for PCA.

Three different search procedures were used with the CFS and Wrapper attribute evaluation methods: greedy search, best-first search and genetic search. Greedy search performs a greedy search through the space of attribute subsets starting with an empty set of features and adding features (forward search) or starting with a full set of features eliminating features (backward search). The search stops if there is no improvement in the expanded subset. We have used a forward search procedure. Best-first search is very similar to greedy search except that the search is improved by backtracking [19]. Best-first search expands the best candidate subset at each point in a greedy fashion but also maintains an ordered list of previous best subsets that it can use for backtracking. Genetic search explores the state space using a genetic algorithm to search the state space by maximizing a fitness function [9]. Different subsets of features are represented as chromosomes. The population of chromosomes evolves over several generations using crossover, mutation and selection. In our experiments we ran the algorithm for 20 generations using a population size of 20, mutation rate of 0.033 and crossover rate of 0.6. The selection was randomized with the probability of selection proportional to the degree of fitness.

3.5 Classifiers

The “goodness” of feature subsets was evaluated by the accuracy of classifiers based on the subsets. This study used two types of classifiers: back propagation neural networks and naïve Bayes classifiers [16]. Both types of classifier have been previously shown to be effective for cancer classification. For example, Khan et al. [10] used the small, round blue-cell tumors as a model to train the neural nets with

perfect classification accuracy and was able to identify the most relevant genes for classification. Liu et al. [14] used naïve Bayes as one of the classification models for the acute lymphoblastic leukemia dataset and the ovarian cancer dataset and achieved satisfactory classification results for both datasets.

3.6 Implementation

The Wilcoxon statistical analysis was performed using SAS Version 9.1 (a statistical package from SAS Institute Inc., Cary, NC, USA) [20]. The NPAR1WAY procedure was used in SAS to run the Wilcoxon test to locate the initial feature subset. The Weka (Waikato Environment for Knowledge Analysis) software package [24] was used for secondary feature selection and classification. Weka was developed by researchers at the University of Waikato in New Zealand. It is a Java-based machine learning open source software package that implements many commonly used machine learning algorithms. It is publicly downloadable from: www.cs.waikato.ac.nz/ml/weka. Other steps in the process were implemented in Perl.

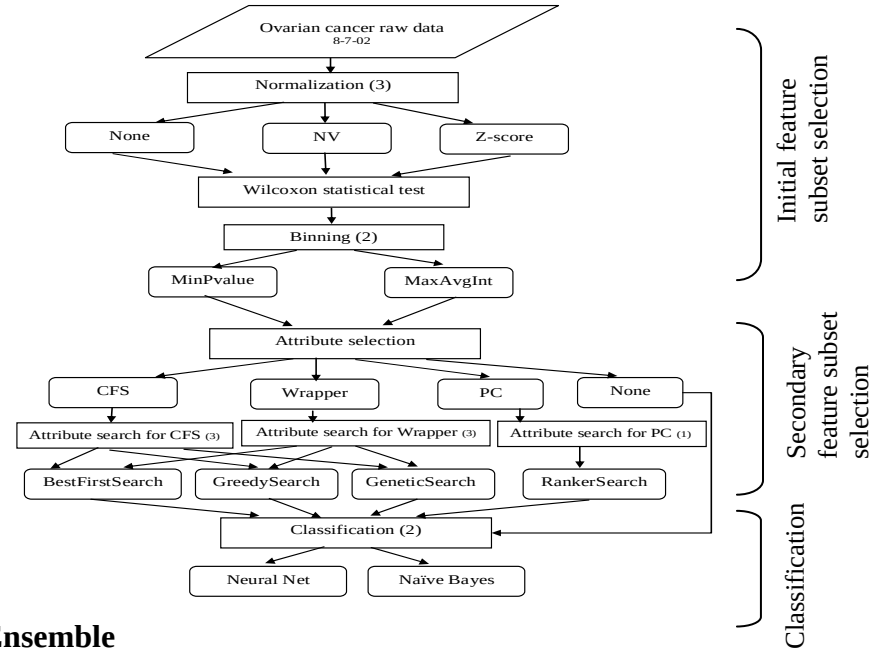


Figure 4 Flowchart of all possible combinations of available options for data preprocessing, feature selection, and classification. The number in parentheses indicates the number of options available for a specific decision.

4 Feature Selection Ensemble

Ensemble methods require both a method for generating members of the ensemble and a voting procedure. We have generated the ensemble by using different combinations of options for the data mining procedure. Figure 4 shows a flowchart of all possible combinations of decisions for the data pre-processing, feature selection and classification. Each path represents a member of the ensemble. There are 96 different paths from normalization to classification in the flowchart resulting in 96 different classifiers. Each of these classifiers was used as part of an ensemble for feature selection using the process shown in Figure 1. A total of 47 unique features were selected by at least one classifier and were evaluated by the voting procedure.

The voting procedure for our ensemble methods works as follows. For each feature (m/z value) that was selected for use by at least one classifier, both a feature score and a weighted feature score were computed. Note that m/z values within 1 were considered to be the same and the feature with the highest weighted feature score was selected over other scores of other consecutive m/z values. The feature score for a feature f_j is the sum of the accuracy values for all classifiers that included the feature. This score rewards features that are selected often by accurate classifiers. The weighted score is a modification of the feature score where the accuracy for each classifier is divided by the number of features selected. This scoring method favors frequently selected features that are members of small feature sets resulting in accurate classifiers.

More formally, the feature score, $s(f_j)$, and weighted feature score, $ws(f_j)$ for feature f_j are defined as follows:

$$s(f_j) = \sum_{i-1}^{N} e_{ij} a_i \qquad (1)$$

$$ws(f_j) = \sum_{i=1}^{N} (1/F_i) e_{ij} a_i \qquad (2)$$

where N is the number of classifiers, $e_{ij}=1$ if f_j is a feature selected for classifier i, a_i is the accuracy of classifier i and F_i is the number of features for classifier i.

After all features were scored, a classifier was constructed using the highest scoring features as described in the next section.

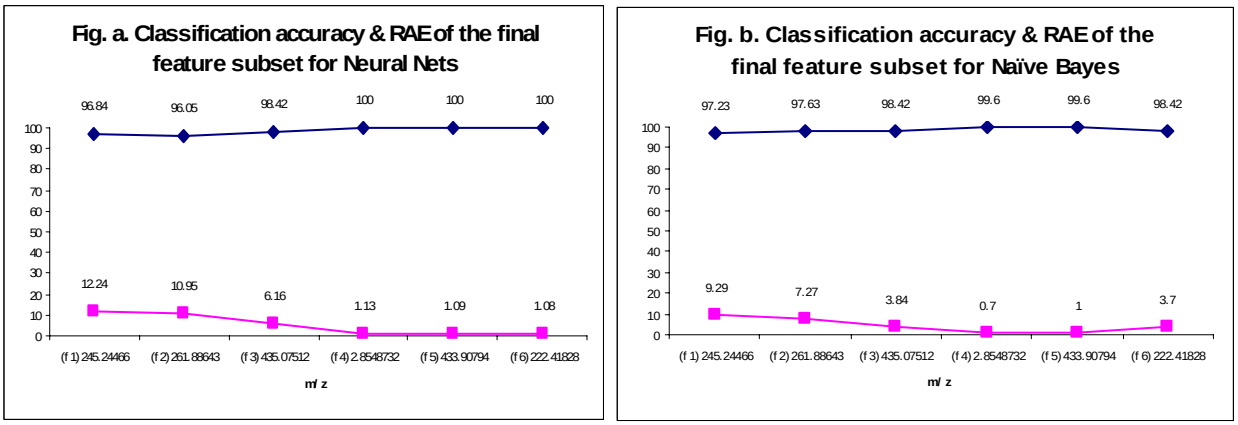


Figure 5 Accuracy of classifiers trained with features selected by the ensemble.

5 Results and Discussion

The ensemble feature selection method provides both a feature score *s* and weighted feature score *ws* for each feature selected by any classifier in the ensemble. Preliminary experimental results indicated that *ws* scoring outperforms *s* scoring. In this paper we only present results with *ws* scoring. When building the final classifier, the m/z with the highest *ws* was first added to the feature subset for classification. Features continued to be added ordered by *ws* until both classification accuracy and relative absolute error (RAE) did not improve. When used with the 8-7-02 data set, this procedure resulted in the selection of six features: m/z = 245.24466, 261.88643, 435.07512, 2.8548732, 433.90794 and 222.41828 which are referred as f_1, f_2, f_3, f_4, f_5 and f_6 respectively. Perfect classification accuracy was achieved for the first four features with the neural net classifier. The addition of features f_5 and f_6 provided not only perfect classification accuracy but also the minimum RAE. RAE is the relative absolute error computed by comparing the square root of the mean squared error with the one obtained if the prediction had been the mean. Ten-fold cross validation was used for all experiments.

Figure 5 shows graphs of the classification accuracy and RAE for both neural net and naïve Bayes classifiers based on the highest scoring features. The graphs plot the classification accuracy and REA as each additional feature was added to the final feature subset. In Figure 5a, the classification accuracy remains the same with the addition of features, f_5 and f_6, but the RAE continues to improve. Use of both accuracy and RAE for feature selection results in a classifier with high accuracy. In Figure 5b, naïve Bayes requires four features in order to achieve a classification accuracy of 99.6% with a minimum RAE of 0.7%.

As an additional validation step, the set of six features selected using the 8-7-02 dataset was subsequently used to build a classifier for a different data set, the 4-3-02 Ovarian Dataset [5]. The 4-3-02 dataset consists of samples from 50 unaffected women and another 50 patients with ovarian cancer [17]. Preprocessing was done using NV normalization and the classifier was a neural network. Ten-fold cross validation with the 4-3-02 dataset using the six features selected with the 8-7-02 dataset resulted in neural net classifier with a classification accuracy of 83%. If the top ten features selected using the 8-7-02 dataset are used, a neural network with classification accuracy of over 90% can be obtained with the 4-3-02 dataset using 10-fold cross validation. Baggerly et al. [3] were not able to reproduce results from these

two datasets using features selected by genetic algorithm partly due to calibration problems. Although our methods do not classify the 4-3-02 dataset perfectly, our results still demonstrate the robustness of the selected features. Figure 6 shows the features selected by other researchers. We consider two m/z values as the same if the difference between the two values is less than 1. Some of the features we selected overlap with features selected by others.

6 Conclusions and Future Work

The literature suggests that there are many different approaches for feature selection that can be effectively used to locate biomarkers of disease. Each approach has its own advantages and disadvantages. However, it has proven to be difficult to replicate biomarker selection using different feature selection methods due to the multi-factorial nature of the features [3]. It has also proven to be difficult to use features selected from one set of data for classification of another data set. We offer an ensemble framework for feature selection for building classifiers with wide data. Features that are selected often, that result in accurate classifiers, and that are part of small feature sets are considered to be more robust. We demonstrate that the features selected by this method give reproducible results with new data.

Baggerly et al. [3] demonstrated that it is difficult to select features that produce reproducible results across experiments. They found that some features that were good discriminators among samples in one experiment did not give satisfactory results in other experiments. Variation in experimental procedures and calibration of equipment was often a problem. These authors suggested the development of guidelines for design and analysis in experiments in order to have reproducible, biologically significantly results. Although our methods do not address many of the issues related to collection and analysis of mass spectrometry data, we do demonstrate that an ensemble approach for feature selection can help provide reproducible results across experiments.

We have demonstrated the effectiveness of our ensemble-based feature selection approach with the widely studied ovarian cancer datasets. We have used a general data mining process with different options for each step to demonstrate the effectiveness of our method. The general approach we describe can easily be used with different options for each data mining step or with a different data mining process. For example, one might incorporate use of recently developed peak finding algorithms such as super smoother [26] and wavelets [18, 23] into the process.

We have also successfully applied our ensemble feature selection approach to an unpublished MALDI data set for identification of nutritional deficiencies and plan to test the approach with additional MALDI and SELDI data sets as well as other types of wide data.

Alex at al. [2]	**245.8296**, **261.88643**, 336.6502, 418.8773, **435.46452**, 437.0239, 465.97198, 687.38131, 4004.826
Sorace and Zhan [21]	2.7921, **245.53704**, **261.8864**, 418.1136, **435.0751**, 464.3617, 4003.645 (these features were selected by stepwise discriminant analysis according to Rule 1)
Vannucci at al. [23]	**245.3**, **433.2, 434.6**, 243.9, 430.6, 241.3, 437.2, 605.2, 431.9
Our work	**2.8549, 222.4183, 245.2447, 2661.8864, 433.9079, 435.0751**

Figure 6 Comparison of selected features.

Acknowledgements

This work was partially supported by funding provided by the Institute for Neurocognitive Science and Technology (INST) at Mississippi State University.

References

1. Adam, B., Qu, Y., Davis, J.W., Ward, M.D., Clements, M.A., Cazares, L.H., Semmes, O.J., Schellhammer, P.F., Yasui, Y., Feng, Z., Wright, Jr. G.L. (2002). Serum protein fingerprinting coupled with a pattern-matching algorithm distinguishes prostate cancer from benign prostate hyperplasia and healthy men. *Cancer Research*, 62(13), 3609-3614.

2. Alexe, G., Alexe, S., Liotta, L., Petricoin, E., Reiss, M., and Hammer, P. (2004) Ovarian cancer detection by logical analysis of proteomic data. *Proteomics*, 4, 766-783.
3. Baggerly, K.A., Morris, J.S., and Coombes, K.R. (2004) Reproducibility of SELDI-TOF protein patterns in serum: comparing datasets from different experiments. *Bioinformatics*, 20(5), 777-785.
4. Baggerly, K.A., Morris, J.S., Wang, J., Gold, D., Xiao, L.C., and Coombes, K.R. (2003) A comprehensive approach to the analysis of matrix-assisted laser desorption/ionization-time of flight proteomics spectra from serum samples. *Proteomics*, 3, 1667-1672.
5. Clinical proteomics program database detailed explanation of proteome quest for data analysis. http://home.ccr.cancer.gov/ncifdaproteomics/ppatterns.asp (accessed on Aug 2005)
6. Coombes, K.R.,Morris, J.S.,Hu, J.,Edmonson,S.R.,and Baggerly,K.(2005)*Nature Biotechnology*,23(3),291-292.
7. Daniel, W. (1978) *Applied Nonparametric Statistics.* Houghton Mifflin.
8. Hall, M. and Smith, L. (1996) Practical feature subset selection for machine learning. *Proceedings of the Australian Computer Science Conference (University of Western Australia)*, February 1996.
9. Jeffries, N.O. (2004) Performance of a genetic algorithm for mass spectrometry proteomics. BMC Bioinformatics, 5, 180.
10. Khan, J. et al. (2001) Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine,* 7, 673 – 679.
11. Li, J., Zhang, Z., Rosenweig, J., Wang, Y.Y., Chan, D.W (2002) Proteomics and bioinformatics approaches for identification of serum biomarkers to detect breast cancer. *Clinical Chemistry*, 48(8), 1296-1304.
12. Lilien, R.H., Farid, H., and Donald, B.R. (2003) Probabilistic Disease Classification of Expression-Dependent Proteomic Data from Mass Spectrometry of Human Serum. *Journal of Computational Biology*, 10(6), 925-946.
13. Liu, B., Cui, Q., Jiang, T., and Ma, S. (2004). A combinational feature selection and ensemble neural network method for classification of gene expression data. *BMC Bioinformatics* , 5, 136.
14. Liu, H., Li, J. and Wong, L. (2002) A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. *Genome Informatics*, 13, 51–60.
15. Medical terminology & drug database. www.stjude.org/glossary (accessed on Dec 2005)
16. Mitchell, T. (1997) *Machine Learning.* McGraw Hill.
17. Petricoin,E.F., et al. (2002) Use of proteomic patterns in serum to identify ovarian cancer.*Lancet*, 359,572-577.
18. Qu, Y. et al. (2003) Data reduction using a discrete wavelet transform in discriminant analysis of very high dimensionality data. *Biometrics* 59, 143−151.
19. Russell, S. and Norvig, P (1995). *Artificial Intelligence: A Modern Approach.* Prentice Hall, Upper Saddle River, New Jersey.
20. SAS Users Manual, http://www.sas.com (accessed on Aug 2005)
21. Sorace, J.M. and Zhan, M. (2003) A data review and re-assessment of ovarian cancer serum proteomic profiling. *BMC Bioinformatics*, 4, 24.
22. Tan, A.C. and Gilbert, D. (2003) Ensemble machine learning on gene expression data for cancer classification. *Appl Bioinformatics*: 2(3 Suppl):S75-83.
23. Vannucci, M., Sha, N., and Brown, P. (2005) NIR and mass spectra classification: Bayesian methods for wavelet-based feature selection. *Chemometrics and Intelligent Laboratory Systems* 77, 139-148.
24. Witten, I. H. and Frank, E. (2005) *Data mining: Practical Machine Learning Tools and Techniques*, 2nd Edition, Morgan Kaufmann, San Francisco.
25. Wu, B., Abbott, T., Fishman, D., McMurray, W., Mor, G., Stone, K., Ward, D., Williams, K., and Zhao, H. (2003). Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. *BioInformatics*, 19(13).
26. Yasui, Y. et al. (2003) An automated peak identification/ calibration procedure for high-dimensional protein measures from mass spectrometers, *J. Biomed. Biotechnol*, 242−248
27. Yeoh, E.J. et al. (2002) Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling, *Cancer Cell*, 1, 133-143.

An ICA-based Feature Selection Method for Microarray Sample Clustering

Lei Zhu
Department of Information Technology
Armstrong Atlantic State University
11935 Abercorn Street
Savannah, Georgia 31419-1997
lzhu@cs.armstrong.edu

Chun Tang
Center for Medical Informatics
Yale University
300 George Street, Suite 501
New Haven, CT 06511
chun.tang@yale.edu

Abstract

DNA microarray technology can be used to measure expression levels for thousands of genes in a single experiment, across different samples. In sample clustering problems, it is common to come up against the challenges of high dimensional data due to small sample volume and high feature (gene) dimensionality. Therefore, it is necessary to conduct feature selection on the gene dimension and identify informative genes prior to the clustering on the samples. This paper introduces a method utilizing independent component analysis (ICA) for informative genes selection. The performance of the proposed method with various array datasets is illustrated.

Keywords: Bioinformatics, Microarray Analysis, Sample Clustering, Independent Component Analysis, Feature Selection

1 Introduction

DNA microarray technology can be used to measure expression levels for thousands of genes in a single experiment, across different samples [6, 7]. Experimental samples can include types of cancers, diseased organisms, or normal tissues. Arrays are now widely used in basic biomedical research for mRNA expression profiling and are increasingly being used to explore patterns of gene expression in clinical research [5, 17, 23, 24, 27]. Applying this technology to investigate the gene-level responses to different drug treatments could provide deep insight into the nature of many diseases as well as lead in the development of new drugs.

In a typical microarray experiment, raw microarray data (images) are first obtained from fluorescence scanners or phosphorimagers, then those images are transformed into gene expression matrices where usually the rows represent genes, and the columns represent various samples. The numeric value in each cell characterizes the expression level of the particular gene in a particular sample. Each row vector of a gene expression matrix represents an expression pattern of a gene, and each column vector is an expression profile of a sample.

Currently, a typical microarray experiment contains 10^3 to 10^4 genes, and this number is expected to reach the order of 10^6. However, the number of samples involved in a microarray experiment is generally less than 100. One of the characteristics of gene expression data is that it is meaningful to analyze from both the gene dimension and the sample dimension. On one hand, co-expressed genes can be grouped

in clusters based on their expression patterns [9, 4, 12]. On the other hand, the samples can be clustered into homogeneous groups. Each group may correspond to some particular macroscopic phenotype, such as clinical syndromes or cancer types [3, 11, 25]. Therefore, both gene clustering and sample clustering are important issues in gene expression data analysis. For example, they are usually the basis for discriminating cancer tissues from healthy ones and revealing biological functions of certain genes.

In sample clustering problems, it is common to come up against the challenges of high dimensional data (i.e., curse of dimensionality) due to small sample volume and high feature dimensionality. High-dimensional data not only bring computational complexity, but also degrade a classifier's performance. In addition, traditional clustering techniques may not be effective in detecting the sample patterns because the similarity measures used in these methods are based on the full gene space and cannot handle the heavy noise existing in the gene expression data. Therefore, it is necessary to conduct feature selection on the gene dimension and identify informative genes prior to the clustering on the samples. In [28], an algorithm named CLIFF (Clustering via Interative Feature Filtering) has been introduced to address the feature selection problem. In [26], a new model called empirical sample pattern detection (ESPD) was proposed to delineate sample pattern quality with informative genes. This paper introduces a method utilizing Independent Component Analysis (ICA) [15, 16] for feature selection and informative genes identification for microarray sample clustering.

Linear transformation methods transform the data into some new space that has some desirable properties. Principal component analysis (PCA) [18] and Independent component analysis (ICA) [15, 16] are two linear transformation methods widely used in microarray analysis. PCA projects the data into a new space spanned by the principal components. Each successive principal component is selected to be orthogonal to the previous ones, and to capture the maximum information that is not already present in the previous components. PCA is probably the optimal dimension-reduction technique according to the sum of squared errors. Applied to expression data, PCA finds principal components, the eigenarrays, which can be used to reduce the dimension of expression data for visualization, filtering of noise and for simplifying the subsequent computational analyses [2, 21].

Originally used in blind source separation (BSS) problems [19], ICA aims to find a transformation that decomposes an input dataset into components so that each component is statistically as independent from the others as possible. ICA has advantage over PCA because ICA exploits higher order statistics and has no restriction on its transformation, whereas PCA exploits only second order statistics and is restricted to orthogonal transformation. ICA has been successfully applied to analyze gene expression data to extract typical gene profiles for gene classification [13]. In [20], Liebermeister applied ICA to gene expression data to find independent modes of gene expression. However, few work has been done in applying ICA for sample clustering.

The remainder of this paper is organized as follows. Section 2 explains how to apply ICA to gene expression data analysis. Section 3 describes the sample clustering problem and a ICA-based method to solve the problem. Section 4 presents experimental results. And the concluding remarks are given in Section 5.

2 Independent Component Analysis for Gene Expression Data

Independent component analysis (ICA) [15, 16] is a relatively new statistical and computational technique that recovers a set of linearly mixed hidden independent factors from a set of measurements or observed data. Unlike principal component analysis (PCA) [18] which seeks for an orthogonal transformation to remove second-order statistical dependency, ICA not only de-correlates the second-order statistical mo-

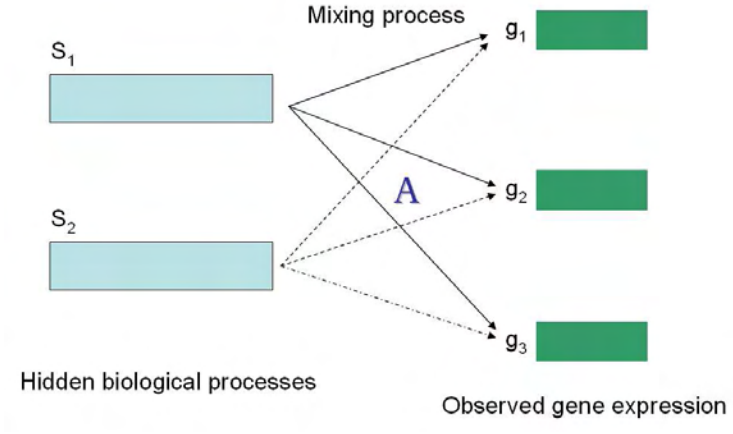


Figure 1: An example of ICA mixing model for gene expression.

ments, but also reduces higher-order statistical dependencies. A typical ICA model assumes that the source signals are not observable, statistically independent and non-Gaussian, with an unknown, but linear, mixing process. The latent variables are assumed non-Gaussian and mutually independent, and they are called the independent components of the observed data. These independent components (ICs), also called sources or factors, can be found by ICA.

To apply ICA to gene expression data analysis, the observed gene expression data of n genes, each of which are measured across m samples, can be represented as a matrix denoted as $\mathbb{X} = \{x_{i,j} | i = 1 \sim n, j = 1 \sim m\} (n \gg m)$, where $x_{i,j}$ corresponds to the expression value of gene g_i on the sample r_j. Each $x_{i,j}$ is a linear mixture of l ($l \leq m$) hidden and independent biological processes, and each process forms a vector $s_i (1 \leq i \leq l)$ representing levels of gene up-regulation or down-regulation. At each condition, the processes mix with different activation levels to determine the vector of observed gene expression levels measured by a microarray sample. These processes can be represented by an ICA mixing model

$$\mathbb{X}^T = \mathbb{A}\mathbb{S}^T \tag{1}$$

where the $n \times l$ matrix $\mathbb{S} = [s_1, s_2, ...s_l]$ contains l independent components corresponding to the independent biological processes, the $l \times m$ matrix $\mathbb{A} = [a_1, a_2, ...a_m]$ is a full ranked mixing matrix, and vectors $a_i (1 \leq i \leq m)$ are the basis vectors of ICA. Note that T represents the transpose of a matrix.

Since both $\mathbb{A}$ and $\mathbb{S}$ are unknown, and $s_i (1 \leq i \leq l)$ are statistically independent, ICA trys to estimate an de-mixing matrix W by an iteration approach such that $\mathbb{Y}$ is a good approximation to the original biological processes $\mathbb{S}$ based on the de-mxing model

$$\mathbb{Y}^T = \mathbb{W}\mathbb{X}^T \tag{2}$$

Where $\mathbb{W} = \mathbb{A}^{-1}$.

To illustrate the ICA mixing model, consider a case listed in Figure 1 where $n = 3$ (three genes), $m = 2$ (two samples), and $l = 2$ (two independent biological processes), then we have

$$\mathbb{X} = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \end{pmatrix}, \tag{3}$$

and

$$\mathbb{X}^T = \begin{pmatrix} x_{11} & x_{21} & x_{31} \\ x_{12} & x_{22} & x_{32} \end{pmatrix} \tag{4}$$

$$= \mathbb{A}\mathbb{S}^T = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \end{pmatrix}, \tag{5}$$

where

$$x_{11} = a_{11} * s_{11} + a_{12} * s_{21}, \qquad x_{12} = a_{21} * s_{11} + a_{22} * s_{21}, \tag{6}$$

$$x_{21} = a_{11} * s_{12} + a_{12} * s_{22}, \qquad x_{22} = a_{21} * s_{12} + a_{22} * s_{22}, \tag{7}$$

$$x_{31} = a_{11} * s_{13} + a_{12} * s_{23}, \qquad x_{32} = a_{21} * s_{13} + a_{22} * s_{23}. \tag{8}$$

Note that $\mathbb{S}$ contains l ($l \leq m$) independent components (ICs) which represents l independent biological processes, and each IC s_i is a vector of n controlling factor $s_i = (s_{i1}, ..., s_{ij}, ..., s_{in})$ where the jth controlling factor corresponds to the jth gene on the original expression data, and each controlling factor represents levels of gene up-regulation or down-regulation on that particular biological process. For the example given above, $\mathbb{S}$ contains three ICs. And from Equation 7 we can see that controlling factor s_{12} represents up-regulation or down-regulation of g_2 (gene No. 2) on biological process s_1, and controlling factor s_{22} represents up-regulation or down-regulation of g_2 (gene No. 2) on biological process s_2, respectively.

3 Problem and Methodology

The sample clustering problem can be stated as follows: given a gene expression matrix $\mathbb{X}$ of n genes, each of which are measured across m samples, the problem is how to find $\mathbb{K}$ mutually exclusive groups of the samples matching their empirical phenotypes, and to find the set of genes which manifests the empirical phenotype partitions.

To solve the above problem, we perform the following steps:

Step 1 - ICA-based Transformation. We apply ICA to $\mathbb{X}$ based on Equation1 and Equation2, and calculate $\mathbb{W}$ and $\mathbb{Y}$, hence $\mathbb{A}$ and $\mathbb{S}$.

Step 2 - Informative Genes Pickup. Based on the assumption that genes having relatively high or low controlling factor values within an IC are more informative than other genes for that biologic process, We hope these genes are also more informative in sample clustering.

For each s_i $(1 \leq i \leq l)$, we first sort the values of all controlling factor values s_{ip} $(1 \leq p \leq n)$, then calculate the informative gene set

$I_i = \{g_j | s_{ij}\ is\ among\ the\ highest\ k\%\ or\ lowest\ k\%\ of\ s_{ip}\ (1 \leq p \leq n)\}$,

where k $(0 < k < 100)$ is a threshold.

Since each gene g_j $(1 \leq j \leq n)$ could be informative for multiple ICs or biologic processes, we further calculate its gene informative scale gis_j by counting the number of occurrences of g_j in all I_i $(1 \leq i \leq l)$. Generally, the higher the gene informative scale value is, the more informative across all ICs (or biologic processes) the gene is.

Dataset	Original data size	Informative gene space size	Runtime (sec)
Leukemia-G1	7129 x 38	62 x 38	15.072
Leukemia-G2	7129 x 34	200 x 34	8.032
Colon Cancer	2000 x 62	80 x 62	13.099

Figure 2: The runtime of ICA-based informative gene selection on each dataset.

Dataset	Full gene space					Informative gene space (ICA)					Informative gene space (PCA)				
	K-means			HC	SOM	K-means			HC	SOM	K-means			HC	SOM
	Ave	Max	Min			Ave	Max	Min			Ave	Max	Min		
Leukemia-G1	0.5672	0.7653	0.4865	0.5562	0.5775	0.7880	0.9474	0.4865	0.5775	0.8506	0.5759	0.8506	0.4865	0.5775	0.4879
Leukemia-G2	0.5086	0.5989	0.4848	0.5009	0.5009	0.5490	0.6108	0.4923	0.5256	0.5722	0.4924	0.5294	0.4848	0.5012	0.5134
Colon Cancer	0.4967	0.6631	0.4918	0.5346	0.4939	0.7277	0.7478	0.4939	0.5346	0.7964	0.6002	0.8223	0.4918	0.5346	0.4966

Figure 3: The Rand Index values reached by applying different methods to different datasets.

Finally, we can form the informative gene set as follows:

$IGenes = \{g_j | gis_j >= t, 1 \leq j \leq n\}$,

where t $(0 < j < l)$ is a cut-off threshold.

Step 3 - Sample Clustering. By using different clustering methods such as K-means, hierarchical clustering (HC), and self-organizing maps (SOM), we can cluster the samples into $\mathbb{K}$ mutually exclusive groups based on the informative gene space instead of the original expression data (full gene space).

4 Experiments and Results

In this section, we will report performance evaluation of the proposed method on the following gene expression datasets:

- **The Leukemia Datasets–** The leukemia datasets are based on a collection of leukemia patient samples reported in [10]. It contains measurements corresponding to acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML) samples from bone marrow and peripheral blood (Therefore, $\mathbb{K} = 2$.). Two matrices are involved: one includes 38 samples (27 ALL vs. 11 AML, denoted as G1), and the other contains 34 samples (20 ALL vs. 14 AML, denoted as G2). Each sample is measured over 7129 genes.
- **Colon Cancer Dataset–** Colon adenocarcinoma specimens (snap-frozen in liquid nitrogen within 20 minutes of removal) were collected from patients. From some of these patients, paired normal colon tissue also was obtained. Cell lines used (EB and EB-1) have been described. RNA was extracted and hybridized to the array as described. Treatment of Raw Data from Affymetrix Oligonucleotide Arrays. The Affymetrix Hum6000 array contains about $65,000$ features, each containing '107 strands of a DNA 25-mer oligonucleotide. Sequences from about $3,200$ full-length human cDNAs and $3,400$ ESTs that have some similarity to other eukaryotic genes are represented on a set of four chips. The microarray dataset consists of 22 normal and 40 tumor colon tissue samples (Therefore, $\mathbb{K} = 2$.). It was reported by Alon et al. [1]. In this dataset, each sample has 2000 genes.

The ground-truth of the partition, which includes such information as how many samples belong to each class and the class label for each sample, is only used to evaluate the experimental results.

The *Rand Index* [22] between the ground-truth of phenotype structure P of the samples and the clustering result Q of an algorithm has been adopted to for the effectiveness evaluation. Let $\boldsymbol{a}$ represent the

number of pairs of samples that are in the same cluster in P and in the same cluster in Q, ***b*** represent the number of pairs of samples that are in the same cluster in P but not in the same cluster in Q, ***c*** be the number of pairs of samples that are in the same cluster in Q but not in the same cluster in P, and ***d*** be the number of pairs of samples that are in different clusters in P and in different clusters in Q. The *Rand Index* [22] is calculated as

$$RI = \frac{a+d}{a+b+c+d}.$$

The *Rand Index* lies between 0 and 1. Higher values of the *Rand Index* indicate better performance of the algorithm.

During the experiment, each dataset has three different gene spaces:

- Full gene space, which is the original dataset.
- ICA-based informative gene space, which consists of informative genes generated by the proposed ICA-based informative gene pickup method. For each dataset, to generate corresponding informative gene sets, we have applied FastICA [14] to estimate the independent components in parallel (using tanh nonlinearity in symmetric estimation mode), and the number of independent components (l) equals the dimension of samples in that dataset (m), which means we have kept all the ICs. Different threshold values of k and t have been used to control the generation of different informative gene sets.
- PCA-based informative gene space, which is the result of PCA-based dimension reduction directly applied to the gene dimension. Each PCA-based informative gene space has a corresponding ICA-based informative gene space where both have the same number of genes.

Figure 2 reports the sizes of the original datasets and their corresponding ICA-based informative gene spaces, as well as the runtime (in seconds) to obtain informative genes. For Leukemia Dataset G1, 62 out of 7129 genes have been selected as informative genes by using ICA-based method with the following experimental parameters: $l = 38$, $k = 0.1$ and $t = 17$. For Leukemia Dataset G2, 200 out of 7129 genes have been selected as informative genes based on parameters $l = 34$, $k = 0.1$ and $t = 10$. For Colon Cancer Dataset, 80 out of 2000 genes have been picked up as informative genes based on parameters $l = 62$, $k = 0.1$ and $t = 10$. The above operations, which only took seconds of time, were conducted using Matlab on a HP desktop PC with P4 2.8 GHz CPU and 512 MB main memory. Since FastICA [14] converges very quickly, efficiency is not a major concern for this experiment.

For three gene spaces on each dataset, first we have applied some unsupervised clustering methods such as K-means, hierarchical clustering (HC), and self-organizing maps (SOM), then we have calculated *Rand Index* for each experimental case. The similarity measurement is *correlation coefficient* [8]. Before applying clustering algorithms, data normalization has been performed as a preprocess step based on the following formula:

$$w'_{i,j} = \frac{w_{i,j} - \overline{w}_i}{\sigma_i},$$

where $w'_{i,j}$ denotes the normalized value for gene i of sample j, $w_{i,j}$ represents the original value, $\overline{w}_i$ is the mean of the values for gene i over all samples, and σ_i is the standard deviation of the i^{th} gene.

Figure 3 provides some experimental results. Note that since the clustering results of K-means are not stable due to random initialization, we have applied the algorithm 100 times on all three gene spaces, and listed the average, maximum and minimum values of *Rand Index*. For Leukemia Dataset G1, K-means algorithm has achieved better performance on the ICA-based informative gene space, which was indicated by the higher average and maximum values of *Rand Index*. Similarly, SOM has much better performance on the ICA-based informative gene space, while hierarchical clustering's performance only

improved slightly. After PCA-based dimension reduction applied to G1 to get 62 informative genes, K-means and hierarchical clustering algorithm have achieved slightly better performance than on the full gene space, but SOM's performance deteriorated.

Similar results have been obtained on other two datasets: Leukemia Dataset G2 and Colon Cancer Dataset. Generally, SOM has gained the largest performance improvement on the ICA-based informative gene space, and K-menas has ranked the second. It seems that hierarchical clustering only had slightly better performance because it is not sensitive to the new informative gene space generated by ICA-based method. Interestingly, three clustering algorithms have performed differently on the PCA-based informative gene space. K-means and hierarchical clustering algorithms have achieved slightly better or similar performance compared with the cases on the full gene space, but SOM has achieved lower performance. In addition, performances on PCA-based informative gene space and the full gene space has been consistently lower than those on ICA-based informative gene space, which illustrated the effectiveness of the ICA-based informative gene pickup method. Therefore, in overall, all three clustering algorithms have achieved best performance on the ICA-based informative gene space.

5 Conclusion and Future Work

In this paper, we have described the problem of sample clustering on high gene dimension datasets. We also have introduced a method utilizing independent component analysis (ICA) to select informative genes. Various clustering algorithms have achieved higher performance based on the new and reduced informative gene space. Currently we are improving the quality of informative genes by conducting ICA-based investigation more thoroughly. We will also apply more sophisticated clustering algorithms on the informative gene space to verify its effectiveness. One of the future work is to give out the biological explanations underlying independent component analysis based on different datasets.

References

[1] Alon U., Barkai N., Notterman D.A., Gish K., Ybarra S., Mack D. and Levine A.J. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide array. *Proc. Natl. Acad. Sci. USA*, Vol. 96(12):6745–6750, June 1999.

[2] Alter O., Brown P.O. and Bostein D. Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl. Acad. Sci. USA*, Vol. 97(18):10101–10106, Auguest 2000.

[3] Azuaje, Francisco. Making Genome Expression Data Meaningful: Prediction and Discovery of Classes of Cancer Through a Connectionist Learning Approach. In *Proceedings of IEEE International Symposium on Bioinformatics and Biomedical Engineering*, pages 208–213, 2000.

[4] Ben-Dor A., Shamir R. and Yakhini Z. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.

[5] Brazma, Alvis and Vilo, Jaak. Minireview: Gene expression data analysis. *Federation of European Biochemical societies*, 480:17–24, June 2000.

[6] Chen J.J., Wu R., Yang P.C., Huang J.Y., Sher Y.P., Han M.H., Kao W.C., Lee P.J., Chiu T.F., Chang F., Chu Y.W., Wu C.W. and Peck K. Profi ling expression patterns and isolating differentially expressed genes by cDNA microarray system with colorimetry detection. *Genomics*, 51:313–324, 1998.

[7] DeRisi J., Penland L., Brown P.O., Bittner M.L., Meltzer P.S., Ray M., Chen Y., Su Y.A. and Trent J.M. Use of a cDNA microarray to analyse gene expression patterns in human cancer. *Nature Genetics*, 14:457–460, 1996.

[8] Devore, Jay L. *Probability and Statistics for Engineering and Sciences*. Brook/Cole Publishing Company, 1991.

[9] Eisen M.B., Spellman P.T., Brown P.O. and Botstein D. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, Vol. 95:14863–14868, 1998.

[10] Golub T.R., Slonim D.K. *et al.* Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, Vol. 286(15):531–537, October 1999.

[11] Golub T.R., Slonim D.K., Tamayo P., Huard C., Gassenbeek M., Mesirov J.P., Coller H., Loh M.L., Downing J.R., Caligiuri M.A., Bloomfield D.D. and Lander E.S. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, Vol. 286(15):531–537, October 1999.

[12] Hastie T., Tibshirani R., Boststein D. and Brown P. Supervised harvesting of expression trees. *Genome Biology*, Vol. 2(1):0003.1–0003.12, January 2001.

[13] Hori,G., Inoue,M., Nishimura,S. and Nakahara,H. Blind gene classification based on ica of microarray data. In *Proc. 3rd Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2001)*, pages 332–336, SanDiego, California, USA, 2001.

[14] A. Hyvärinen. Hut - cis : The fastica package for matlab.

[15] A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[16] A. Hyvärinen and E. Oja. independent component analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

[17] Iyer V.R., Eisen M.B., Ross D.T., Schuler G., Moore T., Lee J.C.F., Trent J.M., Staudt L.M., Hudson Jr. J., Boguski M.S., Lashkari D., Shalon D., Botstein D. and Brown P.O. The transcriptional program in the response of human fibroblasts to serum. *Science*, 283:83–87, 1999.

[18] I. Jollie. *Principal Component Analysis*. Springer-Verlag, 1986.

[19] Jutten C. and Herault J. Blind separation of sources, part i: an adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.

[20] Liebermeister,W. Linear modes of gene expression determined by independent component analysis. *Bioinformatics*, 18:51–60, 2002.

[21] Misra, J., Schmitt, W., Hwang, D., Hsiao, L., Gullans. S., Stephanopoulos, G., and Stephanopoulos, G. Interactive exploration of microarray gene expression patterns in a reduced dimensional space. *Genome Res 2002*, 12:1112–1120, 2002.

[22] Rand, W.M. Objective criteria for evaluation of clustering methods. *Journal of the American Statistical Association*, 1971.

[23] Schena M., Shalon D., Heller R., Chai A., Brown P.O., and Davis R.W. Parallel human genome analysis: Microarray-based expression monitoring of 1000 genes. *Proc. Natl. Acad. Sci. USA*, Vol. 93(20):10614–10619, October 1996.

[24] Shalon D., Smith S.J. and Brown P.O. A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization. *Genome Research*, 6:639–645, 1996.

[25] Slonim D.K., Tamayo P., Mesirov J.P., Golub T.R. and Lander E.S. Class Prediction and Discovery Using Gene Expression Data. In *RECOMB 2000: Proceedings of the Fifth Annual International Conference on Computational Biology*. ACM Press, 2000.

[26] Tang, C., Zhang, A., and Ramanathan, M. ESPD: A Pattern Detection Model Underlying Gene Expression Profiles. *Bioinformatics*, 20(6):829–838, 2004.

[27] Welford S.M., Gregg J., Chen E., Garrison D., Sorensen P.H., Denny C.T. and Nelson S.F. Detection of differentially expressed genes in primary tumor tissues using representational differences analysis coupled to microarray hybridization. *Nucleic Acids Research*, 26:3059–3065, 1998.

[28] Xing E.P. and Karp R.M. Cliff: Clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics*, Vol. 17(1):306–315, 2001.

A Feature Generation Algorithm for Sequences with Application to Splice-Site Prediction

Rezarta Islamaj[1], Lise Getoor[1], and W. John Wilbur[2]

[1] Computer Science Department, University of Maryland, College Park, MD 20742
[2] National Center for Biotechnology Information, NLM, NIH, Bethesda, MD 20894
{rezarta, getoor}@cs.umd.edu, wilbur@ncbi.nlm.nih.gov

Abstract. In this paper we present a new approach to feature selection for sequence data. We identify general feature categories and give construction algorithms for them. We show how they can be integrated in a system that tightly couples feature construction and feature selection. This integrated process, which we refer to as *feature generation*, allows us to systematically search a large space of potential features. We demonstrate the effectiveness of our approach for an important component of the gene finding problem, splice-site prediction. We show that predictive models built using our feature generation algorithm achieve a significant improvement in accuracy over existing, state-of-the-art approaches.

Keywords: feature generation, splice-site prediction.

1 Introduction

Many real-world data mining problems involve data modeled as sequences. Sequence data comes in many forms including: 1) human communication such as speech, handwriting and language, 2) time sequences and sensor readings such as stock market prices, temperature readings and web-click streams and 3) biological sequences such as DNA, RNA and protein. In all these domains it is important to efficiently identify useful 'signals' in the data that enable the correct construction of classification algorithms.

Extracting and interpreting these 'signals' is known to be a hard problem. The focus of this paper is on a systematic and scalable method for feature generation for sequences. We identify a collection of generic sequence feature types and describe the corresponding feature construction methods. These methods can be used to create more complex feature representations. As exhaustive search of this large space of potential features is intractable, we propose a general-purpose, focused **feature generation algorithm (FGA)**, which integrates feature construction and feature selection. The output of the feature generation algorithm is a moderately sized set of features which can be used by arbitrary classification algorithm to build a classifier for sequence prediction.

We validate our method on the task of splice-site prediction for pre-mRNA sequences. Splice sites are locations in the DNA sequence which are boundaries for protein coding regions and non-coding regions. Accurate prediction of splice sites is an important component of the gene finding problem. It is a particularly difficult problem since the sequence characteristics, e.g. pre-mRNA sequence length, coding sequence length, number of interrupting intron sequences and their lengths, do not follow any known pattern, making it hard to locate the genes. The gene finding challenge is to build a general approach that, despite the lack of known patterns, will automatically select the right features to combine.

We demonstrate the effectiveness of this approach by comparing it with a state-of-the-art method, GeneSplicer. Our predictive models show significant improvement in accuracy. Our final feature set, which includes a mix of feature types, achieves a 4.4% improvement in the 11-point average precision when compared to GeneSplicer. At the 95% sensitivity level, our method yields a 10% improvement in specificity.

Our contribution is two-fold. First, we give a general feature generation framework appropriate for any sequence data problem. Second, we provide new results for splice-site prediction that should be of great interest to the gene-finding community.

2 Related Work

Feature selection techniques have been studied extensively in text categorization[1–5]. Recently they have begun receiving more attention for applications to biological data. Liu and Wong [6] give a good introduction for filtering methods used for the prediction of translation initiation sites. Degroves et al. [7] describe a wrapper approach which uses both SVMs and Naive Bayes to select the relevant features for splice sites. Other recent work includes models based on maximum entropy [8], in which only a small neighborhood around the splice site is considered. Zhang et al. [9] propose a recursive feature elimination approach using SVM and Saeys et al. have also proposed a number of different models [10, 11]. Finally, SpliceMachine [12] is the latest addition with compelling results for splice-site prediction.

In addition, there is a significant amount of work on splice-site prediction. One of the most well-known approaches is GeneSplicer proposed by Pertea et al [13]. It combines Maximal Dependency Decomposition (MDD) [14] with second order Markov models. GeneSplicer is trained on splice-site sequences 162 nucleotides long. This splice neighborhood is larger than most other splice-site programs [15]. GeneSplicer, similar to most other programs, assumes that splice sites follow the AG/GT nucleotide-pair consensus for acceptor and donor sites respectively. It uses a rich set of features including position-specific nucleotides and upstream/downstream trinucleotides.

3 Data Description

We validate our methods on a dataset which contains $4,000$ RefSeq[3] pre-mRNA sequences. Each sequence contains a whole human gene with $1,000$ additional nucleotides before and after the annotated start and stop locations of the gene. The base alphabet is $\{a, c, g, t\}$. The sequences have a non-uniform length distribution ranging from $2,359$ nucleotides to $505,025$ nucleotides. In a pre-mRNA sequence, a human gene is a protein coding sequence which is characteristically interrupted by non-coding regions, called introns. The coding regions are called exons and the number of exons per gene in our dataset varies non-uniformly between 1 and 48. The acceptor splice site marks the start of an exon and the donor splice site marks the end of an exon. All the pre-mRNA sequences in our dataset follow the AG consensus for acceptors and GT consensus for donors.

We extract acceptor sites from these sequences. Following the GeneSplicer format, we mark the splice site and take a subsequence consisting of 80 nucleotides upstream from the site and 80 nucleotides downstream. We extract negative examples by choosing random AG-pair locations that are not acceptor sites and selecting subsequences as we do for the true acceptor sites. Our data contains 20,996 positive instances and 200,000 negative instances.

4 Feature Generation

In this section we present a number of feature types for splice-site prediction and their corresponding construction procedures. If applied naively, the construction procedures produce feature sets, which become easily intractable. To keep the number of features at manageable levels, we then propose a general purpose feature generation algorithm which integrates feature construction and selection in order to produce meaningful features.

4.1 Feature Types and Construction Procedures

The feature types that we consider capture compositional and positional properties of sequences. These apply to sequence data in general and the splice-site sequence prediction problem in particular. For each feature type we describe an incremental feature construction procedure. The feature construction starts with an initial set of features and produces the constructed set of features. Incrementally, during each iteration, it produces richer, more complex features for each level of the output feature set.

[3] http://www.ncbi.nlm.nih.gov/RefSeq/

Compositional features A k-mer is a string of k-characters.We consider the general k-mer composition of sequences for k values $2, 3, 4, 5$ and 6. Given the alphabet for DNA sequences, $\{a, c, g, t\}$, the number of distinct features is 4^k for each value of k. There is a total of $5,456$ features for the k values we consider.

Construction Method. This construction method starts with an initial set of k-mer features and extends them to a set of $(k+1)$-mers by appending the letters of the alphabet to each k-mer feature. As an example, suppose an initial set of 2-mers $F_{initial} = \{ac, cg\}$. We construct the extended set of 3-mers $F_{constructed} = \{aca, acc, acg, act, cga, cgc, cgg, cgt\}$. Incrementally, in this manner we can construct levels $4, 5$ and 6.

Region-specific compositional features Splice-site sequences characteristically have a coding region and a non-coding region. For the acceptor splice-site sequences, the region of the sequence on the left of the splice-site position (upstream) is the non-coding region, and the region of the sequence from the splice-site position to the end of sequence (downstream) is the coding region. It is expected that these regions exhibit different compositional properties. In order to capture these differences we use *region-specific k-mers*. Here we also consider k-mer features for k values $2, 3, 4, 5$ and 6. Thus the total number of features is $10,912$.

Construction Method. The construction procedure of upstream and downstream k-mer features is the same as the general k-mer method, with the addition of region indication.

Positional features Position-specific nucleotides are the most common features used for finding signals in the DNA stream data [14–16]. These features capture the correlation between different nucleotides and their relative positions. Our sequences have a length of 160 nucleotides, therefore our basic position-specific feature set contains 640 features.

In addition, we want to capture the correlations that exist between different nucleotides in different positions in the sequence. Several studies have proposed *position-specific k-mers*, but this feature captures only the correlations among nearby positions. Here we propose a *conjunctive position-specific feature*. We construct these complex features from conjunctions of basic position-specific features. The dimensionality of this kind of feature is inherently high.

Construction Method. We start with an initial conjunction of basic features and add another conjunct basic feature in an unconstrained position. Let our basic set be $F_{basic} = \{a_1, c_1, ..., g_n, t_n\}$, where, for example, a_1 denotes nucleotide *a* at the first sequence position. Now, if our initial set is $F_{initial} = \{a_1, g_1\}$, we can extend it to the level 2 set of position-specific base combinations $F_{constructed} = \{a_1 \wedge a_2, a_1 \wedge c_2, ..., g_2 \wedge t_n\}$. Incrementally, in this manner we can construct higher levels. For each iteration, if the number of conjuncts is k we have a total of $\binom{n}{k} \times 4^k$ such features for a sequence of length n.

4.2 Feature Selection

Feature selection methods reduce the set of features by keeping only the useful features for the task at hand. The problem of selecting useful features has been the focus of extensive research and many approaches have been proposed [1–3, 5, 17]. In our experiments we consider several feature selection methods to reduce the size of our feature sets, including *Information Gain (IG), Chi-Square (CHI), Mutual Information (MI)* [18] and *KL-distance (KL)* [2]. Due to space limitations, in the experiments section, we present the combination that produced the best results. We used Mutual Information to select compositional features and Information Gain to select positional features during our feature generation step.

4.3 Feature Generation Algorithm (FGA)

The traditional feature selection approaches consider a single brute force selection over a large set of all features of all different types. We emphasize a type-oriented feature selection approach. The type-oriented approach introduces the possibility of employing different feature selection models for each type set; i.e.

for a feature set whose dimensionality is not too high we may use a wrapper approach [1] in the selection step, while for a large feature type set we may use filter approaches [3]. Also, in this manner features of different types can be generated in a parallel fashion. In order to employ the information embedded in the selected features for sequence prediction, we propose the following algorithm:

- *Feature Generation.* The first stage generates the feature sets for each feature type. We start with several defined feature types. For each feature type, we tightly couple together a feature construction step and a feature selection step and, iterating through these steps, we generate richer and more complex features. We specify a feature selection method for each feature type and thus, during each iteration, eliminate a subset of features that are obtained from the construction method. These features are usually assigned a low selection score and their elimination will not affect the performance of the classification algorithm.
- *Feature Collection and Selection.* In the next stage, we collect all the generated features of different types and apply another selection step. This selection step is performed because features of a particular type may be more important for the sequence prediction. We produce a set of features originating from different feature types and different selection procedures.
- *Classification.* The last stage of our algorithm builds a classifier over the refined set of features and learns a model for the given dataset.

In addition to being computationally tractable, this feature generation approach has other advantages such as the flexibility to adapt with respect to the feature type and the possibility to incorporate the module in a generic learning algorithm.

5 Experimental Results for Splice Site Prediction

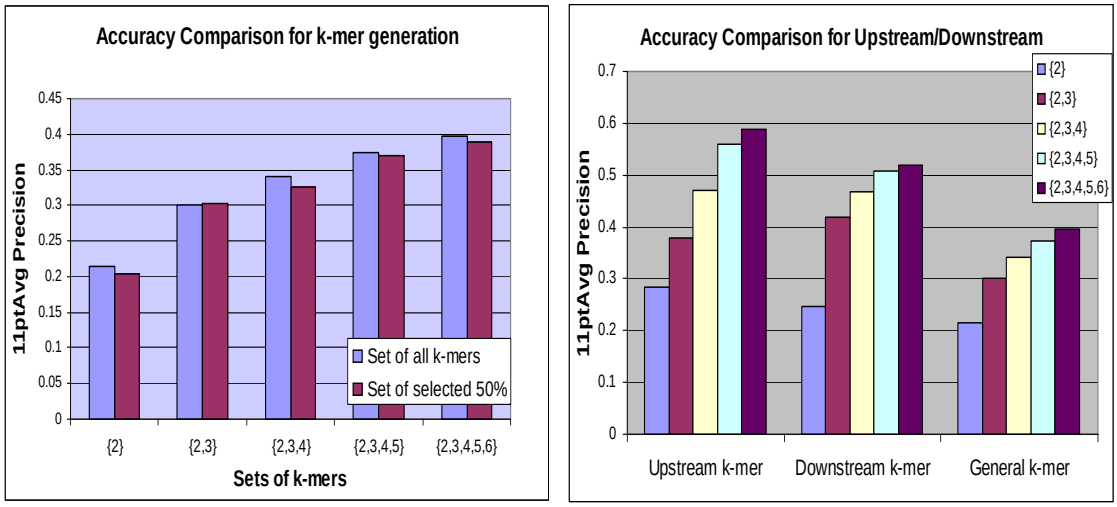
We conducted a wide range of experiments to support our claims, and here we present a summary of them. For our experiments, we considered a range of classifiers. We present results for the classifier that consistently gave the best results, called C-Modified Least Squares (CMLS) [19]. CMLS is a wide margin classifier related to Support Vector Machines (SVM), but has a smoother penalty function. This allows the calculation of gradients which can provide faster convergence.

5.1 Performance Measures

We use the *11-point average* measure[20] to evaluate the performance of our algorithm. To calculate this measure, we rank the test data in decreasing order of scores. For a threshold *t*, the test data points above the threshold are the sequences *retrieved*. Of these, those that are true positives (TP) are considered *relevant*. *Recall* is the ratio of relevant sequences retrieved to all relevant sequences (including those missed) and *precision* is the ratio of relevant sequences retrieved to all retrieved sequences. For any recall ratio, we calculate the precision at the threshold which achieves that recall ratio and compute the average precision. The 11-point average precision (11ptAVG) is the average of precisions estimated at the 11 recall values 0%, 10%, 20%, ., 100%. At each such recall value, the precision is estimated as the highest precision occurring at any rank cutoff where the recall is at least as great as that value.

The measures of *sensitivity (Se)* and *specificity (Sp)* commonly used by the computational biology community correspond respectively to the recall and precision definitions. Another performance measure commonly used for biological data is the *false positive rate(FPr)* defined as $FPr = \left(\frac{FP}{FP+TN}\right)$ where FP, and TN are the number of false positives and true negatives respectively. By varying the decision threshold of the classifier FP can be computed for all recall values. We also present results using this measure.

In all our experiments, the results reported use three-fold cross-validation.



(a) (b)

Fig. 1: (a) 11ptAVG precision results of the different collection sets of k-mers with no selection *(Sets of {2},...,{2,3,4,5,6}-mers)* and 50% of the features after using mutual information for selection *(Sets of selected 50%)* (b) Comparison between different feature type sets performances, upstream k-mers, downstream k-mers, and general k-mers shown in sets of {2},...,{2,3,4,5,6}-mers.

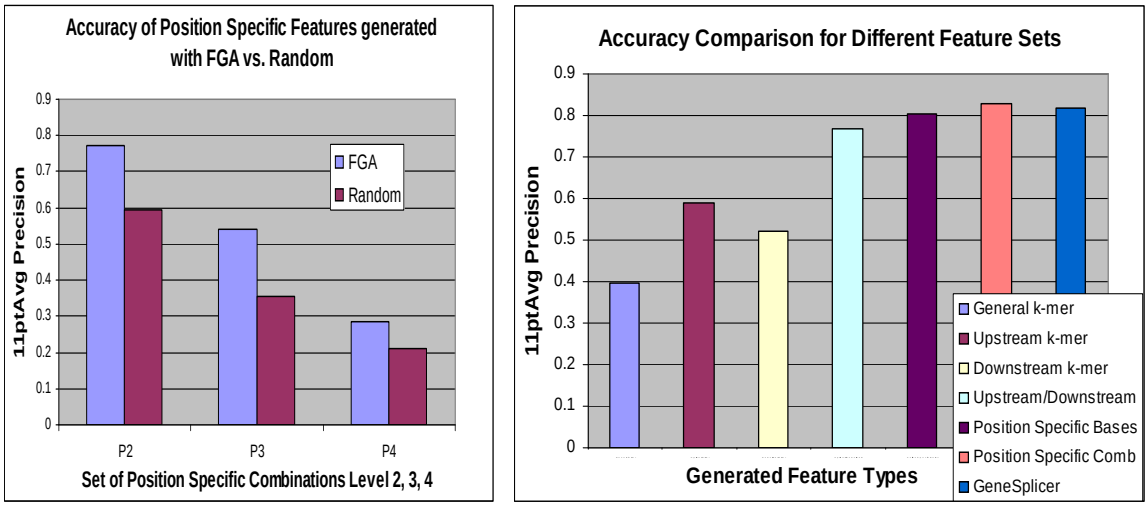
5.2 Accuracy Results of FGA

In the following experiments we present the evaluation of four different feature types, which carry positional and compositional information. As discussed in Section 4.3 initially we evaluate them separately and then identify the best group of features for each type before combining them.

Compositional features and splice-site prediction We examine each k-mer feature set independently for each value of k. We use the whole k-mer set to construct the new $(k+1)$-mer set. In our experiments, we found the *MI* selection method works best for compositional features. Figure 1(a) shows the accuracy results for the general k-mer features as we collect them through each iteration. Note that reducing the number of features in half has little effect on the overall performance. In Figure 1(b) we highlight the contribution of the region-specific k-mer features at each iteration. It is clear that k-mer features carry more information when associated with a specific region (upstream or downstream) and this is shown by the significant increase in their 11ptAVG precisions. We combine upstream and downstream k-mer features and summarize the results in Figure 2(b) along with the individual performances of each feature type. These features show an 11ptAVG precision of 77.18%, as compared to 39.84% of general k-mers.

Next, we collect the generated compositional features in the *feature collection and selection stage* of our algorithm. During this step, we pick $2,000$ compositional features of different types without affecting the performance of the classification algorithm. From this final set we observe that, in general, higher level k-mers are more informative for splice-site prediction. Furthermore, we find that the generated final k-mer feature set reveals more 5-mers and 6-mers originating from the downstream (coding) region. This is to be expected since these features can capture the compositional properties of the coding region.

Positional features and splice-site prediction Position-specific nucleotides, which constitute our basic feature set F_{basic}, give a satisfactory 11ptAvg precision, 80.34%. This is included in the graph in Figure 2(b). An initial observation of the conjunctive position-specific features reveals that, for pair-wise combinations, we have over $200,000$ unique pairs and for combinations of triples this number exceeds 40 million. Using our feature generation algorithm, we generate higher levels of this feature type, starting with the basic position-specific nucleotide features. For each conjunct level we use the construction



(a) (b)

Fig. 2: a) 11ptAvg Precision results for the position specific feature sets generated with FGA algorithm vs randomly generated features. b) Performance results of the FGA method for different feature types as well as the GeneSplicer program

method to get the next level of features. We use the *IG* selection method to select the top scoring $1,000$ features and repeat the generation to get the next level using the selected set of features as the initial set. We explore from one to four conjuncts denoted as $(P1, P2, P3, P4)$.

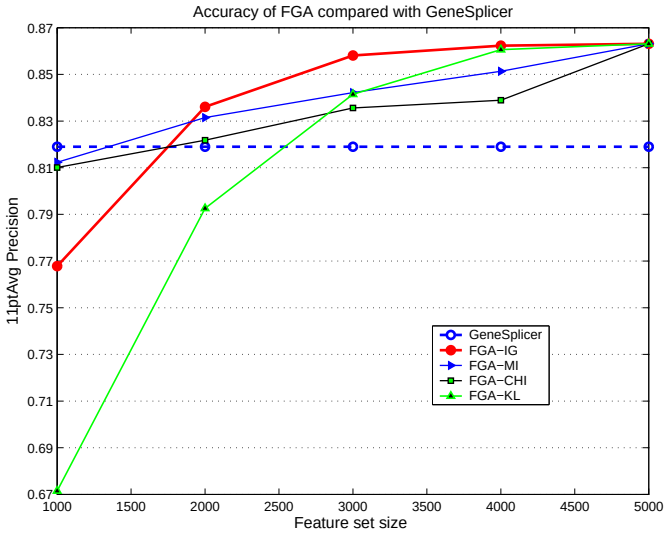
In Figure 2(a), we show the performances of the conjunctive feature sets $P2$, $P3$, and $P4$. For comparison, we introduce a baseline method, which randomly picks $1,000$ conjunctive features from each level of two, three and four conjuncts. We randomly generate 10 rounds of such feature sets from each level and we compute the average performance for the level. We compare our feature generation algorithm against this random generation baseline. As we can see from the figure, FGA outperforms random selection significantly.

In the *feature collection and selection step*, we combine the FGA generated features that carry positional information. Without any loss in 11ptAVG precision we select the top $3,000$ features of this collection. The 11ptAvg precision that this collection set gives for the acceptor splice-site prediction is 82.67% as summarized in Figure 2(b). These results clearly show that using more complex position-specific features is beneficial. In particular, we observe that pairs of position-specific bases, i.e. level 2 features, are a very important feature set that should be exploited. Interestingly, typically they are not considered by existing splice-site prediction algorithms. Figure 2(b) also shows the performance of GeneSplicer on the same dataset. We see that our positional features combination performs better than GeneSplicer.

The final collection and comparison with GeneSplicer In the following set of experiments, we show the results after we collect the features of all types that we have generated. We run our CMLS classification algorithm with a feature set of size $5,000$ containing general k-mers, upstream/downstream k-mers, position-specific nucleotides and conjunctions of position-specific features. We achieve an 11ptAVG precision performance of 86.31%. This compares quite favorably with one of the leading programs in splice-site prediction, GeneSplicer, which yields an accuracy of 81.89% on the same dataset. The precision results at all individual recall points are shown in Figure 3(a). As it can be seen from the figure, our precision results are consistently higher than those of GeneSplicer at all 11 recall points. For these experiments, in Figure 3(b), we have included the results of repeated selection for *IG, MI, CHI* and *KL* feature selection methods. Since the collection stage of our algorithm allows for several feature selection steps, we explore more aggressive feature selection options and see that smaller feature sets of even $2,000$ also

Recall	Precision(Specificity)	
(Sensitivity)	FGA	GeneSplicer
0	1	1
0.1	0.984	0.967
0.2	0.976	0.960
0.3	0.967	0.941
0.4	0.957	0.926
0.5	0.946	0.907
0.6	0.932	0.880
0.7	0.906	0.849
0.8	0.865	0.792
0.9	0.787	0.689
1	0.174	0.095
11ptAVG	0.863	0.819

(a)



(b)

Fig. 3: (a) The precision values for FGA and GeneSplicer at 11 recall points (b) 11ptAverage precision results for FGA varying the feature set size, compared to GeneSplicer

outperform GeneSplicer. Of these, we prefer the *IG* selection method since it retains the high precision of greater than 86% and in such problems of biological nature a higher specificity is very important.

In order to give further details on the difference between the performances of the two programs we present the false positive rates for various sensitivity values in Figure 4. Our feature generation algorithm, with its rich set of features, consistently performs better than GeneSplicer. Our false positive rates are favorably lower at all recall values. At a 95% sensitivity rate the FPr decreased from 6.2 to 4.3%. This is a significant reduction in false positive predictions. This can have a great impact when splice-site prediction is incorporated into a gene-finding program.

6 Conclusions

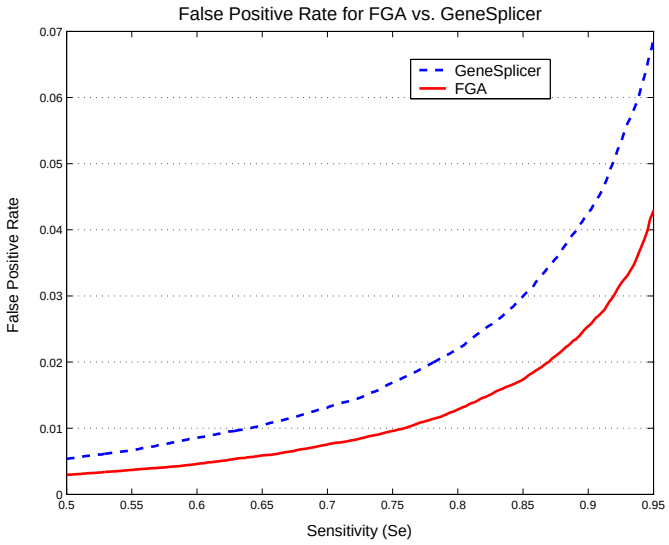
We presented a general feature generation framework, which integrates feature construction and feature selection in a flexible manner. We showed how this method could be used to build accurate sequence classifiers. We presented experimental results for the problem of splice-site prediction. We were able to search over an extremely large space of feature sets effectively, and we were able to identify the most useful set of features of each type. By using this mix of feature types, and searching over combinations of them, we were able to build a classifier which achieves an accuracy improvement of 4.4% over an existing state-of-the-art splice-site prediction algorithm. The specificity values are consistently higher for all sensitivity thresholds and the false positive rate has favorably decreased. In future work, we plan to apply our feature generation algorithm to more complex feature types and other sequence prediction tasks, such as translation start site prediction.

References

1. Kohavi, R., John, G.: The wrapper approach. In: *Feature Extraction, Construction and Selection : A Data Mining Perspective*, Liu,H.,Motoda,H.,eds. Kluwer Academic Publishers (1998)
2. Koller, D., Sahami, M.: *Toward optimal feature selection*. In: ICML. (1996) 284–292
3. Yang, Y., Pedersen, J.: *A comparative study on feature selection in text categorization*. In: ICML. (1997)

Sensitivity	FPr (False Positive Rate)	
(Recall)	FGA	GeneSplicer
0.97	0.059	0.092
0.95	0.043	0.062
0.93	0.033	0.056
0.92	0.030	0.051
0.90	0.025	0.043
0.87	0.020	0.035
0.85	0.017	0.030
0.82	0.015	0.025
0.80	0.012	0.022
0.75	0.009	0.017
0.70	0.007	0.013
0.65	0.010	0.006

(a)



(b)

Fig. 4: (a) The false positive ratio values for FGA and GeneSplicer at various sensitivity thresholds (b) The false positive rate results for FGA varying the sensitivity threshold, compared to GeneSplicer

4. Yu, L., Liu, H.: *Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution*. In: ICML. (2003)
5. Blum, A., Langley, P.: *Selection of relevant features and examples in machine learning*. Artificial Intelligence (1997)
6. Liu, H., Wong, L.: *Data mining tools for biological sequences*. Journal of Bioinformatics and Computational Biology (2003)
7. Degroeve, S., Baets, B., de Peer, Y.V., Rouze, P.: *Feature subset selection for splice site prediction*. In: ECCB. (2002) 75–83
8. Yeo, G., Burge, C.: *Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals*. In: RECOMB. (2003)
9. Zhang, X., Heller, K., Hefter, I., Leslie, C., Chasin, L.: *Sequence information for the splicing of human pre-mRNA identified by support vector machine classification*. Genome Research **13** (2003) 2637–2650
10. Saeys, Y.: *Feature selection for classification of nucleic acid sequences*. PhD thesis, Ghent U., Belgium (2004)
11. Saeys, Y., Degroeve, S., Aeyels, D., de Peer, Y.V., Rouze, P.: *Fast feature selection using a simple estimation of distribution algorithm: a case study on splice site prediction*. Bioinformatics **19** (2003) ii179–ii188
12. Degroeve, S., Saeys, Y., Baets, B.D., Rouz, P., de Peer, Y.V.: *SpliceMachine: predicting splice sites from high-dimensional local context representations*. Bioinformatics **21** (2005) 1332–1338
13. Pertea, M., Lin, X., Salzberg, S.: *GeneSplicer: a new computational method for splice site prediction*. Nucleic Acids Research **29** (2001) 1185–1190
14. Burge, C., Karlin, S.: *Prediction of complete gene structures in human genomic DNA*. Journal of Molecular Biology (1997) 78–94
15. Kim, W., Wilbur, W.: *DNA Splice Site Detection: A Comparison of Specific and General Methods*. In: AMIA. (2002)
16. Zhang, M.: *Statistical features of human exons and their flanking regions*. Human Molecular Genetics **7** (1998) 919–932
17. Brank, J., Grobelnik, M., Frayling, N.M., Mladenic, D.: *Interaction of feature selection methods and linear classification model*. In: Workshop on Text Learning. (2002)
18. Mitchell, T.: *Machine Learning*. The Mc-Graw-Hill Companies, Inc. (1997)
19. Zhang, T., Oles, F.: *Text categorization based on regularized linear classification methods*. Information Retrieval **4** (2001) 5–31
20. Witten, I., Moffat, A., Bell, T., eds.: *Managing Gigabytes*. 2 edn. Van Nostrand Reinhold (1999)

FEATURE SELECTION CONSIDERING ATTRIBUTE INTER-DEPENDENCIES

Manuel Mejía-Lavalle, Eduardo F. Morales[1]
Instituto de Investigaciones Eléctricas, Reforma 113, 62490 Cuernavaca, Morelos, México
[1] INAOE, L.E.Erro 1, 72840 StMa. Tonantzintla, Puebla, México
mlavallle@iie.org.mx, emorales@inaoep.mx

Abstract. With the increasing size of databases, feature selection has become a relevant and challenging problem for the area of knowledge discovery in databases. An effective feature selection strategy can significantly reduce the data mining processing time, improve the predicted accuracy, and help to understand the induced models, as they tend to be smaller and make more sense to the user. Many feature selection algorithms assumed that the attributes are independent between each other given the class, which can produce models with redundant attributes and/or exclude sets of attributes that are relevant when considered together. In this paper, an effective best first search algorithm, called buBF, for feature selection is described. buBF uses a novel heuristic function based on *n-way* entropy to capture inter-dependencies among variables. It is shown that buBF produces more accurate models than other state-of-the-art feature selection algorithms when compared on several synthetic and real datasets.

Keywords: Data mining, Feature selection, *n-way* entropy.

1 Introduction

Data mining is mainly applied to large amounts of stored data to look for the implicit knowledge hidden within this information. To take advantage of the enormous amount of information currently available in many databases, algorithms and tools specialized in the automatic discovery of hidden knowledge within this information have been developed. This process of non-trivial extraction of relevant information that is implicit in the data is known as Knowledge Discovery in Databases (KDD), in which the data mining phase plays a central role in this process.

It has been noted, however, that when very large databases are going to get mined, the mining algorithms get very slow, requiring too much time to process the information. One way to approach this problem is to reduce the amount of data before applying the mining process. In particular, the pre-processing method of feature selection, applied to the data before mining, has been shown to be promising because it can eliminate the irrelevant or redundant attributes that cause the mining tools to become inefficient and ineffective. At the same time, it can preserve-increase the classification quality of the mining algorithm (accuracy) [1].

Although there are many feature selection algorithms reported in the specialized literature, none of them are perfect: some of them are effective, but very costly in computational time (e.g., wrappers methods), and others are fast, but less effective in the feature selection task (e.g., filter methods).

Specifically, wrapper methods, although effective in eliminating irrelevant and redundant attributes, are very slow because they apply the mining algorithm many times, changing the number of attributes each time of execution as they follow some search and stop criteria [2]. Filter methods are more efficient; they use some form of *correlation measure* between individual attributes and the class [3]; however, because they measure the relevance of each isolated attribute, they cannot detect if redundant attributes exist, or if a combination of two (or more) attributes, apparently irrelevant when analyzed independently, are indeed relevant [4].

In this article we propose a feature selection method that tries to solve these problems in a supervised learning context. Specifically, we use a heuristic search alternative, inspired by the Branch & Bound algorithm, which reduces considerably the search space, thus reducing the processing time. Additionally, we propose a novel evaluation criterion based on an *n-way* entropy measure that, at the same time, selects the relevant attributes and discovers the important inter-dependences among variables of the problem.

To cover these topics, the article is organized as follows: Section 2 surveys related work; Section 3 introduces our feature selection method; Section 4 details the experiments; conclusions and future research directions are offered in Section 5.

2 Related Work

The emergence of Very Large Databases (VLDB) leads to new challenges that the mining algorithms of the 1990´s are incapable to attack efficiently. According to [5], from the point of view of the mining algorithms, the main lines to deal with VLDB (scaling up algorithms) are: a) to use relational representations instead of a single table; b) to design fast algorithms, optimizing searches, reducing complexity, finding approximate solutions, or using parallelism; and c) to divide the data based on the variables involved or the number of examples. In particular, some of these new approaches in turn give origin to Data Reduction that tries to eliminate variables, attributes or instances that do not contribute information to the KDD process. These methods are generally applied before the actual mining is performed.

In fact, the specialized literature mentions the *curse of dimensionality*, referring to the fact that the processing time of many induction methods grows dramatically (sometimes exponentially) with the number of attributes. Searching for improvements on VLDB processing power (necessary with tens of attributes), two main groups of methods have appeared: wrappers and filters [5]. We focus our research on filters methods with, near to, optimum solutions because of their relatively low computational cost.

Narendra [6] and others [7], [8], [9] have proposed a filter method for optimal feature selection. In general, they use the Branch & Bound algorithm, starting the search with all the D features and then applying a backward elimination feature strategy, until they obtain d optimal features ($d < D$). Additionally, they use a monotonic subset feature evaluation criterion: i.e., when augmenting (subtracting) one feature to the feature subset, the criterion value function always increases (decreases). The monotonicity property allows us to prune unnecessary sub-trees (e.g., sub-trees that do not improve the solution because they have values less than the bound obtained for another sub-tree). These approaches have demonstrated to be efficient; however, they have several drawbacks, because they need:

- An a priori definition of the number of features d (equal to the maximum tree deep level to consider); this is a problem because, in most cases, the number of relevant attributes is previously unknown,
- To start evaluating all the features (top-down strategy); this strategy represents high computational cost at the beginning of the subset feature search process,
- To use a monotonic subset evaluation criterion; although a monotonic criterion permits safe sub-trees cut offs, it assumes that the features are independent between each other, given the class attribute.

Trying to tackle these problems, in this paper we propose a bottom-up Best First method that is described in the next Section.

3 Bottom-Up Best First

The proposed method basically has two components: a) the evaluation function of each feature subset (in a supervised learning context), and b) the search strategy.

3.1 Evaluation criterion

With respect to the feature subset evaluation criterion, we proposed a non-monotonic function that, essentially, is calculated in a similar way to the Shannon entropy, only that instead of considering the entropy of one single feature, or attribute, against the class attribute (*2-way* entropy, or traditional entropy), it is calculated considering the entropy of two (or more attributes) against the class (*n-way* entropy). With this approach, we sought to capture the inter-dependences among attributes.

Formally, the traditional entropy H of a variable X after observing values of another variable Y is defined as:

$$H(X \mid Y) = -\Sigma_j P(y_j) \; \Sigma_i P(x_i \mid y_j) \log_2 (P(x_i \mid y_j)), \quad (1)$$

where $P(x_i \mid y_j)$ is the posterior probabilities of X given the values of Y. We obtain the *n-way* entropy *Hn* with the same equation but, instead of using the count of only one attribute, we count the number of times that a particular combination of attribute values appears, against the class value, taking into account all the instances of the dataset. In this form, if the *n-way* entropy *Hn* decreases, using a particular feature subset, means that we have additional information about the class attribute. For instance, if U and V are different attribute subsets, C is the class attribute, and if $Hn(U|C) > Hn(V|C)$, then we conclude that subset V predicts better than subset U.

The idea of calculating in this manner the *n-way* entropy is inspired by the work of Jakulin and Bratko [10]. Although they calculate this in a more costly way using the concept of Interaction Gain I. For instance, they obtain the 3-way interactions using:

$$I(X; Y; C) = H(X|C) + H(Y|C) - H(X,Y|C) - \{ H(X) + H(Y) - H(X,Y) \}, \quad (2)$$

so, we experiment with the *n-way* entropy variant *Hn* because of its simplicity and its relative low computational cost.

Nevertheless, a defect or problem with the *n-way* entropy *Hn* is that it decreases quickly when the number of the combined attribute values grows, resulting as ”false” low entropy. In an extreme case, it is possible that we can count as many different combined attribute values as the total number of dataset instances. If we count as many combined attribute values as instances, then the entropy will be zero (perfect). But this does not necessarily reflect, in an effective way, how that combination of attributes is relevant. The specialized literature has already reported how the entropy tends to prefer those attributes that have more different values, then, an attribute randomly generated could be considered better than another attribute observed from the real system.

Although there are some proposals to mitigate the problem (e.g., gain ratio or symmetrical uncertainty), they usually add an extra computational cost; because of that, we directly apply a reward to the *n-way* entropy considering the number of values that a specific attribute (or attributes) can take. Our proposed evaluation criterion, or metric, is defined as:

$$nwM = \lambda \; (Hn) + (1 - \lambda)(\textit{tot. combined attribute values} \; / \; \textit{tot. instances}) \quad (3)$$

With this metric, a balance between the *n-way* entropy *Hn* and the combined attribute values is sought, obtaining a metric, now called *nwM*, to detect relevant and inter-dependant features. The λ parameter can take values between zero and one and it is defined by the user according to how much weight he desires to give to each term. We empirically test the proposed metric, and obtain very promising results (see Section 4).

3.2 Search strategy

With respect to the search strategy, we propose to explore a search tree with forward feature selection or bottom-up schema.

The idea consists in using a best first search strategy: always expanding (aggregates a new feature) to the node (attribute subset) whose metric is the best of the brother nodes (node with the smaller *nwM*) and better than the parent node, stopping the search when none of the expanded nodes are better than the parent node. In this case, following the best first search strategy, the search continues selecting the best non-expanding node, according to the metric, and expanding until none of the children nodes are better than the parent node, and so on.

Thus, the proposed search schema explores the most promising attribute combinations according to the non-monotonic metric, generating several possibly good solutions. At the same time, it carries out sub-tree pruning, when the *nwM* metric has indicated, heuristically, that continuing to explore some of those sub-trees, maybe will not improve the evaluation criterion. The search process stops when the memory has been saturated, or when all the nodes have been expanded. The modified algorithm, called now bottom-up Best First (buBF), is shown in Fig. 1 (|| . || is the set size).

Given a dataset with D features and N instances, and $\lambda \in [0,1)$,

1. obtain *nwM* (2-*way* entropy) for each feature in the dataset;
2. while (available memory) or (unexplored nodes) do begin
3. select for expansion the feature subset F with the best *nwM* and better than his parent node;
4. for I := 1 to (D – || F ||) do begin
5. obtain *nwM* ($F \cup I \mid I \notin F$);
6. end;
7. end;
8. show feature subset with the best *nwM*;

Fig. 1. buBF algorithm.

The proposed search seems like a Branch & Bound strategy, in the sense that it prunes sub-trees that maybe will not conduct to better solutions, according to the evaluation criterion. Nevertheless, it is not exactly equal to the feature selection Branch & Bound schema reported in the specialized literature.

The differences basically consist of:

- Instead of removing attributes and evaluating the resulting feature subset (backward elimination), our method adds attributes and evaluates (forward selection). Using forward selection we will be able to process datasets with more features.
- Instead of using a monotonic evaluation criterion, a non-monotonic criterion is employed. Although sub-tree pruning is not safe using a non-monotonic criterion, our heuristic measure captures attributes inter-dependencies.
- Instead of having to define an a priori tree depth, in our case the tree depth search is variable, and depends on the evaluation criterion: this criterion indicates stopping the depth search when none children node is better than the parent node.
- In our case, adding nodes (attributes) is sought to determine not only the relevant attributes, but also their inter-dependences, being that other methods reported in the literature assumes attribute independence [8].

4 Experiments

We conducted several experiments with synthetic and real datasets to empirically evaluate if buBF can do better in selecting features than other well-known feature selection algorithms, in terms of learning accuracy and processing time. We choose synthetic datasets in our experiments because the relevant features of these datasets are known beforehand.

4.1 Experimentation details

The experimentation objective is to observe the buBF behavior related to classification quality and response time using 10 synthetic datasets, each of them with different levels of complexity. To obtain the

10 datasets we use the functions described in [11]. Each of the datasets has nine attributes (1.salary, 2.commission, 3.age, 4.elevel, 5.car, 6.zipcode, 7.hvalue, 8.hyears, and 9.loan) plus the class attribute (with class label Group "A" or "B"); each dataset has 10,000 instances. The values of the features of each instance were generated randomly according to the distributions described in [11]. For each instance, a class label was determined according to the rules that define the functions. For example, function 9 uses four attributes and classifies an instance following the statement and rule shown in Fig. 2.

---

disposable := (0.67 * (*salary* + *commission*) – 5000 * *elevel* – 0.2 * *loan* – 10000)

IF (disposable > 0) THEN class label := Group "A"
ELSE class label := Group "B"

Fig. 2. A function example.

We experiment too with the corrAL (and corrAL-47: see [12] for details) synthetic dataset, that has four relevant attributes (A0, A1, B0, B1), one irrelevant (I) and one redundant (R); the class attribute is defined by the function Y = (A0 ∧A1) ∨ (B0 ∧ B1). Finally, we test our proposed method with a real database with 24 attributes and 2,770 instances; this database contains information of Mexican electric billing costumers, where we expect to obtain patterns of behavior of illicit customers.

In order to compare the results obtained with buBF, we use Weka´s [13] implementation of ReliefF, OneR and ChiSquared feature selection algorithms. These implementations were run using Weka´s default values, except for ReliefF, where we define to 2 the number of neighborhood, for a more efficient response time. Additionally, we experiment with 7 Elvira´s [14] filter-ranking methods: *Mutual Information, Euclidean, Matusita, Kullback-Leibler-1 and 2, Shannon and Bhattacharyya.*

To select the best ranking attributes, we use a threshold defined by the largest gap between two consecutive ranked attributes (e.g., a gap greater than the average gap among all the gaps [12]). In the case of buBF, we set λ to 0.85 for all the experiments. All the experiments were executed in a personal computer with a Pentium 4 processor, 1.5 GHz, and 250 Mbytes in RAM. In the following Section the obtained results are shown.

4.2 Experimental results

Using 10 synthetic datasets, the features selected by each method are shown in Table 1, where "Oracle" represents a perfect feature selection method (it selects exactly the same features that each function uses to generate the class label). We can observe that, in some cases, the methods almost select the same features, but there are other functions in which the methods disagree. For function 8, only OneR cannot determine any feature subset, because ranks all attributes equally.

Next, we used the selected features for each method as input to the decision tree induction algorithm J4.8 included in the Weka tool. J4.8 is the last version of C4.5, which is one of the best-known induction algorithms used in data mining. We use 10-fold cross validation in order to obtain the average test accuracy for each feature subset. The results are shown in Table 2 (in this case, using all the attributes results in the same accuracy than using only the oracle attributes).

To summarize the obtained results in Table 2, we count the times when buBF wins, losses or ties against the other methods. This information is reported in Table 3, where it can be observed that buBF has a good performance, because there was only loss one time versus ReliefF, and one time versus ChiSquared, but it still maintained good accuracy.

Table 1. Features selected by different methods (10 synthetic datasets).

Function number	Method											
	Oracle	**Mut.Infor**	**Euclidean**	**Matusita**	**Kullback Leibler-1**	**Kullback Leibler-2**	**Shannon**	**Bhattach**	**ReliefF**	**OneR**	**ChiSquar**	**buBF**
1	3	3	3	3	3	9-7-2-8	9-1	3	3	3	3	3
2	1-3	1	2-1	1	1-2	1	9-3-7-1	1	3-1	1	1-2	3-1
3	3-4	4-3	4	4-3	4-3	4-3	3-9-1	4-3	4-3	4-3	4-3	3-4
4	1-3-4	1	2-1	1	1	1	1-9	1	1-4-2	1-2	1-2	4-3-1
5	1-3-9	9-1	9-4	9	9	9-1	1-3	9	9-3-1	9	9	5-2-3-9
6	1-2-3	1-3-2	2	1-3	1-3	1	3	1-3-2	3-1-2	3-1-2	1-3-2	1-2-3
7	1-2-9	9	2-9	9	9-1-2	9	1-9	9-1	9-1-2	9	9-1-2	9-1-2
8	1-2-4	2-1	2-4-1	2-1	2-1-4	2-1	9-3	2-1	1-2-4	-	1-2-4	4-2-1
9	1-2-4-9	9	2-4-9	9-1	9	9	9	9-1	9-1-2	9	9-1-2-4-3	2-1-9
10	1-2-4-7-8-9	4	4	4	4	4	9-1-3	4	8	4	4-8-7-6	6-8-4

Table 2. J4.8´s accuracies (%) using the features selected by each method (10 synthetic datasets).

Function number	Method											
	Oracle/ All	**buBF**	**ReliefF**	**ChiSquar**	**Bhattach**	**Mut.Infor**	**Kullback Leibler-1**	**Matusita**	**OneR**	**Kullback Leibler-2**	**Euclidean**	**Shannon**
1	100	100	100	100	100	100	100	100	100	67	100	67
2	100	100	100	73	73	73	73	73	73	73	73	100
3	100	100	100	100	100	100	100	100	100	100	68	59
4	100	100	90	84	84	84	84	84	84	84	84	84
5	100	91	100	74	74	82	74	74	74	82	74	60
6	99	99	99	99	99	99	87	87	99	68	64	69
7	98	98	98	98	94	86	98	86	86	86	88	94
8	100	100	100	100	99	99	100	99	-	99	100	98
9	97	94	94	97	92	85	85	92	85	85	88	85
10	99	99	80	99	97	97	99	97	98	97	97	80
Avg.	**99.3**	**98.1**	**96.1**	**92.4**	**91.2**	**90.5**	**89.8**	**89.2**	**84.9**	**84.1**	**83.6**	**79.6**

Table 3. buBF accuracy results summary vs. other methods (10 synthetic datasets).

buBF vs.	Method											
	Oracle/ All	**OneR**	**ReliefF**	**ChiSquar**	**Bhattach**	**Mut.Infor**	**Kullback Leibler-1**	**Matusita**	**Shannon**	**Kullback Leibler-2**	**Euclidean**	**Average**
Win	0	7	2	3	7	7	5	8	9	9	8	**5.9**
Loss	2	0	1	1	0	0	0	0	0	0	0	**0.4**
Tie	8	3	7	6	3	3	5	2	1	1	2	**3.7**

With respect to the processing time, this is shown in Table 4. We observe that, although buBF is computationally more expensive than OneR and ChiSquared, these algorithms cannot detect some attribute inter-dependencies; on the other hand, buBF is faster than ReliefF, but with similar, or better, feature selection performance.

To have a better idea of the buBF performance, we can compare the results presented previously against the results produced by an exhaustive wrapper approach. In this case, we can calculate that, if the average time required to obtain a tree using J4.8 is 1.1 seconds, and if we multiply this by all the possible attribute combinations, then we will obtain that 12.5 days, theoretically, would be required to conclude such a process.

Table 4. Averaged processing time for each method (10 synthetic datasets).

Exhaustive wrapper	ReliefF	OneR	ChiSquared and Elvira	buBF
1,085,049 secs. (12.5 days)	573 secs. (9.55 mins.)	8 secs.	1 sec.	71 secs. (1.18 mins.)

In order to observe how the selected features (Table 1) respond with another classifier, we use these features as input to the Naïve Bayes Classifier (NBC) included in the Weka tool. Results are shown in Table 5. Again, buBF obtains satisfactory accuracy results.

Table 5. NBC´s averaged accuracies (%) for 10-fold-cross validation using the features selected by each method (10 synthetic datasets).

Function number	Method											
	Oracle	buBF	Matusita	Kullback Leibler-1	Bhattach	Mut.Infor	ChiSquar	ReliefF	Euclidean	Kullback Leibler-2	OneR	Shannon
1	89	89	89	89	89	89	89	89	89	67	89	67
2	69	69	69	64	69	69	64	69	64	69	69	68
3	65	65	65	65	65	65	65	65	66	65	65	58
4	76	76	76	76	76	76	70	69	70	76	70	76
5	68	68	68	68	68	68	68	68	68	68	68	60
6	71	71	72	72	71	71	71	71	59	60	71	58
7	89	89	86	89	88	86	89	89	86	86	86	88
8	99	99	98	99	98	98	99	99	99	98	50	98
9	89	88	88	85	88	85	88	88	86	85	85	85
10	98	98	98	98	98	98	97	80	98	98	98	80
Avg.	**81.3**	**81.2**	**81**	**81**	**81**	**80.5**	**80**	**78.7**	**78.5**	**77.2**	**75.1**	**73.8**

When we test with the corrAL and corrAL-47 datasets [12], our method was the only that can remove the redundant attribute (Table 6; results for FCBF method was taken from [12]). This suggests that our method, although requires more processing time, is a good approach to capture inter-dependencies among attributes. On the other hand, buBF processing time is competitive when we try to use wrapper feature selection methods.

Finally, testing over the electric billing database, buBF obtains the best accuracy ties with Kullback-Leibler-2, but with less attributes (Table 7).

We point out that we do not carry out comparisons against Branch & Bound methods because, in general, these require a previous definition of the number of attributes to select, which is not necessary with buBF.

Table 6. Features selected by different methods (corrAL and corrAL-47 datasets).

Method	Features selected	
	corrAL	**corrAL-47**
buBF	B1, B0, A1, A0	A0, A1, B0, B1
ReliefF	R, A0, A1, B0, B1	R,$B1_1$,A0,$A0_0$,B1,$B1_0$,B0,$B0_0$,$B0_2$,A1,$A1_0$
$FCBF_{(log)}$	R, A0	R, A0, A1, B0, B1
$FCBF_{(0)}$	R, A0, A1, B0, B1	R, A0, A1, B0, B1

Table 7. J4.8´s accuracies (%) for 10-fold-cross validation using the features selected by each method (electric billing database).

Method	**Total features selected**	**Accuracy (%)**	**Pre-processing time**
buBF	5	97.50	1.5 mins.
Kullback-Leibler 2	9	97.50	6 secs.
All attributes	24	97.25	0
ChiSquared	20	97.18	9 secs.
OneR	9	95.95	41 secs.
ReliefF	4	93.89	14.3 mins.
Euclidean distance	4	93.89	5 secs.
Shannon entropy	18	93.71	4 secs.
Bhattacharyya	3	90.21	6 secs.
Matusita distance	3	90.21	5 secs.
Kullback-Leibler 1	4	90.10	6 secs.
Mutual Information	4	90.10	4 secs.

5 Conclusions and Future Work

We have presented a new algorithm for feature selection that tries to overcome some drawbacks found in Branch & Bound feature selection algorithms. Thus, the proposed method follows a forward attribute selection (instead of backward, like other methods do) finding reductions in processing time, because it is less costly to obtain the evaluation criterion for few attributes than for all the features.

Additionally, we propose a new subset evaluation criterion, that considers a balanced *n-way* entropy with respect to the combined attribute values; this metric is not very expensive and, due to the fact that is non-monotonic, heuristically allows pruning the search tree, with additional processing time savings. Furthermore, the *n-way* entropy considers the inter-dependences among features, obtaining not only isolated relevant features, and doing unnecessary a previously definition of the tree depth.

From the experimental results, the proposed method buBF represents a promising alternative, compared to other methods, because of its acceptable processing time and good performance in the feature selection task.

Some future research issues arise with respect to buBF improvement. For example: further experimentations with more real databases; comparing against other similar methods (e.g., Liu´s ABB [15]); using another metric variations to eliminate the λ parameter (e.g., DKM) and using more efficient

search methods (e.g. multi-restart hill-climbing); improving the tree pruning strategy and test the method with data sets with more instances and attributes.

References

1. Guyon, I., Elisseeff, A., An introduction to variable and feature selection, Journal of machine learning research, 3, 2003, pp. 1157-1182.
2. Kohavi, R., John, G., Wrappers for feature subset selection, Artificial Intelligence Journal, Special issue on relevance, 1997, pp. 273-324.
3. Piramuthu, S., Evaluating feature selection methods for learning in data mining applications, Proc. 31st annual Hawaii Int. conf. on system sciences, 1998, pp. 294-301.
4. Molina, L., Belanche, L., Nebot, A., Feature selection algorithms, a survey and experimental eval, IEEE Int.conf.data mining, Maebashi City Japan, 2002, pp. 306-313.
5. Mitra, S., et.al., Data mining in soft computing framework: a survey, IEEE Trans. on neural networks, vol. 13, no. 1, January, 2002, pp. 3-14.
6. Narendra, P., Fukunaga, K., A branch and bound algorithm feature subset selection, IEEE Trans. computers, vol. 26, no. 9, sept 1977, pp. 917-922.
7. Yu, B., Yuan, B., A more efficient branch and bound algorithm for feature selection, Pattern Recognition, vol. 26, 1993, pp. 883-889.
8. Frank, A., Geiger, D., Yakhini, Z., A distance-B&B feature selection algorithm, Procc. Uncertainty in artificial intelligence, México, august. 2003, pp. 241-248.
9. Somol, P., Pudil, P., Kittler, J., Fast Branch & bound algorithms for optimal feature selection, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 26, no. 7, july 2004, pp. 900-912.
10. Jakulin, A., Bratko, I., Testing the significance of attribute interactions, Procc. Int. conf. on machine learning, Canada 2004, pp. 409-416.
11. Agrawal, R., Imielinski, T, Swami, A., Database mining: a performance perspective, IEEE Trans. Knowledge data engrg. Vol. 5, no. 6, 1993, pp. 914-925.
12. Yu, L., Liu, H., Efficient feature selection via analysis of relevance and redundancy, Journal of Machine Learning Research 5, 2004, pp. 1205-1224.
13. www. cs.waikato.ac.nz/ml/weka, 2004.
14. www. ia.uned.es/~elvira/ , 2004.
15. Liu, H. Motoda, and M. Dash. A monotonic measure for optimal feature selection. In Proceedings of European Conference on Machine Learning,, 1998, pp. 101-106.

Pairwise Constraints-Guided Dimensionality Reduction

Wei Tang* Shi Zhong†

Abstract

Dimensionality reduction is a commonly used technique to handle high-dimensional data. It is well-studied in both unsupervised learning (clustering) and supervised learning (classi cation)—Principal Component Analysis and Fisher's Linear Discriminant Analysis are representative examples of the two categories, respectively. In this paper, we exploit a common type of background knowledge in the form of pairwise data constraints for e ective feature reduction. Pairwise constraints specify whether a pair of data instances are similar (i.e., should be grouped together) or not, and are often called *must-link* or *cannot-link* constraints. They naturally arise in many practical clustering problems. Unfortunately, both unsupervised and supervised dimensionality reduction techniques cannot take advantage of pairwise data constraints; the former do not consider any prior knowledge and the latter need labeled data (i.e., a class label for each instance). We propose two ways of reducing data dimensionality based on direct or indirect use of pairwise data constraints: constraints-guided feature projection and constrained co-clustering. We evaluate the proposed approaches on high-dimensional text clustering problems and experimental results are promising.

Keywords: Dimensionality Reduction; Feature projection; Feature Clustering; Clustering with Pairwise Constraints

1 Introduction

Very high-dimensional data such as text documents and market basket data, presents a challenge to traditional machine learning methods. The complexity of data mainly comes from sparsity and high dimensionality. In order to alleviate this problem, many research e orts have been made toward dimensionality reduction. In unsupervised learning, the most well-known method is the Principal Components Analysis (PCA) technique [12], which performs a linear transformation that projects data to a low-dimensional space such that maximum variance in the original data is preserved. Latent Semantic Indexing (LSI) [6], a similar method, is used for text data, and map text documents to a low-dimensional "topic" space spanned by some underlying latent concepts. In supervised leaning, the most well-known method is the Fisher's Linear Discriminant Analysis (LDA) [9] method. Given a set of labeled instances, LDA aims to nd one or more directions along which di erent classes can be best separated.

Unfortunately, both types of these existing methods, cannot take advantage of a common type of background knowledge—pairwise data constraints, which are available in many application domains. Generally, the pairwise constraints we considered can be divided into *must-link* constraints and *cannot-link* constraints. A *must-link* constraint means that the pair of instances must be in the same group while a *cannot-link* constraint means that the pair of instances must be in two di erent classes. We are interested in pairwise data constraints for the following reasons:

- The pairwise constraints are more general than class labels in type of knowledge, which we can always generate from the labeled data but cannot do so inversely, In addition, when there is not

*Dept. of CSE, Florida Atlantic University, 777 Glades Road, Boca Raton, FL 33431
†Data Mining and Research, Yahoo! Inc, 701 First Ave, Sunnyvale, CA 94089

enough labeled data to apply the supervised learning methods, a clustering approach with the supervision of constraints derived from the incomplete class information is more useful;

- Pairwise constraints are more natural in some scenarios and easier to collect than class labels. For example, in text or image retrieval systems with user feedback, users are more willing to provide answers on whether a set of retrieved items are similar or not than to specify explicit class labels, which is di cult and time-consuming for users to do;
- In some application domains, the class information can change dynamically. For example, in a network intrusion detection system, we will certainly encounter new attack types never seen before but we can collect pairwise constraints on a newly emerged category, which could be di cult to be detected by classi cation models based on existing category labels.

In this paper we propose two ways of reducing data dimensionality based on pairwise data constraints: constraints-guided feature projection and constrained co-clustering. The rst approach projects data into a low-dimensional space such that the (summed) distance between *must-link* data instances is minimized and that between *cannot-link* instances is maximized in the projected space. The solution to our formulated constrained optimization problem leads to an elegant eigenvalue decomposition problem similar in form to PCA/LDA. The second approach does feature clustering and bene ts from pairwise constraints via a constrained co-clustering mechanism [7]. Even though the constraints are imposed only on data instances (rows), the feature clusters (columns) are in uenced since row clustering and column clustering are entangled together and mutually-reinforced in the co-clustering process.

We evaluate our dimensionality reduction techniques on high-dimensional text clustering problems. Since the labeled data instances are not available, we cannot perform classi cation. We combine the proposed approaches with the existing clustering algorithms such as spherical kmeans algorithm [8], semi-supervised kmeans algorithm [14], and information theoretic co-clustering algorithm [7]. The experimental results demonstrate the bene ts of our proposed methods on signi cantly improved clustering performance. Although existing semi-supervised clustering techniques [1, 2] also have achieved improved clustering performance with pairwise constraints, our approaches provide the added bene t of lower dimensionality (thus lower computational complexity).

This paper is organized as follows. Section 2 describes the proposed pairwise constraints-guided feature projection and feature clustering algorithms. Section 3 presents experimental results on text clustering. Finally Section 4 concludes this paper and discusses future directions.

2 Pairwise Constraints-Guided Dimensionality Reduction

In this section, we rst present the pairwise constraints-guided feature projection approach, then describe the constrained co-clustering algorithm.

2.1 Pairwise Constraints-Guided Feature Projection. Given a set of pairwise data constraints, we aim to project the original data to a low-dimensional space, in which *must-link* instance pairs are close and *cannot-link* pairs far apart.

Let $X = \{x | x \in R^d\}$ be a set of d-dimensional column vectors and $F_{d\times k} = \{F_1, \ldots, F_k\}$ a projection matrix containing k orthogonal unit-length d-dimensional vectors. Let $C_{ml} = \{(x_1, x_2)\}$ be the set of all must-link data pairs and C_{cl} the set of all cannot-link data pairs. We aim to nd an optimal projection matrix F that maximizes the objective function

$$f = \max_F \sum_{(x_1,x_2)\in C_{cl}} \|F^T x_1 - F^T x_2\|^2 - \sum_{(x_1,x_2)\in C_{ml}} \|F^T x_1 - F^T x_2\|^2 , \tag{2.1}$$

subject to the constraints

$$F_i^T F_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} , \tag{2.2}$$

where $\|\cdot\|$ denotes L_2 norm.

There exists a direct solution to the above optimization problem. The following theorem shows that the optimal projection matrix $F_{d\times k}$ is given by the rst k eigenvectors of matrix $M = CDC^T$, where each column of $C_{d\times m}$ is a di erence vector $x_1 - x_2$ for a pair (x_1, x_2) in C_{ml} or C_{cl} and $D_{m\times m}$ is a diagonal matrix with each value on the diagonal corresponding to a constraint (1 for a cannot-link pair and -1 for a must-link pair).

THEOREM 2.1. *Given the reduced dimensionality k, the set of* must-link *constraints C_{ml} and* cannot-link *constraints C_{cl}, construct matrix $M = CDC^T$, where C and D are defined above. Then the optimal projection matrix $F_{d\times k}$ is comprised of the first k eigenvectors of M corresponding to the k largest eigenvalues.*

Proof. Consider the objective function

$$\begin{aligned} f &= \sum_{(x_1,x_2)\in C_{cl}} \|F^T(x_1 - x_2)\|^2 - \sum_{(x_1,x_2)\in C_{ml}} \|F^T(x_1 - x_2)\|^2 \\ &= \sum_{(x_1,x_2)\in C_{cl}} \sum_l F_l^T(x_1 - x_2)(x_1 - x_2)^T F_l - \sum_{(x_1,x_2)\in C_{ml}} \sum_l F_l^T(x_1 - x_2)(x_1 - x_2)^T F_l \\ &= \sum_l F_l^T \left[\sum_{(x_1,x_2)\in C_{cl}} (x_1 - x_2)(x_1 - x_2)^T - \sum_{(x_1,x_2)\in C_{ml}} (x_1 - x_2)(x_1 - x_2)^T \right] F_l \\ &= \sum_l F_l^T (CDC^T) F_l \\ &= \sum_l F_l^T M F_l \ , \end{aligned} \tag{2.3}$$

where F_l's are subject to constraints $F_l^T F_h = 1$ for $l = h$ and 0 otherwise.

Using the traditional Lagrange multiplier optimization technique, we write the Lagrangian

$$L_{F_1,\ldots,F_k} = f(F_1, \ldots, F_k) + \sum_{l=1}^{k} \xi_l (F_l^T F_l - 1) \tag{2.4}$$

by taking the partial derivative of $L_{F_1,\ldots,F_k}$ with respect to each F_l and set it to zero, we get

$$\frac{\partial L}{\partial F_l} = 2MF_l + 2\xi_l F_l = 0 \qquad \forall l = 1, \ldots, k \tag{2.5}$$

$$\Rightarrow \quad MF_l = -\xi_l F_l \qquad \forall l = 1, \ldots, k \ . \tag{2.6}$$

Now it is obvious that the solution F_l is an eigenvector of M and $-\xi_l$ the corresponding eigenvalue of M. To maximize f, F must be the rst k eigenvectors of M which makes f the sum of the k largest eigenvalues of M.

When d is very large, $M_{d\times d}$ is a huge matrix which can present di culties to the associated eigenvalue decomposition task. In this case, we don't really need to compute M since its rank is most likely much lower than d and we can use the Nystrom method [3] to calculate the top k eigenvectors more e ciently.

2.2 Constrained Co-clustering. Feature clustering in general can be viewed as a special case of feature projection, with each projection vector containing only 0/1 values (for hard clustering). What is described in this section, however, is a unique feature clustering method that cannot be regarded as a special case of feature projection since it involves co-clustering of data instances and features. Usually the feature reduction is just a means to some end (clustering or classi cation), thus after the feature projection process described in the previous section we run data clustering in the projected data space. For the co-clustering approach, the means and end are mixed together. What we add to this unique approach in this paper is to exploit the pairwise constraints in the co-clustering process.

The co-clustering algorithm used here is proposed in [7] and aims to minimize the following objective function

$$I(X;Y) - I(\hat{X};\hat{Y}) \tag{2.7}$$

subject to the constraints on the number of row and column clusters. $I(X;Y)$ is the mutual information between the row random variable X, which governs the distribution of rows, and the column random variable Y, which governs the distribution of columns. $\hat{X}$ and $\hat{Y}$ are variables governing the distribution of clustered rows and clustered columns, respectively. An iterative algorithm was used in [7] to alternate between clustering rows and clustering columns to reach a local minimum of the above objective function.

Due to space limit, we omit detailed discussion of the co-clustering algorithm and readers are referred to [7]. Also, here we just concisely describe how we involve constraints in the co-clustering process: The constraints only a ect the row/data clustering step algorithmically and the impact on column/feature clustering is implicit. For must-link data pairs, we merge the rows and replace each instance by the average; for cannot-link data pairs, we separate a pair if they are in the same cluster after an iteration of row clustering, by moving the instance that is farther away from cluster centroid to a di erent cluster. Essentially, the idea of handling constraints is similar to the existing work [14, 1] but we get the feature clustering involved by co-clustering. This combination of pairwise constraints and co-clustering seems to have not appeared elsewhere in the literature.

3 Experiments

In this section, we provide empirical results to show the bene t of our dimensionality reduction methods for high dimensional text clustering, using comparisons to the clustering algorithms that do not consider such "supervised" feature reduction strategies.

3.1 Datasets. For our experiments, we constructed three datasets from the 20-Newsgroup collection [11]. The 20-Newsgroup collection consists of approximately 20,000 newsgroup articles gathered evenly from 20 di erent Usenet newsgroups. From the original dataset, three datasets are created by selecting particular group categories. *News-Similar-3* consists of 3 newsgroups on similar topics (`comp.graphics`, `comp.os.ms-windows` and `comp.windows.x`) with signi cant overlap between clusters due to cross-posting. *News-Related-3* consists of three newsgroups on related topics (`talk.politics.misc`, `talk.politics.guns` and `talk.politics.mideast`). *News-Different-3* consists of three well-separated newsgroups that cover quite di erent topics (`alt.atheism`, `rec.sport.baseball` and `sci.space`). All the datasets were converted to the vector-space representation following several steps—tokenization, stop-word removal, and removing words with very high-frequency and low frequency [8]. Since semi-supervised co-clustering algorithm can directly cluster the document-term matrix (treated as a probability distribution), we did not apply TF-IDF weighting. For the spherical kmeans algorithms, we used TF-IDF weighting since it helps improve performance. Table 3.1 summarizes the properties of the datasets.

Table 1: Datasets used in experimental evaluation

Dataset	Instances	Dimensions	Classes
News-Similar-3	295	1864	3
News-Related-3	288	3225	3
News-Different-3	300	3251	3

3.2 Constraint Generation and Handling. In this section, we describe how we generate pairwise data constraints and how we process them in our experiments prior to feature reduction.

Since the external class labels are available in our benchmark data sets, we randomly select pairs of di erent instances and create *must-link* and *cannot-link* depending on whether the class labels of the two instances are the same or di erent. We rst consider how to pre-process the *must-link* constraints. Since the set of *must-link* constraints represents an equivalence relation, we can take transitive closure over them. After getting the equivalent connected components consisting of instances connected by *must-link*, we replace the instances within each components with their average and modify the data set and the *cannot-link* constraints accordingly. As for the *cannot-link* constraints, it is shown in [5] that nding a feasible solution for *cannot-link* constraints is much harder than for *must-link* constraints (actually NP-complete). Therefore, we adopted a heuristic method which assigns each instance into its nearest cluster where no previous instances involved in the same *cannot-link* reside during the assignment procedure, which is similar to the approach used in [14].

3.3 Experimental Setting. In our experiments, we adopted *normalized mutual information* (NMI) as our clustering evaluation methods. NMI, an external validation metric, estimates the quality of the clustering with respect to the given true labels of the datasets [13]. If $\hat{Z}$ is the random variable denoting the cluster assignments of the instances and Z is the random variable denoting the underlying class labels, the NMI is de ned as

$$NMI = \frac{I(\hat{Z}; Z)}{(H(\hat{Z}) + H(Z))/2} \tag{3.8}$$

where $I(Z; \hat{Z}) = H(Z) - H(Z|\hat{Z})$ is the mutual information between the random variables Z and $\hat{Z}$, $H(Z)$ is the Shannon entropy of Z and $H(Z|\hat{Z})$ is the conditional entropy of Z given $\hat{Z}$ [4].

In order to avoid bias caused by the di erent number of reduced dimensionality, we varied its value from 10 to 50 with a increment step of 10. The experimental result are similar for the di erent number of dimensionality, hence we choose $k = 20$ to present the result. To initialize the seeds for clustering, we adopted the strategy in [10], which can accelerate the convergence of clustering and lead to a better local clustering solution.

We repeated the experiment for each dataset with 5 runs of 2-fold cross-validation. In order to demonstrate the e ect of constraints in clustering, at each fold, 50% of dataset is set aside as the test set while the remaining is used as the training set. All constraints are generated and pre-processed used the methods mentioned in Section 3.2 from the training set. The clustering algorithms were run on the training set and NMI was calculated only on the test set. The results were averaged over 5 runs.

3.4 Results and Analysis. For each of our 3 datasets we tested the following clustering algorithms:

- `sp-kmeans`: the original spherical kmeans algorithm [8];
- `sp-kmeans+fp`: the unsupervised spherical kmeans algorithm applied after constraints-guided feature projection;

- `sp-kmeans+pc`: the original semi-supervised spherical kmeans algorithm with pairwise constraints, without dimensionality reduction [14];
- `sp-kmeans+pc+fp`: the semi-supervised spherical kmeans with pairwise constraints, on projected data;
- `co-clustering`: the original information theoretic co-clustering algorithm proposed by Dhillon et al [7];
- `co-clustering+pc`: the constrained co-clustering described in Section 2.2.

In our first experiment, we compare the performance of `sp-kmeans+pc`, `sp-kmeans+fp`, `sp-kmeans+pc+fp` and the unsupervised `sp-kmeans` algorithm. The results are shown in figure 1(a), 1(c) and 1(e). As the results demonstrated, the `sp-kmeans+pc+fp` algorithm outperform the `sp-kmeans+pc` algorithm with certain portions of pairwise constraints, but with the further increment of pairwise constraints the benefit gained from feature projection is not as significant as that from directly applying constraints instancewise. It reveals that dimensionality reduction via pairwise constraints-guided feature projection can extract meaningful relevant features and improve the quality of clustering. It should be cautioned, however, reducing the dimensionality also causes possible information loss. Thus we do not want to reduce the dimensionality too low to destroy the purpose of capturing discriminative power of features. Furthermore, although the performance of algorithm `sp-kmeans+fp` is almost as good as the other two semi-supervised methods. It is definitely better than the original spherical k-means algorithm, and seems to have captured most of the information in pairwise constraints for extracting discriminative features.

In our second experiment, we compared the performance of `co-clustering+pc` with the unsupervised `co-clustering` algorithm. The results are shown in figure 1(b), 1(d) and 1(f), from which we can see that when increasing the number of pairwise constraints the performance of constrained co-clustering algorithm improves significantly compared to the unguided version. As we discussed in Section 2.2, the co-clustering interweaves instance (row) clustering and feature (column) clustering at the same time. Imposing the pairwise constraints on instances can also affect the parameters for the feature clustering, hence indirectly contributes to the feature clustering part as well.

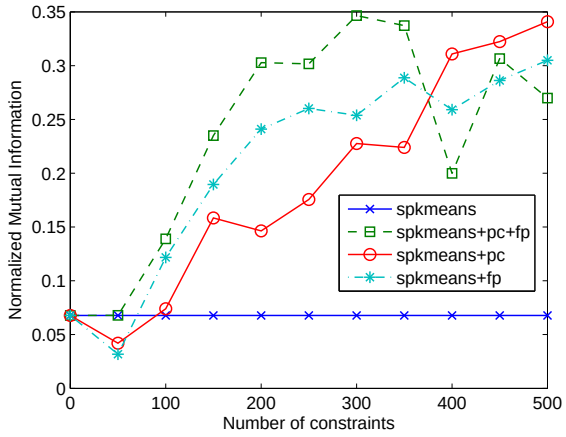
4 Conclusions and Future Work

In this paper, we have introduced the pairwise constraints guided dimensionality reduction, and proposed two clustering algorithm to utilize this idea to improve the quality. Pairwise constraints guided dimensionality reduction is a new way of imposing supervision to improve the quality of clustering. The experimental results on the selected text datasets demonstrate the efficacy of our proposed algorithms.

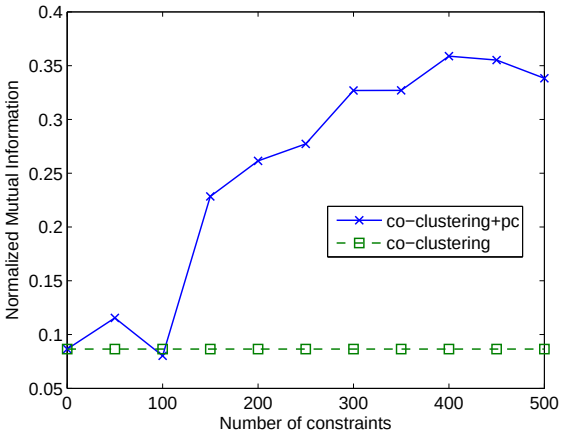
The performance compare to the other use of pairwise constraints, such as distance learning, remains unknown. In our future work, experimental comparison to distance learning techniques will be performed. Although the feature projection via pairwise constraints can make certain achievements, the number of projected features is chosen ad hoc in our experiments. How to find out the appropriate number for the feature projection is another interesting research topic.

5 Acknowledgements

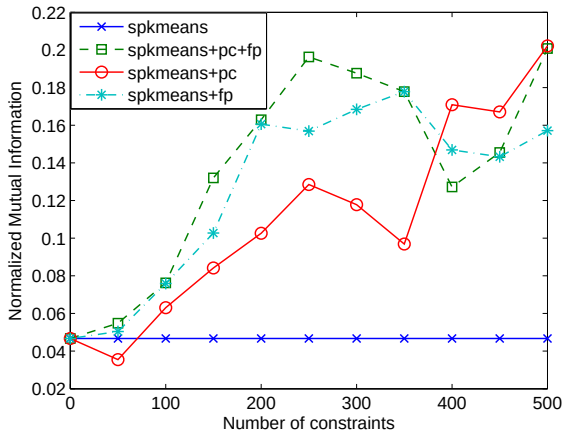
We would like to thank the anonymous reviewer for helpful comments.



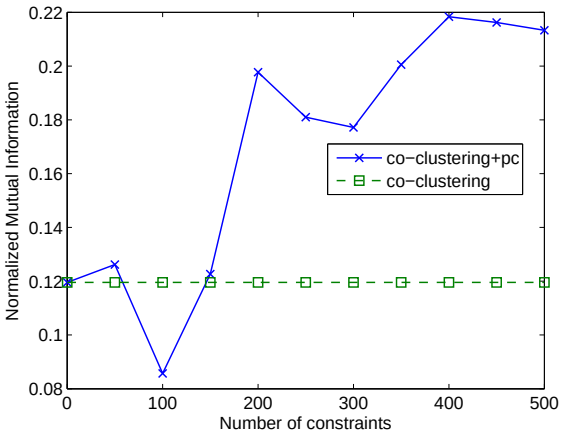
(a) Feature projection on News-Similar-3



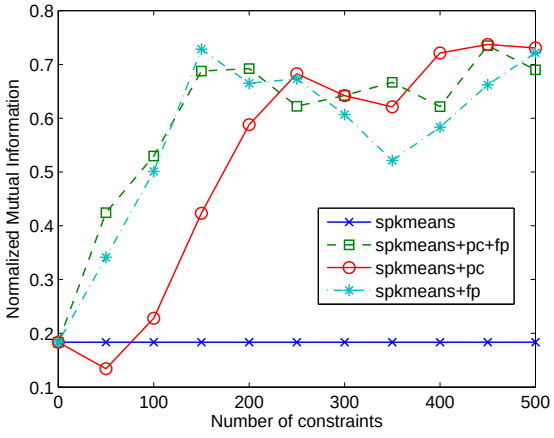
(b) Feature clustering on News-Similar-3



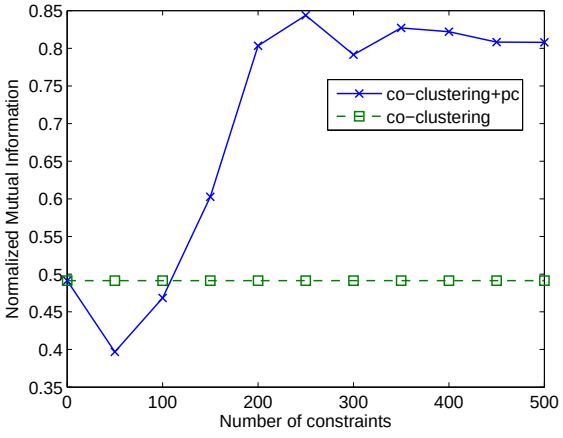
(c) Feature projection on News-Related-3



(d) Feature clustering on News-Related-3



(e) Feature projection on News-Different-3



(f) Feature clustering on News-Different-3

Figure 1: Results on the text data sets

References

[1] S. Basu, A. Banerjee, and R. J. Mooney. Semi-supervised clustering by seeding. In *Proc. 19th Int. Conf. Machine Learning*, pages 19–26, 2002.

[2] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proc. 10th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining*, pages 59–68, Seattle, WA, August 2004.

[3] Christopher Burges. Geometric methods for feature extraction and dimensionality reduction: a guided tour. Technical Report MSR-TR-2004-55, Microsoft, 2004.

[4] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, 1991.

[5] I. Davidson and S. S. Ravi. Clustering with constraints: Feasibility issues and the k-means algorithm. In *Proc. 5th SIAM Int. Conf. Data Mining*, Newport Beach, CA, 2005.

[6] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[7] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proc. 9th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining*, pages 89–98, August 2003.

[8] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42:143–175, 2001.

[9] R. O. Duda, P. E. Hart, and D. H. Stork. *Pattern classification*. Wiley Interscience, 2nd edition, 2000.

[10] I. Katsavounidis, C. J. Kuo, and Z. Zhang. A new initialization technique for generalized lloyd iteration. *IEEE Signal Processing Letters*, 1(10):144–146, October 1994.

[11] K. Lang. News weeder: learning to lter netnews. In *Proc. 12th Int. Conf. Machine Learning*, pages 331–339, 1995.

[12] K. Pearson. On lines and planes of closest t to systems of points in space. *Philosophical magazine*, 2(6):559–572, 1901.

[13] A. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. In *Workshop on Artificial Intelligence for Web Search*, pages 58–64, July 2000.

[14] K. Wagsta , C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proc. 18th Int. Conf. Machine Learning*, pages 577–584, 2001.

Unsupervised feature selection scheme for clustering of symbolic data using the multivalued type similarity measure

Bapu B. Kiranagi[1], D.S. Guru[1] and Venkat N. Gudivada[2]

[1]Department of Studies in Computer Science, University of Mysore, Manasagangotri, Mysore-570 006, India.
Email: bapukiranagi@lycos.com, guruds@lycos.com

[2]College of Information Technology and Engineering, Huntington Campus, Marshall University, USA.
email: gudivada@marshall.edu

Abstract:

In this paper, a simple and efficient unsupervised novel feature selection scheme for clustering of symbolic patterns is proposed. The proposed technique makes use of the mutlivalued type similarity measure proposed in (Guru et.al, 2004) for estimating the degree of similarity between two symbolic patterns.

Experiments on three standard data sets have been conducted in order to study the efficacy of the proposed methodology.

Keywords: Symbolic patterns, Similarity measure, Multivalued data type, feature selection, Unsupervised learning of symbolic patterns.

1. Introduction

Cluster analysis plays an important role in the field of pattern recognition, image processing, remote sensing, medicine etc. In general, cluster analysis is of great importance in classifying a heap of information, which is in the form of data patterns, into manageable and meaningful structures. An excellent study of many published works on conventional cluster analysis can be found in (Hartigan, 1975; Jain et.al 1999). The concept of clustering has also been extended to the patterns described by realistic/unconventional data types called symbolic data types (Gowda & Diday, 1992). Unlike conventional data sets, symbolic data sets are more unified by means of relationships and they appear in the form of continuous ratio, discrete absolute, interval, modal, multivalued and also multivalued data with weights (Bock and Diday, 2000), which are very much generic than the conventional ones. It is mentioned that these types of data play a major role in extending data mining to knowledge mining (Bock and Diday, 2000). Symbolic data analysis finds its applications in shape representation, character recognition etc.

Since a decade, many proximity measures were proposed for different types of data sets (Gowda and Diday, (1992); Gowda and Ravi, 1995(a, b); Prakash, 1998; Denoeux and Masson, 2000; Ichino and Yaguchi (1994)). Though the methods work on symbolic patterns, the degree of proximity between two symbolic patterns is assumed to be crisp and symmetric. Indeed, it is quite natural that the proximity values are themselves symbolic and are not necessarily symmetric. In this direction Guru et. al, (2004) have proposed a new similarity measure. This newly defined multivalued type similarity measure forms the basis for the feature selection scheme and the clustering algorithm proposed in this paper.

Nevertheless, all the methods will be cumbersome if the number of features describing patterns is very large. The length of such vectors representing the patterns depends upon the number of features recorded, leading to the creation of multidimensional features space. Apart from the analysis, the storage of the patterns will also be a problem. Hence, reducing the number of features or feature selection becomes necessary. However, many techniques or algorithms for conventional data have been proposed. The readers are directed to the paper (Liu and Yu, 2005) which has given an exhaustive survey on feature selection scheme for clustering conventional data. But, for symbolic data a very few techniques have been suggested (Chouakria et. al., 1995; Nagabhushan, 1995). Chouakria et. al., (1995), have applied PCA only for symbolic interval data. Similarly, Nagbhushan et. al., (1995) have devised a mathematical model

enabling the dimensionality reduction of n-dimensional symbolic data, particularly of interval type, to a lower-dimensional k-dimensional symbolic interval version. But it is a known fact that the PCA and other feature extraction algorithms tend to fail in identifying informative features if the features are of unlabeled data (Wolf and Bileschi, 2005). Except for these two methods, no concrete work can be found in dimensionality reduction/ feature selection of symbolic patterns. Further, it is well known that the feature selection methods help in retaining original features inorder to maintain the features physical interpretation (Liu et. al., 2005).

In view of this we present, an unsupervised feature selection scheme for symbolic data by the usage of the non symmetric similarity matrix obtained through the novel similarity measure to estimate the degree of similarity between two symbolic patterns proposed in (Guru et. al., 2004). The similarity measure unlike other available methods approximates the degree of similarity by multivalued type data and in addition it is non-symmetric. Furthermore, an unsupervised classification scheme is also explored by introducing the concept of Mutual Similarity Value (MSV). Experiments on three standard data sets have been conducted inorder to study the validity of the proposed methodology.

The rest of the paper is organized as follows. In section 2, a brief review of the similarity measure (Guru et. al., 2004) for estimating the degree of similarity between two symbolic patterns is given. An unsupervised feature selection scheme for symbolic data is proposed in section 3. Section 4, discusses an clustering method for symbolic patterns based on the concept of Mutual Similarity Value. The experimental results are presented in section 5 and finally, conclusion is given in section 6.

2. Review of the Similarity Measure

Let F_i and F_j be two symbolic patterns described by n interval valued features as follows (Guru et. al., 2004)

$F_i = \{F_{i1}, F_{i2}, \ldots, F_{in}\}$

$$i.e\ F_i = \{(f_{i1}^-, f_{i1}^+), (f_{i2}^-, f_{i2}^+), ..., (f_{in}^-, f_{in}^+)\}$$

$F_j = \{F_{j1}, F_{j2}, \ldots, F_{jn}\}$

$$i.e\ F_j = \{(f_{j1}^-, f_{j1}^+), (f_{j2}^-, f_{j2}^+), ..., (f_{jn}^-, f_{jn}^+)\}$$

The degree of similarity between patterns F_i and F_j is estimated based on degrees of overlapping in each feature of the patterns. The degree of similarity of the k^{th} feature of the pattern F_i with respect to the corresponding (k^{th}) feature of the pattern F_j, is given by the relative overlapping between them. The feature intervals $F_{ik} = [f_{ik}^-, f_{ik}^+]$ and $F_{jk} = [f_{jk}^-, f_{jk}^+]$ may or may not overlap. In case of no overlapping, the degree of similarity of the pattern F_i to F_j with respect to the k^{th} feature is said to be zero. In the case of overlapping, there are three possibilities. In the first possibility where the interval F_{ik} contains completely the interval F_{jk}, the similarity of F_i to F_j with respect to k^{th} feature is maximum (value 1), while in the second possibility where the interval F_{ik} is fully contained in the interval F_{jk}, the similarity of F_i to F_j is the ratio of $|F_{ik}|$ to $|F_{jk}|$. In the third possibility the intervals F_{ik} and F_{jk} partially overlap. In this situation, the degree of similarity of F_i to F_j with respect to k^{th} feature is defined to be the ratio of part of F_{ik} overlapping with F_{jk} to $|F_{jk}|$.

Hence, the degree of similarity of pattern F_i to F_j, with respect to the k^{th} feature is given by

$$s_{i\to j}^{k} = \left(\frac{|F_{ik} \bigcap F_{jk}|}{|F_{jk}|} \right) \text{ ....(1)}$$

Thus, the similarity of the pattern F_i to F_j with respect to all n features turned out to be mutlivalued, and is given by

$$S_{i\to j} = [s_{i\to j}^{1}, s_{i\to j}^{2}, s_{i\to j}^{3}, ..., s_{i\to j}^{n}] \text{ .....(2)}$$

This gives the multivalued type representation to the degree of similarity of F_i with respect to F_j.

Similarly, the degree of similarity of the pattern F_j to F_i with respect to the k^{th} feature is estimated as follows:

$$s_{j\to i}^{k} = \left(\frac{|F_{ik} \bigcap F_{jk}|}{|F_{ik}|} \right) \text{ .....(3)}$$

Thus, the similarity of the pattern F_j to F_i with respect to all n features is

$$S_{j\to i} = [s_{j\to i}^{1}, s_{j\to i}^{2}, s_{j\to i}^{3}, ..., s_{j\to i}^{n}] \text{ .....(4)}$$

It can be perceived from the above that the similarity matrix apart from being multivalued, is not necessarily symmetric. Some more details on this similarity measure can be found in (Guru et. al., 2004)

3. The Proposed Feature Selection Scheme

In this section a simple and robust unsupervised feature selection scheme is proposed. The proposed feature selection scheme, in addition to being suitable for symbolic data, does not use the class labels.

The scheme makes use of the multivalued similarity matrix obtained through the similarity measure (section 2) on symbolic patterns. The similarity matrix gives the information of the feature closeness among the patterns. Since each entry in the obtained similarity matrix is multivalued with n components, a single matrix M of size $m^2 \times n$ can be constructed, m being the number of patterns in the data set. The standard deviation in each component in the matrix M is computed and its average is found out by taking the ratio of the sum of the standard deviations to the number of features. Now, the features in the original pattern matrix whose components in the M possess standard deviation less than the average standard deviations are identified and marked as insignificant. It is considered that these components in the similarity matrix are less significant in discriminating the samples and hence the corresponding features in the pattern matrix are marked as the insignificant features. Further, the credence of insignificancy of the marked features is calculated by computing the sum of the correlation possessed by the marked insignificant feature's components in M with other components. The marked features are labeled as insignificant in pattern matrix, if the feature's components in M posses less total correlation than the components of the other features which are not marked as insignificant features. Hence, if all the features marked as insignificant are eliminated then the resulting clusters will be less cohesive. Therefore, the marked insignificant features whose corresponding components posses high total correlation than those of the features which are not marked as insignificant are not labeled as insignificant features. This phase, in addition to being important in identifying the significant features, helps in increasing the intra cluster cohesion. Consequently a dimensionally reduced pattern matrix and correspondingly a recomputed proximity matrix M by applying the similarity measure on the dimensionally reduced pattern matrix are obtained. Algorithmically it can be represented as follows:

Algorithm: Proposed Feature selection scheme

Input: Pattern Matrix M×n where M is the number of samples and n is the dimension of each pattern.

Output: Dimensionally reduced pattern matrix.

Method:

Step1: Apply symbolic similarity measure (Guru et. al., 2004) on the input pattern matrix to obtain the similarity matrix (M), $S_{i\to j} = [s^1_{i\to j}, s^2_{i\to j}, s^3_{i\to j}, ..., s^n_{i\to j}] \forall$ i, j =1,2,…, m of size m × m. where m is the total number of symbolic patterns.

Step2: Identify components in M whose standard deviation is less than the average standard deviation and mark them as insignificant. The average standard deviation is computed by taking the ratio of sum of standard deviations of each component in the matrix M to the number of features.

Step3: Label the marked features as insignificant, if the sum of the correlation possessed by the marked insignificant feature components in M with other components is less than the sum of the correlation possessed by the unmarked feature components in M with other feature components.

Step4: Remove the labeled insignificant features from the pattern matrix to obtain the dimensionally reduced pattern matrix.

Algorithm ends

4. Clustering of Symbolic Data

As the similarity measure (Guru et. al, 2004) is non symmetric and mutlivalued type, the conventional clustering algorithms cannot be applied on the obtained proximity matrix. In view of this, in this section we propose a modified agglomerative clustering technique by using the concept of MSV (Guru et. al, 2004). The MSV between two patterns is defined to be the magnitude of the vector, which is the sum of the scalar times of the vectors representing the degree of similarity between the patterns. Fig.1 gives a pictorial representation of this concept in two-dimensional space

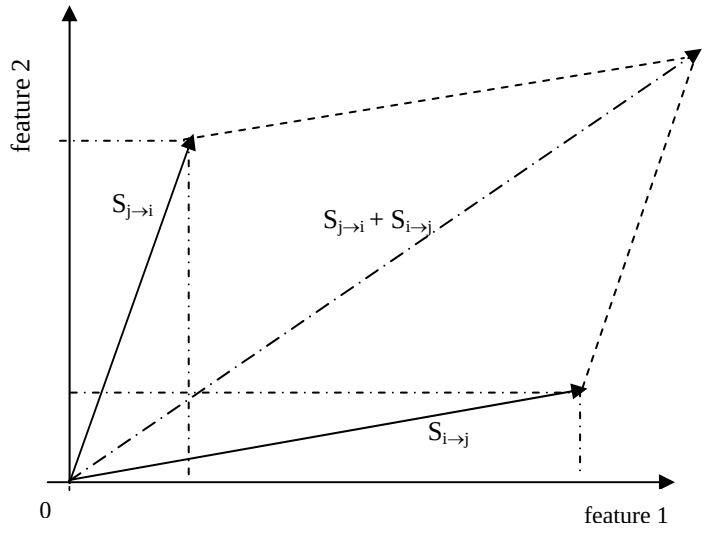


Fig. 1 Computation of MSV

i.e. MSV= $\left|\alpha.S_{i\to j} + \beta.S_{j\to i}\right|$, where α and β are scalars

Since it is an agglomerative clustering, initially m clusters, each consisting an individual pattern, are created, where m is the total number of patterns. Two patterns belonging to two different clusters possessing the maximum MSV are chosen and subsequently the corresponding clusters are merged together into a single cluster. If there are many such pairs of clusters, then they are merged together at the same stage and a composite symbolic pattern which represents the merged patterns is created. A

composite interval for a particular feature is created by representing its minimum value with the least value among that particular feature of the merged components and the maximum value by the maximum among the same feature of the merged patterns. Hence, a composite object is a collection of such features of the merged patterns.

This process of merging is continued until a composite pattern is created by aggregating all symbolic patterns after feature reduction. At each and every stage of merging, say p^{th} stage, with Ct_p number of symbolic patterns, each representing a cluster, it is recommended to compute the similarity matrix, of size $Ct_p \times Ct_p$, of those Ct_p composite symbolic patterns by the use of the similarity measure (section 2). Since each entry in the similarity matrix is multivalued with r components, a single matrix M of size $Ct_p^2 \times r$ can be constructed. The cluster separability factor, sf_p among the Ct_p clusters at the p^{th} stage is computed to be the ratio of the sum of the standard deviations of the values in each component of M to Ct_p. A novel cluster indicator function CI is introduced to identify the actual number of clusters present in the data set as $CI(p) = \frac{|sf_p - sf_{p+1}|}{|sf_p - sf_{p-1}|}$. The number of clusters Ct_p obtained at the stage p for which the CI(p) is local maximum and relatively larger than its neighbors (CI(p-1) and CI(p+1)) is taken as the actual number of clusters.

Thus, the proposed clustering methodology is as trivial as follows.

Algorithm: Clustering of symbolic patterns

Input: The dimensionally reduced similarity matrix $S_{i\to j} = [s^1_{i\to j}, s^2_{i\to j}, s^3_{i\to j}, ..., s^r_{i\to j}]$ $\forall$ i, j =1,2,…, m of size m × m where m is the total number of patterns.

Output: Clusters.

Method:

Step1. Let C = {C_1, C_2,…,C_m}, initially contain m number of clusters each with an individual sample.

Step2. Initialize C = {C_1}; p=0; Ct_p =1; // here p refers stage number and Ct_p refers the number of clusters at p^{th} stage.

Repeat

- Merge two clusters C_p and C_q if there exist two patterns F_i and F_j respectively in C_p and C_q possessing maximum MSV.
- replace the respective composite symbolic pattern by its merged composite symbolic patterns in C
- p = p+1, $Ct_p = Ct_{p-1}+L$
- compute cluster seperability factor sf_p

Until (Single cluster containing all the patterns is obtained).

Step 3. Compute CI (q), $\forall$ q = 1 … p.

Step6. The number of clusters Ct_q obtained at the stage q where CI(q) is local maximum and relatively larger than its neighbors is taken as the actual number of clusters.

Step7. Then the composite objects formed at the stage where CI(q) is local maximum are represented using unique representation scheme.

Algorithm ends

5. Experimental Results

For the purpose of validating the proposed methodologies for their efficacy, we have conducted several experiments on different data sets of type interval and qualitative. The results of only three experiments one on fat oil patterns (Gowda and Diday, (1992); Gowda and Ravi, (1995 (a,b)); Ichino and Yaguchi, (1995)),second on Microcomputer patterns (Gowda and Diday, (1992); Gowda and Ravi, (1995 (a,b)); Ichino and Yaguchi, (1995)) and the third on Microprocessor (Gowda and Ravi, (1995 (a,b)); Ichino and

Yaguchi, (1995)) are presented. Throughout the experimentation, for sake of simplicity the weight factors α and β are set to 1.

The proposed feature selection scheme when applied on the Fatoil data (Table 1), identified the fourth feature (i.e. Saponification value) as insignificant and in Microcomputer data (Table 2) the fifth feature (key value) as insignificant if the clustering is to be done based on the similarity among the patterns. While on Microprocessor data (Table 3) no feature was labeled as insignificant. This means that all the features of microprocessor data have major role to play to increase the intra cluster cohesion of the patterns.

On these pattern matrices which have only significant features, the unsupervised clustering algorithm has been applied. When applied on Fatoil data the algorithm resulted with 2 clusters: {0,1,2,3,4,5}{6,7}, on Microcomputer data the algorithm resulted with 2 clusters: {0,1,2,3,4,5,7,8,9,10,11}{6} and on Microprocessor data resulted with three clusters: {0,1,7} {2,3,6,5,4} {8}.

The superiority of the proposed method can be better understood when it is compared with other methodologies. It can be noticed in Table 4 that the methods (Gowda and Diday, (1992); Ichino and Yaguchi, (1995); Gowda and Ravi, (1995 (b)) group the fat oil patterns into 2 clusters and the method (Gowda and Ravi, (1995 (a))) groups the patterns into 3 clusters based on their own cluster indicator function which acts as a stopping criterion. The proposed clustering technique when applied on the dimensionally reduced fat oil data, through the proposed reduction technique resulted with 2 clusters which are same as that of the methods (Gowda and Diday, (1992); Ichino and Yaguchi, (1995); Gowda and Ravi, (1995 (b))). But on dimensionally reduced microcomputer, the proposed methodology, resulted with 2 clusters which is same as the results obtained by applying Ichino and Yaguchi, (1995), Gowda and Ravi (1995(b)) measures on the microcomputer pattern matrix without dimensionally reduction. However on microprocessor data all the methods, though not exact, result with similar clusters. It should be noted that the results considered of other clustering methodologies are those obtained by their application on non reduced complete pattern matrix. Further, the clusters obtained through the proposed methodology are compared with dendrograms of the cluster formation of the patterns illustrated in Guru et. al.,(2004). It can be noticed in the work (Guru et.al, 2004) that the dendrogram representing the cluster formation of fat oil data is cut at the stage where three clusters are formed, results with clusters (0,1) (2,3,4,5) (6,7) and when the agglomeration is continued to the next stage then the resulting clusters are same as the clusters obtained through the proposed clustering methodology that too with reduced features. Like wise on microcomputer data, the clusters formed in Guru et.al., (2004) are exactly same as the clusters obtained through the proposed methodology. On microprocessor data, in Guru et.al., (2004), if the dendrgoram is cut at the stage where three clusters are formed then the resulting clusters: (0,1,4,7) (2,3,5,6) (8), are similar to the one obtained by the proposed clustering methodology.

Hence, it is obvious that the proposed feature selection scheme and the unsupervised classification, along with the similarity measure are more superior, robust and efficient than the other available methods.

6. Conclusion

In this paper, a simple, robust and an unsupervised feature selection scheme for clustering of symbolic data is proposed. In addition, a modified agglomerative method by introducing the concept of Mutual Similarity Value (MSV) for clustering of symbolic patterns is also presented. The validness of the proposed selection scheme is shown through experimentation.

References:

Bock H.H and Diday E.(Eds.), 2000. “Analysis of symbolic data”. Springer Verlag publication.

Chouakria A., Diday E and P.Cazes (1995). Extension of the principal component analysis to interval data. Proceedings of New Techniques and Technologies for Statistics, Bonn 1995.

Denoeux T. and Masson M., 2000. “Multidimensional scaling of interval valued dissimilarity data”. Pattern Recognition Letters 21, pp 83-92.

Gowda K.C and Diday E., 1992. “Symbolic clustering using a new similarity measure”. IEEE Trans SMC Vol. 22, No 2, pp 368-378.

Gowda K.C. and Ravi T.V., 1995 (a). “Agglomerative Clustering of Symbolic Objects using the concepts of both dissimilarity and dissimilarity”. Pattern Recognition Letters 16, pp 647-652.

Gowda K.C. and Ravi T.V., 1995 (b). “Divisive Clustering of Symbolic Objects using the concepts of both dissimilarity and dissimilarity”. Pattern Recognition Vol 28, No 8, pp 1277-1282.

Guru D.S , Kiranagi B. B. and Nagabhushan P, 2004. “Multivalued type proximity measure and concept of mutual similarity value for clustering symbolic patterns”. Journal of Pattern Recognition Letters, Vol 25 (10), pp 1203-1213.

Hartigan J.A., (1975). “Clustering algorithms”. Wiley New York.

Ichino M and Yaguchi H., 1994. “Generalized Minkowski metrices for mixed feature type data analysis”. IEEE Trans on system, man and cybernetics Vol 24, No 4, April.

Jain A.K, Murty M.N and Flynn P.J., 1999. “Data Clustering: A Review”. ACM computing Surveys, Vol 31, No 3, Sept. pp 264-324.

Liu Huan and Yu Lei, 2005. “Towards integrating feature selection algorithms for classification and clustering”. IEEE Transactions on Knowledge and Data Engineering. Vol. 17, No.4, pp 491-502.

Liu Huan, Edward R. Dougherty, Jennifer G. Dy, Kari Torkkola, Eugene Tuv, Hanchuan Peng, Chris Ding, Fuhui Long, Michael Berens, Lance Parsons, Zheng Zhao, Lei Yu and George Forman,, 2005. “Evolving Feature selection”. IEEE Transactions on Intelligent Systems, Vol 20, No. 6, pp 64-76.

Nagabhushan P., Gowda K.C and Diday E., 1995. “Dimensionality reduction of symbolic data”. Journal of Pattern Recognition Letters, 16(2), pp 219-223.

Prakash S.H.N., 1998. “Classification of remotely sensed data: some new approaches”. Ph.D. Thesis. University of Mysore. Mysore, India.

Wolf L. and Bileschi S.M, 2005. “Combining variable selection with dimensionality reduction”. The proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 801-806.

Table1. Fat oil data

Pattern No.	Sample	Specific gravity	freezing pt	Iodine value	Saponification value	Major fatty acids
0	Linseed oil	[0.930,0.935]	[-27.0, -18.0]	[170.0,204.0]	[118.0,196.0]	L, Ln, O, P, M
1	Perilla oil	[0.930,0.937]	[-5.0, -4.0]	[192.0,208.0]	[188.0,197.0]	L, Ln, O, P, S
2	Cottonseed oil	[0.916,0.918]	[-6.0 , -1.0]	[99.0,113.0]	[189.0,198.0]	L, O, P, M, S
3	Sesame oil	[0.920,0.926]	[-6.0, -4.0]	[104.0,116.0]	[187.0,193.0]	L, O, P, S, A
4	Camellia oil	[0.916,0.917]	[-21.0, -15.0]	[80.0, 82.0]	[189.0,193.0]	L, O
5	Olive oil	[0.914,0.919]	[0.0, 6.0]	[79.0, 90.0]	[187.0,196.0]	L, O, P, S
6	Beef tallow	[0.860,0.870]	[30.0,38.0]	[40.0, 48.0]	[190.0,199.0]	O, P, M, S, C
7	Hog fat	[0.858,0.864]	[22.0,32.0]	[53.0, 77.0]	[190.0,202.0]	L, O, P, M, S, LU

Table 2. Microcomputer data

Pattern No.	Sample	Display	RAM	ROM	Microprocessor	Keys
0	Apple II	Color	48	10	6502	52
1	Atari 800	Color	48	10	6502	57-63
2	Vic 20	Color	32	11-16	6502A	64-73
3	Sorcerer	B/W	48	4	Z80	57-63
4	Zenith H8	Built in	64	1	8080A	64-73
5	Zenith H 89	Built in	64	8	Z80	64-73
6	HP 85	Built in	32	80	HP	92
7	Horizon	Terminal	64	8	Z80	57-63
8	Challenger	B/W	32	10	6502	53-56
9	OhioSci.11	B/W	48	10	6502C	53-56
10	TRS-80I	B/W	48	12	Z80	53-56
11	TRS-80 III	Built in	48	14	Z80	64-73

Table3. Microprocessor data

Pattern No	MPU	Clock (MHz)	General Registers	Instructions (bytes)	Cache Size	Cache Type
0	i386DX	[16.0, 33.0]	8.0	123.0	-	NULL
1	i386SX	[12.0, 20.0]	8.0	123.0	-	NULL
2	i486SX	[25.0, 50.0]	8.0	214.0	8192.0	Common
3	I48SX	[20.0, 20.0]	8.0	129.0	8192.0	Common
4	68020	[12.0, 33.0]	8.0	99.0	256.0	Instruction
5	68030	[16.0, 50.0]	8.0	105.0	512.0	Independent
6	6840	[25.0, 25.0]	8.0	140.0	8192.0	Independent
7	MB86901	[20.0, 25.0]	120.0	64.0	-	Null
8	MB86930	[20.0, 40.0]	136.0	68.0	4096.0	Independent

Table 4.Results based comparison

Methodology	**Fat oil**	**Microcomputer**	**microprocessor**
	Description of the clusters	**Description of the clusters**	**Description of the clusters**
Ichino and Yaguchi (1994)	{0,1,2,3,4,5}{6,7}	{0,1,2,3,4,5,7,8,910,11}{6}	(0,1,4,5)(2,3,6)(7,8)
Gowda and Diday (1992)	{0,1,2,3,4,5}{6,7}	{0,1,9,10}{6}{2,8}{3,4,5,117}	Not available
Gowda and Ravi (1995(a))	{0,1}{ 2,3,4,5}{6,7}	{0,1,3,5,7,8,9,10,11}{2}{6}{4}	(0,1,4,5,7) (2,3,6) (8)
Gowda and Ravi (1995(b))	{0,1,2,3,4,5}{6,7}	{0,1,2,3,4,5,7,8,910,11}{6}	(0,1, 2,3,4,5,6) (7,8)
Guru et.al (2004)	{0,1}{ 2,3,4,5}{6,7}	{0,1,2,3,4,5,7,8,910,11}{6}	(0,1,4,7) (2,3,5,6) (8)
Proposed methodology	**{0,1,2,3,4,5} {6,7}**	**{0,1,2,3,4,5,7,8,910,11} {6}**	**(0,1,7) (2,3,6,5,4) (8)**

Aiming for Parsimony in the Sequential Analysis of Activity-Diary Data

Elke Moons and Geert Wets
University of Hasselt
Transportation Research Institute
Wetenschapspark 5 bus 6
3590 Diepenbeek
Belgium
Tel. +32-11-26.91.26 – +32-11-26.91.58
Email: elke.moons@uhasselt.be – geert.wets@uhasselt.be

Abstract

This paper aims at a better understanding in the impact of simplification in a sequential analysis of activity-diary data using a feature selection method. To this effect, the predictive performance of the Albatross model, which incorporates nine different facets of activity-travel behaviour, based on the original full decision trees is compared with the performance of the model based on trimmed decision trees. The more parsimonious models are derived by first using a feature selection method to determine the irrelevant variables which are then left out of the further model building process. The results indicate that significantly smaller decision trees can be used for modelling the different choice facets of the sequential system without loosing much too much in predictive power. The performance of the models is compared at two levels: the choice facet level, at which we compare the performance of each facet separately and the trip level, comparing the correlation coefficients that determine the strength of the associations between the observed and the predicted origin-destination matrices. The results indicate that the model based on the trimmed decision trees predicts activity diary schedules with a minimum loss of accuracy at the choice facet level. Moreover, the results show a slightly better performance at the trip matrix level.

Keywords: activity-travel behaviour, parsimony, sequential analysis, feature selection, decision trees

1. Introduction

In the past few years, activity-based forecasting of travel demand has become a major field of interest in transportation research. The aim of activity-based models is to predict which activities will be conducted where, when, for how long, with whom and with which transport mode. Rule-based models have proven to be very flexible when compared to utility-maximising models (Arentze *et al.*, 2001) and they also perform well in predicting transport choice behaviour if an induction technique is used (Wets *et al.*, 2000). Although these rule-based models perform very well, they also show some limitations. Most of them are based on quite complex rule sets. However, already in the Middle Ages, there was a call for simplicity: William of Occam's razor states that `Nunquam ponenda est pluralitas sin necesitate', meaning `Entities should not be multiplied beyond necessity' (Tornay, 1938). It was born in the Middle Ages as a criticism of scholastic philosophy, whose theories grew ever more elaborate without any corresponding improvement in predictive power. In the intervening centuries it has come to be seen as one of the fundamental tenets of modern science and today it is often invoked by learning theorists as a justification for preferring simpler models over more complex ones. However, Domingos (1998) learned us that it is tricky to interpret Occam's razor in the right way. The interpretation "Simplicity is a goal in itself" is essentially correct, while "Simplicity leads to greater accuracy" is not.

While a larger number of rules may be valuable when one wishes to better understand the data, from a predictive perspective a large number of rules may imply that the decision tree induction algorithm has over-fitted the data. The obtained decision tree structure may then be very unstable and sensitive to highly correlated covariates.

Feature selection offers a solution to reduce the number of irrelevant attributes and as a consequence often the size of the decision tree will also be reduced. The key notion underlying feature selection is that the number of decision rules is reduced by selecting and deleting irrelevant features, based on some statistical measure. The impact of feature selection on the predictive performance of rule-based models is however not a priori clear. On the one hand, because the irrelevant variables are deleted, feature selection may not have a substantial negative effect on predictive performance. However, a smaller decision tree may also result in a higher probability of misclassification, leading to worse predictive performance. It is against this background that this paper reports the findings of a methodological study that was conducted to gain a better understanding of the influence of a smaller set of decision rules on the predictive performance of a sequential models of activity scheduling behaviour, the Albatross model. Moons *et al.* (2002) investigated the influence of irrelevant attributes on the performance of the decision tree for the transport mode, the travel party, the activity duration and the location agent of the Albatross model system. We found that the use of considerable less decision rules did not result in a significant drop in predictive performance compared to the original larger set of rules that was derived from the activity-travel diaries. In this paper, the question `To what extent can this result be generalised to the complete Albatross model system (represented by nine different choice facets)?' is inspected.

In order to be able to look at the results in the right context, we will first shortly describe the Albatross system in the next section, followed by a brief introduction to the different methods used to perform the analysis. Then, feature selection is applied to decision rule induction and the results will be discussed in Section 3. The predictive performance will be evaluated on each facet separately (by means of the accuracy) and at trip matrix level where the correlation coefficients that determine the strength of the associations between the observed and predicted origin-destination matrices are judged against each other. Conclusions are drawn in the final section.

2. Methods

2.1 The Albatross system

The Albatross model was developed for the Dutch Ministry of Transportation (Arentze and Timmermans, 2000). In this study, we used the data that were used to find the set of rules for the original model.

This rule-based model relies on a set of Boolean decision rules that are used to predict activity-travel patterns. These rules were extracted from activity-diary data. The activity scheduling process is sequential in nature. Figure 1 provides a schematic representation of the Albatross scheduling model.

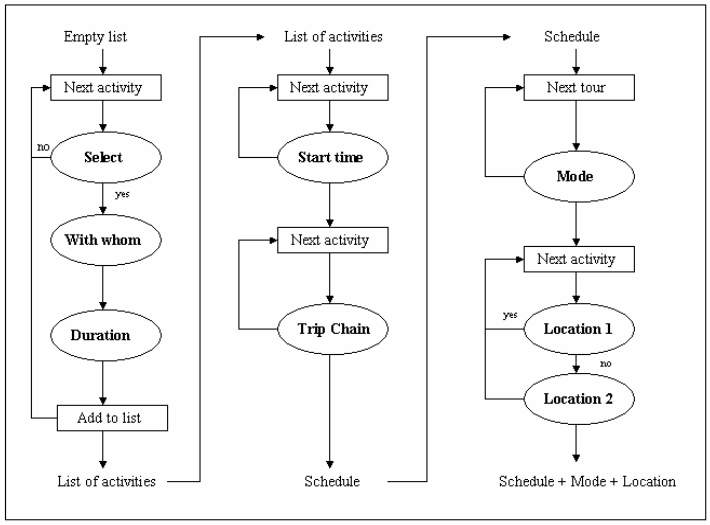


Figure 1: Albatross scheduling engine

The activity scheduling agent of Albatross is based on an assumed sequential execution of decision trees to predict activity-travel patterns. Before the sequential execution starts, the main transport mode (i.e. mode for work, referred to as mode 1) will be predicted. The model next executes a set of decision rules to predict which activity will be inserted in the schedule. It then determines, based on another sets of rules, with whom the activity is conducted and the duration of the activity. The order in which activities are evaluated is pre-defined as: daily shopping, services, non-daily shopping, social and leisure activities. The assignment of a scheduling position to each selected activity is the result of the next two steps. After a start time interval is selected for an activity, trip-chaining decisions determine for each activity whether the activity is to be connected with a previous and/or next activity. Those trip chaining decisions are not only important for timing activities but also for organizing trips into tours. The next step involves the choice of transport mode for other purposes (referred to as mode2) and the choice of location. Possible interactions between mode and location choices are taken into account by using location information as conditions of mode selection rules.

2.2 Decision Tree Induction: C4.5

Decision tree induction can be best understood as being similar to parameter estimation methods in econometric models. The goal of tree induction is to find the set of Boolean rules that best represents the empirical data. The original Albatross system was derived using a Chi-square based approach (Moons, 2005). In this paper, however, the trees were re-induced using the C4.5 method (Quinlan, 1993) because this method can be easily combined with the Relief-F feature selection. Arentze *et al.* (2000) found approximately equal performance in terms of goodness-of-fit of the two methods in a representative case study. The C4.5 algorithm works as follows. Given a set of I observations taken from activity-travel diary data, consider their values on n different explanatory variables or attributes $x_{i1}, x_{i2}, \ldots, x_{in}$ and on the response variable $y_i \in \{1,2,\ldots,p\}$ for $i = 1,\ldots, I$. Starting from the root node, each node will be split subsequently into internal or terminal nodes. A leaf node is terminal when it has no offspring nodes. An internal node is split by considering all allowable splits for all variables and the best split is the one with the most homogeneous daughter nodes. The C4.5 algorithm recursively splits the sample space on X into increasingly homogeneous partitions in terms of the response variable Y, until the leaf nodes contain only cases from a single response class. Increase in homogeneity achieved by a candidate split is measured in terms of an information gain ratio. After building the tree, pruning strategies are adopted. This means that the decision tree is simplified by

discarding one or more sub-branches and replacing them with leaves. For a detailed description, we refer to Wets *et al.*, 2000.

2.3 Feature Selection: Relief-F

Feature or variable selection strategies are often implied to explore the effect of irrelevant attributes on the performance of classifier systems. One can distinguish between two types of feature selection approaches: the filter and the wrapper approach. Both methods have been compared extensively (Hall, 1999a, 1999b; Koller and Sahami, 1996). In this analysis, the filter approach, more specifically the Relief-F feature selection method, is opted for since it can handle multiple classes of the dependent variable (the nine different choice facets that we are predicting range from two to seven classes) and above that it is easily combined with the C4.5 induction algorithm.

Feature selection strategies can be regarded as one way of coping with the correlation between the attributes. This is relevant because the structure of trees is sensitive to the problem of multi-collinearity, which implies that some variables would be redundant (given the presence of other variables). Redundant variables do not affect the impacts of the remaining variables in the tree model, but it would simply be better if they were not used for splitting. Therefore, a good feature selection method for this analysis would search for a subset of relevant features that are highly correlated with the class variable that the tree-induction algorithm is trying to predict, while mutually having the lowest possible correlations.

Relief (Kira and Rendall, 1992), the predecessor of Relief-F, is a distance-based feature weighting algorithm. It imposes a ranking on features by assigning each a weight. Features with the highest weights are considered to be the most relevant, while those with values close to zero or with negative values are judged irrelevant. The weight for a particular feature reflects its relevance in distinguishing the classes. In determining the weights, the concepts of *near-hit* and *near-miss* are central. A *near-hit* of instance *i* is defined as the instance that is closest to *i* (based on Euclidean distance between two instances in the *n*-dimensional variable space) and which is of the same class (concerning the output variable), while a *near-miss* of *i* is defined as the instance that is closest to *i* and which is of a different class. The algorithm initially assigns the value zero to each attribute, and this will be adapted with each run through the instances of the data set. It attempts to approximate the following difference of probabilities for the weight of a feature *X*:

$$W_X = P(\text{different value of X} \mid \text{nearest instance of different class}) - P(\text{different value of X} \mid \text{nearest instance of same class}).$$

So, Relief works by random sampling an instance and locating its nearest neighbour from the same and opposite response class. By removing the context sensitivity provided by the "nearest instance" condition, attributes are treated as mutually independent, and the previous equation becomes:

$$\text{Relief}_X = P(\text{different value of X} \mid \text{different class}) - P(\text{different value of X} \mid \text{same class}).$$

Relief-F (Kononenko, 1994) is an extension of Relief that can handle multiple classes and noise caused by missing values, outliers, etc. To increase the reliability of Relief's weight estimation, Relief-F finds the *k* nearest hits and misses for a given instance, where *k* is a parameter that can be specified by the user. For multiple class problems, Relief-F searches for nearest misses from each different class (with respect to the given instance) and averages their contribution. The average is weighted by the prior probability of each class.

3. Analysis and Results

The overall aim of this study is to investigate whether a simplification of the rule sets underlying the Albatross model leads to a significant loss in predictive power. This simplification will be obtained by reducing the set of decision rules through the application of a feature selection method. The original Albatross model consists

of nine choice facets. For each of these choice facets, a set of decision rules was extracted from activity-travel diaries. To predict activity-travel patterns, these decision trees are executed sequentially in the Albatross system according to some scheduling process model (Arentze and Timmermans, 2000). We will investigate the effect of simpler rules for each choice facet.

3.1 The Data

The analyses are based on the activity diary data used to derive the original Albatross system. The data were collected in February 1997 for a random sample of 1649 respondents in the municipalities of Hendrik-Ido-Ambacht and Zwijndrecht (South Rotterdam region) in the Netherlands.
The activity diary asked respondents, for each successive activity, to provide information about the nature of the activity, the day, start and end time, the location where the activity took place, the transport mode (chain) and the travel time per mode, if relevant, accompanying individuals, and whether the activity was planned. Open time intervals were used to report the start and end times of activities. A pre-coded scheme was used for activity reporting. More details can be found in Arentze and Timmermans (2000).

3.2 Study Design

The original data set is split into two subsets. A training set, containing the first 75% of the cases, on which the different models will be built and optimised. The remaining 25% of the cases make up the validation or test set that can be used to compute the accuracies (percentage of correctly classified instances), etc. These percentages are arbitrary but are common practice in validation studies (see e.g. Wets *et al.*, 2000).

We will first build decision trees for each of the nine choice facets, using the C4.5 algorithm (Quinlan, 1993). This approach will be called the full approach. The C4.5 trees were induced based on one simple restriction: the final number of cases in a leaf node must meet a minimum. For eight out of the nine choice facets, this minimum was set to 15 (except for the very large data set of the `select'-dimension, where this number was set to 30). In a second approach, the feature selection approach, we will first identify the relevant attributes for each of the nine choice facets separately, based on the Relief-F feature selection method with the k parameter set equal to 10. Next, the C4.5 trees were built based on the same restriction as in the full approach, though only the remaining relevant attributes were used. To determine the selection of variables, the following procedure was adopted. Several decision trees were built, each time removing one more irrelevant attribute, as they appeared lowest in the ranking that has been provided by the FS method. For each of these decision trees, the accuracy was calculated and compared to the accuracy of the decision tree of the full approach. The smallest decision tree, which resulted in a maximum decrease of 2% in accuracy compared to the decision tree including all features, was chosen as the final model for a single choice facet in the feature selection approach. This strategy was applied to all nine dimensions of the Albatross model.

3.3 Results

At first, we will take a closer look at the average length of the observed and predicted sequences of activities. In the observed patterns, the average number of activities equals 5.160 for the training set and 5.155 for the test set. This average length offers room for 1-3 flexible activities complemented with 2-4 in-home activities. Considerable variation occurs, however, as indicated by the standard deviation of approximately 3 activities.

Method	Training set	Test set
Full approach	5.286	5.286
	(2.953)	(2.937)
FS approach	5.014	4.907
	(3.033)	(2.921)

Table 1: Average number of predicted activities in sequences (standard deviation between brackets)

We observe in Table 1 that in general the full approach predicts activity sequences that are somewhat too long, while those of the feature selection approach are rather a little bit too short.

The results of these different methods will now be compared at two levels of aggregation: the choice facet level and the trip matrix level. At the choice facet level, we will discuss the number of attributes that remained in the final decision tree model for each of the two approaches and the probability of a correct prediction for each decision tree. At the trip matrix level, correlation coefficients are calculated to measure the degree of correspondence between the observed and the predicted Origin-Destination matrices.

3.3.1 Choice Facet Level

Tables 2 and 3 provide the results of the analyses conducted to assess model performance at the choice facet level. The first column of these tables presents the nine choice facets of Albatross. The second column lists the levels of the Y-variable, while the third column gives the total number of attributes that were considered to build the final decision tree. The fourth column depicts the size of the decision tree. Column five reports the probability of a correct prediction and in the last column a measure of relative performance, where the probability of a correct prediction is compared to the probability of a correct prediction under a null model. This null model assigns a new case to a category of the Y-variable with a probability, equal to the number of observed cases in the category divided by the total number of cases in the data set.

Decision tree	# alts	# attrs	# leafs	E	e_{ratio}
Mode for work	4	32	8	0.598	0.155
Selection	2	40	35	0.686	0.052
With-whom	3	39	72	0.499	0.223
Duration	3	41	148	0.431	0.145
Start time	6	63	121	0.408	0.285
Trip chain	4	53	8	0.802	0.576
Mode other	4	35	63	0.524	0.222
Location 1	7	28	30	0.540	0.264
Location 2	6	28	47	0.372	0.214

Table 2: Model performance: choice facet level (full approach)

Decision tree	# alts	# attrs	# leafs	E	e_{ratio}
Mode for work	4	2	6	0.595	0.147
Selection	2	1	1	0.669	0.000
With-whom	3	4	51	0.467	0.173
Duration	3	4	38	0.368	0.051
Start time	6	8	1	0.172	0.000
Trip chain	4	10	13	0.811	0.596
Mode other	4	11	60	0.508	0.196
Location 1	7	6	15	0.513	0.222
Location 2	6	8	14	0.312	0.141

Table 3: Model performance: choice facet level (FS approach)

The results of the previous analyses show that, in general, the full approach outperforms the FS approach on the dimensions separately. On the other hand, feature selection generally generates considerably less complex decision trees than the full approach. One exception is the 'trip chaining' choice facet, which more leafs in the final tree with FS than in the tree without feature selection. A logical consequence of this result is that the measure of relative performance of the models with FS is somewhat smaller.
The most important variables for both approaches do not differ that much, but if differences can be discerned, they can then often be explained by high correlations between variables.

3.3.2 Trip Matrix Level

At trip matrix level, we compare the number of trips made from a certain origin to a certain destination. Correlations were calculated between observed and predicted matrix entries in general and for trip matrices

that are disaggregated on transport mode. The variation of the correlation coefficient can be largely explained by the variation in the number of cells between matrices. The general OD matrix has 400 cells (20 origins and 20 destinations) and the OD matrix by mode 2000. As could be expected, the fit decreases with an increasing number of cells.

Matrix	ρ (o, p) (Full approach)	ρ (o, p) (FS approach)
None (train)	0.962	0.957
Mode (train)	0.885	0.887
None (test)	0.942	0.947
Mode (test)	0.856	0.849

Table 4: Model performance: trip matrix level

In Table 4 the performance of the different models on the training and the test data set is given. The results indicate that all correlation coefficients are similar. Both approaches perform equally well and if there is a difference it does not exceed the 1% level.

4. Conclusion

In the last decade, rule-based models that predict travel behaviour based on activity diary data have been suggested in the literature. These models usually perform very well, though, very often, they are based on a very complex set of rules. Moreover, research in the field of psychology (Gigerenzer *et al.* 1999) has learned us that simple models often predict human behaviour very well. In fact, the call for simplicity is a question of all ages. Occam's razor, that has to be situated already in the Middle Ages, being an important example. It is in this light that this paper should be regarded. We tried to simplify the complex set of rules used to determine the Albatross system by performing two similar analyses: one with and one without irrelevant variables, while in the second analyses, at same time we cut back in the number of variables. The results of the tree-induction algorithms can namely be heavily influenced by the inclusion of irrelevant attributes. On the one hand, this may lead to over-fitting, while one the other hand, it is not evident whether the inclusion of irrelevant attributes would lead to a substantial loss in accuracy and/or predictive performance. The aim of this study reported therefore was to further explore this issue in the context of the Albatross model system.

The results show that the models that make up their decisions based on one or a few variables are not in any case second to the complex analysis. This comes as a welcome bonus. In fact, more or less the same results were obtained at the trip matrix level. At the choice facet level, one can observe that a strong reduction in the size of the trees as well as in the number of predictors is possible without adversely affecting predictive performance too much. Thus, at least in this study, there is no evidence of substantial loss in predictive power in the sequential use of decision trees to predict activity-travel patterns.
The results indicate that using feature selection in a step prior to tree induction can improve the performance of the resulting model. It should be noted, however, that predictive performance and simplicity are not the only criteria. The most important criterion is that the model needs to be responsive to policy sensitive attributes and for that reason policy sensitive attributes, such as for example service level of the transport system, should have a high priority in the selection of attributes if the model is to be used for predicting the impact of policies. The feature selection method allows one to identify and next eliminate correlated factors that prevent the selection of the attributes of interest during the construction of the tree, so that the resulting model will be more robust to policy measures.

These findings endorse the primary belief that people, because of their limitations in knowledge and time, rely for their choices on some simple heuristics. Since, in the Albatross system, we are trying to predict nine different choices on travel behaviour made by human beings, this might give an idea on why these simple models do not necessarily perform worse than the complex models. However, if simple models are able to predict the choices of a human being, this can mean two things: either the environment itself is perceived as simple, or the complex choice process can be described by simple models. Since activity-based transport

modellers keep developing systems with an increasing complexity in order to try to understand the travel behaviour undertaken by humans, we acknowledge that the environment is not simple. However, whether it is perceived as simple by human beings, remains an open question.

5. References

Arentze, T. A. and Timmermans, H. J. P. (2000) *Albatross: A Learning-Based Transportation Oriented Simulation System* Eindhoven University of Technology, EIRASS.

Arentze, T., Hofman, F., van Mourik, H., Timmermans, H. and Wets, G. (2000), Using decision tree induction systems for modeling space-time behavior, *Geographical Analysis,* **32,** pp. 330-350.

Arentze, T. A., Borgers, A. W. J., Hofman, F., Fujii, S., Joh, C.-H., Kikuchi, A., Kitamura, R., Timmermans, H. J. P. and van der Waerden, J. (2001) Rule-based versus utility-maximizing models of activity-travel patterns: a comparison of empirical performance. In *Travel Behaviour Research: The Leading Edge* Ed. D A Hensher, Pergamon Press, Oxford, pp. 569-584.

Domingos, P. (1998) Occam's two razors: The sharp and the blunt. Proceedings of the fourth international conference on knowledge discovery and data mining, pp. 37-43.

Gigerenzer, G., Todd, P.M. and the ABC Research Group. (1999) *Simple Heuristics That Make Us Smart*. Oxford University Press, New York.

Hall, M. A. (1999a) *Correlation-based Feature Selection for Machine Learning,* Ph.D. dissertation, University of Waikato.

Hall, M. A. (1999b) Feature selection for Machine Learning: Comparing a correlation-based filter approach to the wrapper. Proceedings of the Florida Artifical Intelligence Symposium (FLAIRS), Orlando, Florida.

Kira, K. and Rendall, L. A. (1992) A practical approach to feature selection. Proceedings of the Ninth International Conference on Machine Learning, Aberdeen, Scotland, UK, Morgan Kaufmann Publishers, San Mateo, pp 249-256.

Koller, D. and Sahami, M. (1996) Toward optimal feature selection. Proceedings of the 13th International Conference on Machine Learning, Bari, Italy, pp 284-292.

Kononenko, I. (1994) Estimating attributes: analysis and extensions of relief. Proceedings of the 7th European Conference on Machine Learning, Catania, Italy, Springer Verlag, pp 171-182.

Moons, E., Wets, G., Aerts, M., Vanhoof, K., Arentze, T. A. and Timmermans, H. J. P. (2002) The impact of irrelevant attributes on the performance of classifier systems in generating activity schedules. Proceedings of the 81st Annual Meeting of the Transportation Research Board, Washington, D.C.

Moons, E. (2005) *Modelling Activity-Diary Data: Complexity or Parsimony?* Ph. D. dissertation, Limburgs Universitair Centrum, Diepenbeek, Belgium.

Quinlan, J. R. (1993) *C4.5 Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo.

Tornay, S. (1938) *Ockham: Studies and selections*. La Salle, IL: Open Court..

Wets, G., Vanhoof, K., Arentze, T. A., Timmermans, H. J. P. (2000) Identifying Decision Structures Underlying Activity Patterns: An Exploration of Data Mining Algorithms *Transportation Research Record,* **1718**, pp. 1-9.

An Ensemble of Anomaly Classifiers for Identifying Cyber Attacks*

Carlos Kelly† Diana Spears‡ Christer Karlsson§ Peter Polyakov¶

Abstract

A novel approach is presented that bridges the gap between anomaly and misuse detection for identifying cyber attacks. The approach consists of an ensemble of classifiers that, together, produce a more informative output regarding the class of attack than any of the classifiers alone. Each classifier classifies based on a limited subset of possible features of network packets. The ensemble classifies based on the union of the subsets of features. Thus it can detect a wider range of attacks. In addition, the ensemble can determine the probability of the type of attack based on the results of the classifiers. Experimental results demonstrate an increase in the rate of detecting attacks as well as accurately determining their type.

Keywords: intrusion detection, classifier ensemble

1 Problem Description.

In our current information age, and with the timely issue of national security, network security is an especially pertinent topic. One important aspect of computer network security is *network intrusion detection*, i.e., the detection of malicious traffic on a computer network.

There are two main approaches to designing Network Intrusion Detection Systems (NIDS): *anomaly detection* and *misuse detection*. Both are essentially classifiers, i.e., they label incoming network packets as "attack," "non-attack," and if an attack perhaps what type of attack. Anomaly and misuse detection are complementary approaches to intrusion detection. Anomaly detection consists of building a model of normal computer usage, and tagging outliers as "anomalies." Such systems are typically computationally efficient, but only yield a binary classification – "attack" or "non-attack" [5, 12]. Misuse detection systems match potential attacks (e.g., network packets) against a database of known attacks (called *signatures*). If there is a match, then the data (packet) is labeled an "attack," and the class of attack is considered to be the same as that of the matching signature. Unfortunately, misuse detection systems tend to be slow, especially if their database of signatures is large [10].

The main thrust of our research is to bridge the gap between anomaly and misuse detection. Anomaly NIDS classify packets quickly in comparison to misuse based NIDS, but without as much information about the attack. We have created an ensemble of anomaly-based NIDS that refines the binary classification of each ensemble member and yields more detailed classification information than each member alone. Therefore, if anomaly detection precedes misuse detection, then our system will partially refine the output of anomaly detection, thereby accelerating the processing of the misuse detection system. The pipeline can be summarized as: (1) anomaly detection, (2) refinement by ensemble, and (3) misuse detection. If the computational cost of the second step, refinement by ensemble, is lower than the computational benefit that it yields by shortening the run-time of the misuse detection system, then our approach will be beneficial overall. Whether this is the case, depends on the size of the database of signatures that one maintains. Over time, as people (and systems) increase their knowledge base of attacks, we expect our approach to become increasingly more useful.

This paper describes only steps (1) and (2) of the pipeline above. Step (3) will be addressed as part of future work. In the remainder of this paper, we describe our ensemble approach, as well as experimental evaluation results that show its effectiveness for intrusion detection. Here, it is assumed that the data consists of Transmission Control Protocol (TCP) packets, sent over the network. The data we used is from the DARPA/MIT Lincoln Laboratory database (see http://www.ll.mit.edu/IST/ideval/index.html).

*Supported by the ONR URI grant "Anomaly and misuse detection in network traffic streams," subcontract PENUNV48725.

†Mathematics Department, University of Wyoming.

‡Computer Science Department, University of Wyoming.

§Computer Science Department, University of Wyoming.

¶Mathematics Department, University of Wyoming.

2 A Novel Ensemble Approach.

An *ensemble* of classifiers is a collection of classifiers that are combined into a single classifier. Most of the research conducted on classifier ensembles assumes homogeneous ensembles of binary classifiers, and assumes that the ensemble also outputs a binary classification. The purpose of such ensembles is to increase classification accuracy, e.g., with voting, *bagging*, or *boosting* [1]. One notable exception is the *stacked generalization* approach of Wolpert [13]. Stacked generalization assumes a heterogeneous ensemble of different classifiers, each with its own "area of expertise." Nevertheless, the purpose of stacked generalization is also to increase classification accuracy, without changing the set of classes.

To the best of our knowledge, *our approach to ensembles is the first to utilize a heterogeneous set of classifiers for the purpose of increasing information (refined classification), rather than classification accuracy.* Specifically, our approach takes a suite of classifiers (currently two), each of which outputs a binary classification, and combines them to output a probability distribution over *seven* classes. We expect the ensemble output to be increasingly more informative as the number of classifier components is increased beyond two. Furthermore, if the classifiers run on the data in parallel, adding more classifiers to the ensemble would not increase the overall time to apply the method, i.e., it is highly scalable. However, this is our first investigation into such an ensemble, so we begin with two classifiers.

The essence of our approach is to empirically build an *ensemble probability classification matrix*, abbreviated as *EPCM*, based on system performance on test data. In other words, in machine learning one typically trains the system on training data, and then tests its performance on test data. We instead partition the data into three sets: the training data, the testing data, and the validation data. Each individual classifier is trained separately on the training data. Note that the training data is attack-free – because anomaly detection systems learn models of normal (friendly) user data, and then use these models to detect anomalies, which are labeled "attacks." After training, each system has a hypothesis regarding the nature of "non-attack" data. These hypotheses are applied to the testing data, to make predictions regarding whether each system thinks each packet is an "attack" or a "non-attack." We also use the known information (from the DARPA website) on the test data regarding whether each packet is an attack or not, and if it is an attack then what class of attack (from the seven known classes). All of this information is automatically combined into an EPCM, which predicts a probability distribution over the seven classes, based on the outputs of the systems in the ensemble and the true classes of the packets (as defined by DARPA/MIT). The last step is to test the performance of the ensemble on a set of validation data, for which there is no advance knowledge given to the system regarding the (true) data classification.

Why do we expect our novel approach to work? The key to our success is the notion of *inductive bias*, or simply *bias*. Mitchell defines *bias* as "any basis for choosing one generalization over another, other than strict consistency with the observed training instances" [9]. The hypotheses output by our classifiers are special instances of what Mitchell calls "generalizations." An example of a bias is the choice of what attributes the classifier system considers. For instance, one system might only look at the header information in a packet when classifying the packet as a type of attack, whereas another system might only look at the packet payload. Certainly the choice of attributes will affect the types of attacks that the system is able to identify. One system might be able to detect some classes of attacks; another system might be able to detect other classes. In general, the classes of attacks detectable by two systems could be disjoint or overlap. By combining two systems with very different biases, we increase the set of detectable attacks. Furthermore, by exploiting known differences in system biases, we can further refine our classification knowledge. For example, if one system says "attack" and the other says "non-attack," then this combined information can tell us (with high probability) what *kind* of attack it is most likely to be. To better understand the synergistic effects of combining the information, we formalized the biases of each of the two systems. From this formalization, one can understand the classes of attacks for which each system is best suited to identify. This is the essential rationale behind our ensemble approach.

3 Ensemble Components.

Our ensemble is composed of two anomaly NIDS, LERAD [5] and PAYL [12]. LERAD's hypotheses are rule sets of expected (normal) user behavior, and PAYL's hypotheses are byte distributions derived from normal packets. Each of these systems is described in greater detail, below.

Some preprocessing of the raw network dump data was necessary for LERAD and PAYL to be able to process packets. A tool (te.cpp) provided on Mahoney's web site http://www.cs.fit.edu/mmahoney/dist preprocessed the raw network data into streams for LERAD. A Perl script (a21.p1), also provided by Mahoney, transformed

the streams into a LERAD-readable database format. The preprocessing tool te.cpp was altered so that it also produced a file of packets readable by PAYL.

Also, some postprocessing was required. LERAD and PAYL produce a list of packets that the systems consider to be "attacks" (anomalies). We created a tool that produces alarm statistics by comparing the output of LERAD and PAYL to the DARPA/MIT labeled attacks. For further details on this postprocessing stage, see Section 5 below.

Finally, before we describe each system, note that we used LERAD unmodified as found on Mahoney's web site, listed above. However, the source code for PAYL is not currently available, and therefore we re-implemented the algorithms based on [12].

3.1 LERAD and Its Biases. As mentioned above, LERAD learns a set of classification rules. Rules take the following general form: $(a_i = v_j \wedge \cdots \wedge a_n = v_m) \rightarrow (a_k = v_p \vee \cdots \vee a_q = v_s)$ where the a's are attributes and the v's are values of these attributes. Only conjunction is allowed in the rule antecedent and only disjunction is allowed in the rule consequent. An example rule might be:
If the destination port number is 80 and the source port number is 80, then the first word in the payload is GET or the first word in the payload is SET or the number of bytes in the payload is greater than 60.
Recall that each of these rules describes normal (benign) system use.

LERAD's classification algorithm is the following. Each new example (packet) receives a score, which is the sum of rule violations. If the score exceeds a threshold, T_L, defined below, then the packet is classified as an "attack."

LERAD's training algorithm inputs a set of attack-free training examples, and outputs a rule set, R. The algorithm begins with rule creation, then does rule sorting and, finally, rule pruning.

LERAD has many implicit inductive biases embedded within the system. We selected those that are most relevant to the construction of our ensemble and formalized them. By doing this, we were better able to understand and predict the types of attacks for which LERAD is best suited to detect.

What are these relevant biases of LERAD? One is the set of attributes considered by LERAD, called S_L. We know that $|S_L| = 23$, and the specific attributes are the packet date, time, last two bytes of the destination IP address, last four bytes of the source IP address, the source and destination port numbers, the TCP flags for the first, next to last and last TCP packets, the duration in seconds, the number of payload bytes, and the first eight words in the payload.

A second bias is the threshold, T_L, used by LERAD during classification. Before formalizing this threshold, we first repeat the formula for the anomaly score for each new example (packet), which we consider a bias. From [5] this is: $score_{anomaly} = \sum_{i=1}^{m} \frac{n_i \cdot t_i}{e_i} F_{r_i}$ where F_{r_i} is 0 if rule r_i is satisfied and 1 if it is not satisfied, m is $|R|$ (i.e., the number of rules), n_i is the rule support for rule r_i (defined above), e_i is the number of expressions in the antecedent of rule r_i, and t_i is the time that has elapsed since the rule was last violated. Then the threshold, T_L, is: $ln(score_{anomaly})/ln(10) > 4.5$. Finally, the last bias that we considered relevant in LERAD is the fact that its hypotheses take the form of rules, which we already formalized syntactically above.

3.2 PAYL and Its Biases. The classification hypotheses of PAYL are byte distributions, derived from the training data (see Figure 1).distribution is an empirically-derived approximation of a probability distribution $P(b_0, b_2, \ldots, b_{255})$, i.e., the probability of seeing each ASCII byte in a packet of a certain type. The types of packets are those that have a particular destination port number or a particular payload length. In other words, for each unique port number and payload length, PAYL associates (and learns) a probability distribution over the individual bytes in the payload. In fact, the full hypothesis of PAYL consists of a set of *profiles*. Each profile consists of a pair of 256-valued vectors (one for each byte). The first element of the pair is an average byte distribution, $P(b_0, b_1, \ldots b_{255})$, and the second element of the pair is a vector of standard deviations from the means, i.e., $(\sigma_0, \sigma_1, \ldots, \sigma_{255})$. Classification involves both a distance function and a threshold. If the distance between the byte distribution of a new example and the byte distribution of the hypothesis (which represents a profile of normal behavior) exceeds the threshold, then the new example is labeled an "attack."

PAYL's training algorithm consists of first classifying the training examples (packets) according to their destination port number and payload length. Then, the mean and standard deviation are calculated for each byte.

PAYL also has implicit inductive biases embedded within the system, and we selected those that are most

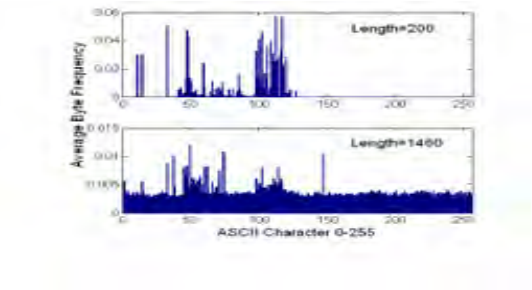


Figure 1: Sample byte distributions for different payload lengths of port 80 on the same host server.

relevant to the construction of our ensemble. The first bias is the set of attributes considered by PAYL, called S_P. Note that $|S_P| = 3$. These attributes are the destination port number, the number of bytes in the payload, and the distribution of bytes in the payload.

The second aspect of PAYL that we consider to be a bias is its distance function for computing the distance between two distributions. The function used by PAYL is (from [12]): $d(e, \overline{y}) = \sum_{i=0}^{255} \frac{|e_i - \overline{y_i}|}{\sigma_i + \alpha}$ where $\overline{y}$ and σ are the average and standard deviation, i.e., elements of the profile that constitutes PAYL's hypothesis, e is an example, and α is a conditioning variable needed to prevent divide-by-zero errors.

The threshold, T_P, was not formalized. It was derived empirically, as described in Section 5, below.

3.3 Example Application of the System Biases. By making LERAD and PAYL's biases explicit, we have been able to analyze and understand them better. From this process, we have drawn the following conclusions about the suitability of these two systems to detecting different attack characteristics:

- LERAD is sensitive to only a subset of the information in a packet, namely, the packet header and the first eight words.
- Two packets that differ by even a single byte (among the attribute fields that LERAD examines) are likely to be classified differently by LERAD. A packet that satisfies both the antecedent and consequent of a rule can be made to violate the rule by changing a single byte in one of the fields (attributes) of the consequent.
- PAYL is sensitive to only a subset of the information in a packet, namely, the packet payload and the destination port number.
- Two packets that differ by a single byte are unlikely to be classified by PAYL as being different – because the byte distributions of the two packets will probably be very similar.

The following example illustrates how these biases of LERAD and PAYL translate into specialized detection capabilities:

EXAMPLE 1. Consider the *Denial of Service* attack called "Back." This attack is a malformed web request to an HTTP port with the payload, "Get //// ..." followed by 6000-7000 slashes. One of the biases of PAYL is that it examines the byte distributions. For this particular attack, the byte distribution of the payload is almost exclusively centered on the "/" character in the ASCII table. This implies that PAYL will almost surely detect the attack. One of the biases of LERAD is that it examines the relationships of the first eight words in the payload of a message. Since "GET" followed by 6000-7000 slashes is an unusual relationship, we would also expect LERAD to detect this attack. Therefore, the "Back" attack is a specific type of attack that would be detected by both systems.

4 The Network Data.

As mentioned above, we are working with the DARPA/MIT Lincoln Laboratory database of packets. We decided to only work with the 1999 data, since the 1998 data does not include a key to differentiate normal packets from attack packets. The data from 1999 is five weeks long. The first and third weeks are attack free, whereas the

second, fourth and fifth weeks contain attacks. Each TCP packet is a binary sequence not exceeding 64,000 bytes in length. The DARPA web site classifies attacks into five categories: 1.**Denial of Service (DoS)**, 2.**User to Root (U2R)**, 3.**Remote to User (R2U)**, 4.**Probes.**, and 5. **Data.**

This classification is standard, comprehensive, and still modern [11]. Nevertheless, this DARPA classification does not result in mutually exclusive classes. Therefore, we have expanded the classification to include overlaps as two additional classes, thus resulting in a total of seven attack classes. The two additional attack classes are: 6. **Data and User to Root** and 7. **Data and Root to Local**. The 1999 DARPA data was divided into three sets. The training set consisted of all the data that was attack free. Training data: 03/08-03/12 (week one) and 03/22-03/26 (week three). Of the remaining data, the test and validation data sets were divided as follows: test data: 03/29-03/31, 04/02, 04/05 and 04/09; validation data: 04/01, 04/06-04/08. Both the test and validation data sets contain many attacks, though some attacks occur only in the test set and other attacks occur only in the validation set. There are slightly more attacks in the test set than in the validation set.

5 Ensemble Implementation.

In this section we describe in detail step (2) of the pipeline, introduced in Section 1, and called the "refinement by ensemble" step. The purpose of the ensemble is to produce a classification vector associated with a new packet, which could be an attack. For the ensemble, we concentrate only on test data that consists of attacks, i.e., non-attack packets are ignored.[1] There are a couple of reasons that we did this. For one, these are the only packets whose classification needs to be refined. Second, segmenting the data to determine the temporal boundaries of non-attacks proved to be very difficult – it proved challenging to determine the exact duration of non-attack packets. Therefore, our test data and validation data focused on attack packets only, and how to refine their classification.

When combining LERAD and PAYL, there are four possibilities for the combined outputs of the two systems: L-yes and P-yes, L-yes and P-no, L-no and P-yes, and L-no and P-no, where "yes" means the packet is labeled an "attack" and "no" means the packet is labeled a "non-attack." One of these pairs becomes the input to the ensemble system, for each new example/packet. The output of the ensemble for one pair is a probability distribution, called the *probability classification vector*, which gives the probabilities that the new example falls into each of the seven possible attack classes, described in Section 4 above. In summary, given a new packet, which we also call a *sample* to be consistent with statistical definitions, there are four possible events corresponding to the four possible class labels given by the pair of classifier systems. These input events need to be converted to an output probability distribution over the seven attack classes. Recall that this process is performed on the test data set. In other words, the training data is used for LERAD and PAYL to learn their hypotheses, and then these hypotheses make predictions over the test data to discover anomalies that are different from the hypotheses about normal user behavior. We combine LERAD and PAYL's predictions on the test data with the true (based on the DARPA web site) classifications of the test data packets. Then, we convert this information into a probability classification vector for each pair of outcomes from LERAD and PAYL. These vectors are joined together in an ensemble probability classification matrix (EPCM), and output by the ensemble (see Table 2).

To accomplish this, the first step is to formalize, in probability terminology, what precisely we are trying to find during step (2), i.e., what is the formal representation of an ensemble probability classification matrix (EPCM)? The answer is that we want to find $P(C|E)$, where C is a classification vector, i.e., a probability distribution over the seven classes, and E is an input event, i.e., the pair of labels given by LERAD and PAYL, such as L-yes and P-no. This probability value cannot be approximated directly from the results of the test data. We need to use a conditional probability rule to calculate this conditional probability. The conditional probability rule that we use is: $P(Y|X) = (P(Y,X)/P(X))$.

For example, suppose we have the results from the test data in Table 1. Each table entry that is not listed under "Sum" represents $P(Y,X)$, i.e., the frequency (which is an estimate of the probability based on the test data) of a packet giving a certain pair of binary outputs by LERAD and PAYL *and* being in a particular attack class (based on the DARPA/MIT web site classification of the test data). Furthermore, the "Sum" entry at the bottom of each column represents $P(X)$, i.e., the frequency of a certain pair of binary outputs by LERAD and PAYL. Using the conditional probability rule, given above, we calculate $P(Y|X)$, which is the output of the ensemble. Continuing our example from Table 1, by applying the conditional probability rule we get Table 2.

[1] We use the same criteria as DARPA did to label packets as "attacks."

Attack	y/y	y/n	n/y	n/n	Sum
Class 1	4	8	5	6	23
Class 2	6	4	2	1	13
Class 3	1	16	1	10	28
Class 4	0	0	2	1	3
Class 5	0	6	0	7	13
Class 6	2	1	1	0	4
Class 7	0	0	1	0	1
Sum	13	35	12	25	85

Table 1: Matrix of frequencies of attack events and classifier labels for LERAD/PAYL.

Attack	y/y	y/n	n/y	n/n
Class 1	0.3077	0.2286	0.4167	0.2400
Class 2	0.4615	0.1143	0.1667	0.0400
Class 3	0.0770	0.4571	0.0833	0.4000
Class 4	0	0	0.1667	0.0400
Class 5	0	0.1714	0	0.2800
Class 6	0.1538	0.0286	0.0833	0
Class 7	0	0	0.0833	0

Table 2: An ensemble probability classification matrix (EPCM), which is output by the ensemble. Each column is a probability classification vector.

Note that this is a matrix consisting of vectors (the columns) – one for each input event/sample, giving the vector output that is a probability distribution over the seven possible attack classes. This is what we call the "ensemble probabilistic classification vector."

For each new packet in the final validation data we can now use these vectors for classification. In particular, we run LERAD and PAYL on this new packet. If we get L-yes and P-no, then the ensemble predicts (using Table 2) the probability that the attack is of type Denial of Service is approximately 0.2268. The probability that the attack is of type User to Root is approximately 0.1143, and similarly for the remaining classes of attacks.

Given this ensemble output information, a misuse detection system could restrict its search and computations to a small subset of possible attack signatures when trying to find the most similar previous attack. The reasons for continuing with a misuse detection system are that our ensemble outputs probabilities – however a match with a signature could give further confirmation of the attack class, and also a stored signature could be used for predicting the attacker's next move.

We conclude this section by noting the role that the system inductive biases played in determining the probability classification vectors. Note that if the ensemble input is L-yes and P-yes, then the ensemble will conclude that the highest probability is that we either have an attack of Class 1 (Denial of Service) or an attack of Class 2 (User to Root). Having a high probability of being an attack of Class 1 can be explained in terms of the system biases. Recall Example 1 from Section 3.3, which was an example of a Denial of Service attack. In that case, the large number of slashes indicated that such an attack would be manifested as an unusual byte distribution and would therefore be likely to be detected by PAYL. Furthermore, the usual relationship between the slashes and one of the keywords indicated that such an attack would also probably be detected by LERAD. Based on the system biases, we therefore predicted that Denial of Service attacks would frequently result in L-yes and P-yes. Table 2 indeed confirms our prediction.

In summary, our analysis of system biases was quite helpful for both predicting and understanding the output of our ensemble. Future versions of our ensemble approach will investigate building an ensemble from first principles, based on bias analyses, rather than using a purely empirical approach.

5.1 Parameter Tuning. LERAD's process of learning a rule set involves a random element (see Section 3.1). Nevertheless, our experimental investigations revealed that there are not significant differences in performance

Attack	y/y	y/n	n/y	n/n	Sum
Class 1	5	9	8	1	23
Class 2	6	4	2	1	13
Class 3	1	13	9	5	28
Class 4	1	1	0	1	3
Class 5	2	8	2	1	13
Class 6	1	2	1	0	4
Class 7	0	0	1	0	1
Sum	16	37	23	9	85

Table 3: A random frequency matrix with the same row sums as in Table 1.

arising as a result of alternating the random seed. Therefore, we fixed LERAD's random seed to be 0, and all results described in this paper assume this same seed.

We ran extensive empirical experiments to find optimal settings for the parameters of PAYL: $T_P = 256$ and $\alpha = 0.1$. These are the values that are used in all of the empirical experiments, described below.

6 A Matrix-Matrix Comparison.

6.1 Another Matrix for Comparison. To evaluate the quality of our ensemble output, we require a comparison against a reasonable standard. For this purpose, we decided to use a *random frequency matrix*. Such a matrix is created using randomly-chosen matrix entries that are weighted based on the relative frequency of each class of attack in the test data. In other words, it is not purely random, but contains useful information about attack frequencies, and it is constructed from the test data – just like our ensemble is.

The particular methodology for creating the random frequency matrix was to use the test data to determine both the attack frequencies and to ensure that the random frequency matrix has the same row sums as the actual frequency matrix created from the test data (which was shown in Table 1 and was used directly for building the ensemble). In other words, *both* the ensemble probability classification matrix (EPCM) and the random frequency matrix are constructed based on information from the actual frequency matrix derived from the test data set. The difference between them is that the EPCM has probability entries that directly reflect the test data, whereas the random frequency matrix has characteristics that reflect those of the test data, but includes some randomness. An example of a transformation of an actual frequency matrix to a random frequency matrix is shown in Table 3. Then, we convert the random frequency matrix into entries that are probabilities, just like we did for the EPCM in Section 5. We call this final matrix a *weighted random probability classification matrix*, abbreviated *WRPCM*.

Finally, observe that the WRPCM is randomly created. Therefore comparing the EPCM with one WRPCM is statistically meaningless. To resolve this issue, we created 10,000 WRPCMs to compare with one EPCM, and took the mean and standard deviation of the differences as our evaluation.

6.2 Evaluation Metric. We created a *validation probability matrix (VPM)* over the validation data set – for the validation data set this is "ground truth" and is used as the performance standard. To measure the distance between the EPCM or a WRPCM and the VPM, we used the standard *Euclidean metric*, which sums distances between pairs of matrix entries.

We applied the Euclidean evaluation metric to compare the EPCM-VPM distance versus WRPCM-VPM distance, on the validation data set. The following section describes the results of these comparisons.

6.3 Experimental Results. The average distance from the EPCM-VPM distance value to the 10,000 WRPCM-VPM distance values is 0.7264, and the standard deviation is 0.1516. Using the Euclidean metric, we find that the distance between the EPCM-VPM value and the mean of the WRPCM-VPM values is 0.3463, and the EPCM-VPM value is 2.507 standard deviations from the mean of the WRPCM-VPM values.

6.4 Interpretation of Results. The EPCM is more than 2.5 standard deviations closer (which is better) to the VPM (considered "ground truth") on the validation set than the average of the 10,000 WRPCMs. In other words, a weighted random guess has a very low chance of being more accurate than the EPCM. In particular,

the probability that a random accuracy variable X is less than the ensemble accuracy is $P(X \leq 0.3463) = P(((X - \mu)/\sigma) \leq ((0.3463 - \mu)/\sigma) = P(z \leq -2.5075) = F(-2.5073) = 0.0062$, assuming distances are normally distributed. From the experimental results, we found that only 24 of the 10,000 WRPCM accuracies were better than those of the ensemble, which is quite low.

7 Summary and Future Work.

In summary, we have introduced a novel approach to an ensemble of classifiers that is designed for classification refinement, rather than for improving classification accuracy. Our experimental results indicate that our approach is very promising, and applicable to intrusion detection. In particular, our ensemble increased the number of attack classes from one to seven. Furthermorecloser (which is better) to the correct VPM on the validation data than the average of its competitors (the WRPCMs).

Our ensemble has an important role to play in refining the binary classifications output by the anomaly detection systems, prior to running a misuse detection system. The final step of the pipeline process described in Section 1, that of feeding the ensemble output into a misuse detection system, needs to be accomplished as part of future work. For example, we might use SNORT [10], which is the most widely available commercial misuse detection system. SNORT has a rule associated with each attack, so we might consider using our ensemble to partition the rule set according to attack class, and then check a potential attack packet with the rules from the class to which there is the greatest probability (according to the ensemble) that the attack belongs. This would increase SNORT's classification speed. It is interesting to note that for this paradigm, valuable information would be produced by the ensemble even if all classifiers (members) of the ensemble individually classified the candidate packet as a "non-attack." This is because even if its component classifiers label a packet as a "non-attack," the ensemble still predicts a class of attack for the packet. Therefore, if the packet does indeed turn out to be an attack, the ensemble will be especially helpful.

Finally, recall that we mentioned earlier that this ensemble approach is not only scalable, but is likely to benefit in performance from the incorporation of additional classifiers. We intend to explore this fruitful future direction for our research.

References

[1] E. Alpaydin, *Introduction to Machine Learning*, MIT Press: Cambridge, MA, 2004.

[2] Defense Advanced Research Projects Agency, *DoD Standard Transmission Control Protocol*, Information Processing Techniques Office, Arlington, VA (1990).

[3] R. Durst and T. Champion and B. Witten and E. Miller and L. Spagnuolo, *Testing and evaluating computer intrusion detection systems*, Comm. of the ACM, 42(7), (1999), pp. 53–61.

[4] J. W. Haines and R. P. Lippmann and D. J. Fried and E. Tran and S. Boswell and M. A. Zissman, *1999 DARPA Intrusion detection system evaluation: Design and procedures*, MIT Lincoln Laboratory Tech. Report.

[5] M. V. Mahoney, *A machine learning approach to detecting attacks by identifying anomalies in network traffic*, Ph.D. dissertation, Florida Tech., 2003.

[6] M. V. Mahoney and P. K. Chan, *An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection*, Proc. RAID, (2003), pp. 220–237.

[7] ——, *Learning rules for anomaly detection of hostile network traffic*, Proc. Third International Conference on Data Mining (ICDM), (1987).

[8] J. S. Milton and J. C. Arnold, *Introduction to Probability and Statistics Principles and Applications for Engineering and the Computer Sciences*, Third Ed., McGraw-Hill: NY, 1995.

[9] T. M. Mitchell, *Machine Learning*, McGraw-Hill: Boston, MA, 1997.

[10] L. Schaelicke and K. Wheeler and C. Freeland, *SPANIDS: A scalable network intrusion detection loadbalancer*, Proc. Comp. Frontiers, (2005), pp. 315–332.

[11] J. Wang, *Loss-sensitive rules for intrusion detection and response*, Ph.D. dissertation, Univ. of Pennsylvania, 2004.

[12] K. Wang and S. J. Stolfo, *Anomalous payload-based network intrusion detection*, Proc. RAID, (2004), pp. 1–12.

[13] D. H. Wolpert, *Stacked generalization*, Neural networks 5, (1992), pp. 241–259.

Scaled Entropy and DF-SE: *Different* and Improved Unsupervised Feature Selection Techniques for Text Clustering

Deepak P[Δ*], Shourya Roy[#]
[*]Department of CS&E, IIT Madras, Chennai, India
[#]IBM India Research Lab, IIT Delhi, Hauz Khas, New Delhi, India
deepakswallet@gmail.com, rshourya@in.ibm.com

Abstract

Unsupervised feature selection techniques for text data are gaining more and more attention over the last few years. Text data is different from structured data, both in origin and content, and they have some special differentiating properties from other types of data. In this work we analyze some such features and exploit them to propose a new unsupervised feature selection technique called *Scaled Entropy*. Our experiments on standard corpora show that *Scaled Entropy* is *different* from other existing techniques and outperforms them more often than not. We have proposed a technique, inspired by Spearman Rank Correlation Co-efficient [1], for comparing different feature selection methods in terms of selected features. We have shown that the feature selection techniques which are significantly uncorrelated according to this measure, can be combined to produce better hybrid methods. As another contribution of this work, we propose two such hybrid unsupervised feature selection techniques. One of them, combination of *Scaled Entropy* and *Document Frequency*, works significantly better than the state-of-the-art techniques on standard text clustering corpora.

Keywords: Feature Selection, Text Clustering, Entropy, Scaled Entropy, Correlation Analysis

1. Introduction

Data mining techniques have gained a lot of attention of late. Two principal techniques in the aforesaid arena are clustering and classification. The fundamental difference between these two techniques comes from the fact that clustering does not require any *class label* information for every object, like classification. Clustering is the technique of grouping similar objects together to divide a collection into groups or clusters. Data in each group should share a common property – often proximity according to some defined distance measure. Two extensive survey papers [2,3] on clustering contain overview of commonly used clustering techniques. Text clustering [4] is the technique of grouping a collection of text documents, articles, Web pages etc. based on some similarity measure. Conventionally, documents to be clustered are represented as vectors of (normalized and/or idf-scaled) term frequencies. The number of elements in the vector would correspond to the size of the vocabulary (collection of all distinct terms) in the corpus. This representation has an inherent problem – *Curse of Dimensionality* [5] and associated sparseness of the data. As every document is represented as a vector of size equal to the vocabulary size, hence most of the entries (corresponding to the terms not present in that document) in the vector would be zero. Clustering vectors containing tens of thousands of entries causes performance bottleneck and hurts the accuracy of the clustering algorithm also. Hence to obtain meaningful clustering result, it is absolutely necessary to reduce the dimensionality of the feature space by reducing the size of the vocabulary. Feature selection is a technique for doing the same by selecting a subset of relevant and important features from the entire vocabulary and representing documents as vectors of selected features only. People have done lots of work in the area of feature selection for text classification. Some survey papers [6,7] review popular techniques in good detail. Supervised feature selection techniques for classification typically exploit the correlation between class labels and features to select the subset of features which are

[Δ] Work done while doing internship at IBM India Research Lab

most discriminating. Feature selection for clustering [8] or unsupervised feature selection is different from its supervised counterparts because of unavailability of class label information.

Although unsupervised feature selection techniques are less matured than their supervised counterparts, there exist a few techniques worth mentioning. Notable among them are linear Feature Selection techniques such as *Document Frequency* (DF) [9], *Entropy Based Ranking* (EN) [10], *Term Strength* (TS) [11] and *Term Frequency Variance* (TF) [8,12]. According to DF, the more number of documents a term occurs in, the more important the term is. TS of a term is measured based on the conditional probability of occurrence in the second half of a pair given that it has occurred in the first half. EN projects the entire vector space of documents onto a single term, and calculates the entropy of the projection. The lesser the entropy of the projection, the more important the term is. TF considers the importance of a term as being proportional to the variance of its term frequency. All these methods are linear in terms of number of documents in the corpus and we will refer to them as *low-cost* techniques. There are some other unsupervised techniques which are quadratic in number of documents such as the *Entropy* measure proposed in [13] and *Term Contribution* measure proposed in [9].

In this work, we propose a new low-cost technique for unsupervised feature selection called *Scaled Entropy* (SE). It exploits a property, which is very typical of text documents, for better feature selection: **presence of an attribute is more important for clustering than absence**. This technique compares well and quite frequently outperforms other state-of-the-art low-cost unsupervised feature selection techniques. We have proposed a technique, inspired by *Spearman Rank Correlation Co-efficient* [1], for comparing different feature selection methods. According to this measure, SE is considerably different, in terms of selected features, from other techniques. We have also shown that two different feature selection techniques can be combined to produce better feature selection techniques. We have observed that two feature selection techniques can be most effectively combined if they are both good in performance and 'different' (according to our proposed measure). Finally, we proposed one such hybrid method which outperforms with significant margin, other state-of-the-art low-cost unsupervised feature selection techniques. We consistently use K-Means [14] clustering algorithm to compare feature selection techniques, throughout this paper.

The rest of the paper is organized as follows. Section 2 describes the proposed Scaled Entropy technique and the intuition behind it. Section 3 describes the proposed measure to quantify agreement between different feature selection techniques based on the ranked list of selected features. Description of the experiments that evaluate various feature selection techniques on standard corpora and their results with an analysis of the results comprise Section 4. Section 5 concludes the paper by summarizing the contributions and listing out pointers for future work.

2. Scaled Entropy

2.1 Asymmetric Information Content Hypothesis

One unique property of text documents is that the presence of a word in a document is more informative than the absence of a word. As an illustrative example, consider two documents d_1 and d_2. We only know that d_1 contains terms *loan*, *interest*, *ATM* and credit along with other terms whereas d_2 does not contain these terms. Based on only this much information we can say that d_1 is likely to belong to a cluster of documents on *financial organizations* but nothing can be said about d_2. We refer to this hypothesis as the *Asymmetric Information Content* (AIC) hypothesis. Moreover, the information content regarding class membership of a document increases faster than linearly with the number of occurrences of a word in it. Euclidean distance, a metric which weighs presence and absence with the same weighting, is not considered as a good distance measure for text documents, as opposed to its widespread usage in other forms of data as image and bio-medical data [15].

A good feature should be able to distinguish between different classes of documents. In other words, when a set of documents is projected on a feature dimension, the projection of documents belonging to different classes should be well separated. Further it is observed in text data, the purity (uniformity of labels) increases away from the origin in a discriminating dimension. Figure 1 shows the projection of a 2-class (a subset of the R6 dataset described later) text dataset on a discriminating feature dimension. The documents are colored according to their labels.

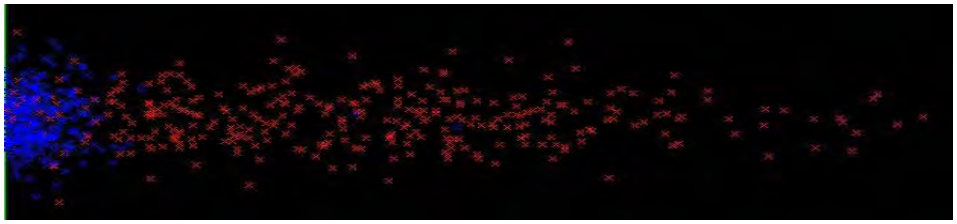


Figure 1. Projection of a 2-class text dataset onto a discriminating feature

2.2 Scaled Entropy Technique

Based on the observations mentioned in the previous section, we lay down three claims that, we believe are among the desiderata for a text feature selection technique. All these are for selecting a feature when the set of documents are projected on the dimension corresponding to the feature.

Claim 1. Skewed distributions are better than uniform distributions.
Claim 2. Features with clusters away from the origin should be preferred to those which have clusters closer to the origin.
Claim 3. Features with a sparse cluster away from the origin, should be preferred to features that have a dense cluster nearer to the origin.

Claim 1 is the idea behind the EN technique and is applicable for non-text data also. The others are based on weighting presence and absence asymmetrically and also that the importance increases faster than linearly with presence. The remaining part of this section gives the formulation of *Scaled Entropy* motivated by these three claims.

The projection of a document vector **d** on the dimension corresponding to the feature X is given by the dot product $\mathbf{d}.\mathbf{i}_X$, where $\mathbf{i}_X$ is the unit vector along the same dimension. Let $f_i(X)$ be the fraction of documents for which the projected value on X dimension be i. The Scaled Entropy(SE) value of feature X is calculated as per the following formula.

$$\sum_i \frac{f_i(X)\log(1+f_i(X))}{i}$$

Quality of feature X is inversely related to SE(X). The formulation similar to that of EN[10]. The division by i makes SE different from EN. This scales down the contribution of a set of documents by distance from the origin. Importance of a feature being inversely proportional to the contribution, it satisfies *Claim 2* explicitly and implicitly aids *Claim 3*. We will see how SE compares with other feature selection techniques in section 3 and 4.

3. Comparing Feature Selection Techniques

Before going to evaluate the goodness of the SE technique, we would like to show that SE is considerably different from other feature selection techniques in terms of the features it selects. Any feature selection technique gives a ranked list of features in descending order of importance. We wanted to use some of the well known techniques for comparing two ranked lists such as Spearman Rank Correlation Co-efficient

[1], Kendall-Tau Distance etc [16]. These techniques have an inherent assumption that the size of these two ranked lists are equal (say n) and each list contains same n distinct elements. However, top n features from two different feature selection techniques may not be (and most likely too) same. In the following subsection we propose a modification to Spearman Rank Correlation Co-efficient [1] technique to compare two lists which may contain different elements. Using the proposed measure, we compare state-of-the-art feature selection techniques with SE and show that SE is a considerably different technique. This observation eventually led us to another significant contribution of this work. If two feature selection techniques are considerably different and give good results independently then an intelligent combination of them is expected to give better results. We do not attempt to testify this hypothesis as it is evidently not unintuitive and an empirical justification of this hypothesis isn't essential for the problem that this paper tries to address. Carrying this hypothesis forward, the more different two good techniques are, the better it would be to combine them. We will introduce a couple of such hybrid methods in section 4.

3.1 Modified Spearman Correlation Coefficient

Spearman Rank Correlation Coefficient [1] (SCC) is a nonparametric (distribution-free) rank statistic proposed by Spearman[1] in 1904 as a measure of the strength of the associations between two variables. Given two ranked lists of size n the following measure M gets a value in the range [-1, +1] where the absolute value indicates the extent of correlation and the sign indicates the type of correlation.

$$M = 1 - 6\sum \frac{d^2}{n(n^2-1)}$$; where d is the difference in rank of corresponding variables

Two identical lists will have M value +1, two completely opposite lists will have M value -1. Now if the two lists may contain different elements then Spearman Rank Correlation Co-efficient cannot be used to compare them. To aid the comparison of such lists, we define the Modified Spearman Correlation Coefficient (MSCC) by the following formula.

$$M' = 1 - 6\sum \frac{d^2}{n(n+1)(2n+1)}$$

Where d is the difference in rank for every element that occurs in the union of the top n features of the two lists. The rank of a feature not present in a list is considered as (n+1). It can be verified that MSCC satisfies all the above mentioned boundary conditions i.e. for identical lists it would be +1 and for completely disjoint lists it would be -1. We choose not to explain MSCC and its properties in greater detail due to space constraints.

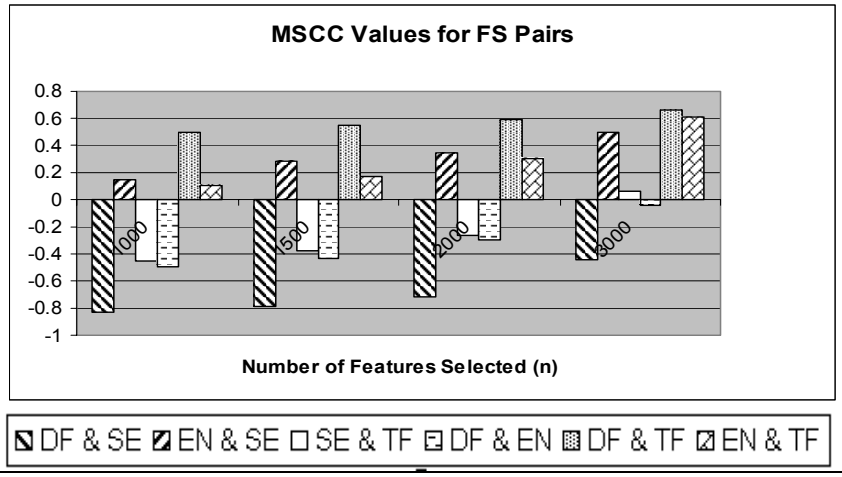
3.2 Analysis

We present the MSCC values for every pair of feature selection techniques DF, EN, TF and SE. Further, we define a *Composite Correlation Measure* for each technique as the sum of the correlations of that technique with the others in the set. The MSCC values have been computed for different values of n. The results presented here are results on the Reuters 6-cluster dataset (1359 documents and 11019 words), a subset of Reuters-21578, details of which can be found in a later section. As can be seen from the graph, SE is maximally different from DF and maximally similar to EN. Further, SE has the lowest composite similarity measure and hence can be said to be maximally different from other techniques. Another striking observation is that DF and TF appear to have a strong correlation. It may be noted that the correlation measure would increase as n increases, as the number of common features would increase with n. On a related note, we did this same analysis for each of these four methods with the supervised "Information Gain" (IG) [9] feature selection technique on the same dataset. It was found that SE is again highly uncorrelated with IG, the MSCC (for n = 1000) value being -0.63. DF, EN and TF had MSCC

[1] http://en.wikipedia.org/wiki/Charles_Spearman

values (n=1000) of 0.29, -0.2 and 0.37 respectively with IG. Based on this analysis we can conclude that SE is significantly different from the state-of-the-art techniques.

Table 1. Results of Correlation Analysis



Feature Selection Technique	Composite Correlation Measure
DF	**-1.74**
EN	**+1.20**
SE	**-2.54**
TF	**+2.44**

4. Experiments and Results

In this section, we present detailed experimental results to compare SE with other low-cost feature selection techniques. For each feature selection technique, we perform K-Means clustering [14] using Weka[2] on a test dataset and the quality of clustering results is considered as the metric to judge the goodness of the feature selection technique. We take the best of 5 runs with random initial seed value to reduce the effect of the bias of K-Means on the starting centers. Finally, based on our observation (mentioned in section 3) we propose two *hybrid* feature selection methods which are combination of two pairs of most dissimilar methods. We compare these new techniques with existing techniques similarly.

4.1 Datasets

We perform extensive experimentation with the Reuters 6 cluster dataset (R6) which is a subset of the Reuters-21578 dataset[3] containing uniquely-labeled documents. Classes considered were crude, trade, grain, money-fx, ship, interest. R6 contains 1359 documents and the size of the vocabulary after stopword removal is 11019. Further, we present results on the Classic3 dataset. Classic3[4] data set contains 1400 aerospace systems abstracts from the Cranfield collection, 1033 medical abstracts from the Medline collection and 1460 information retrieval abstracts from the Cisi collection, making up 3893 documents in all. After preprocessing, this data set had a vocabulary of 4303 words.

4.2 Hybrid Feature Selection Techniques

In section 3, we proposed that different and good feature selection techniques could be combined to create better hybrid feature selection techniques. We introduce two such measures in this section. The DF-SE technique is a combination of the DF and SE techniques. Top *n* features for the DF-SE technique would be the union of the top *n/3* features from the DF technique and top *2n/3* features from SE technique. This asymmetric division is of importance and stems from our preliminary results which show that DF performance peaks at a very low value of *n*, whereas other techniques peak much later. The other hybrid

[2] Weka Toolkit : http://www.cs.waikato.ac.nz/ml/weka/
[3] http://www.daviddlewis.com/resources/testcollections/reuters21578/
[4] ftp://ftp.cs.cornell.edu/pub/smart

measure SE-TF is a symmetric combination of SE and TF techniques. Every feature is tagged by the sum of its ranks according to TF and SE (lesser the sum, the better). Top *n* features are selected from this ranked list.

4.3 Clustering Quality Validation Measures

We use *purity* [17] and *entropy* [9] to validate the clustering quality. The class labels of documents are used only for validation and not for feature selection or clustering. Purity is defined as weighted sum of the fraction of documents of maximally represented class for each cluster. Entropy is defined as the weighted sum of the entropies of the clusters, the entropy for a cluster calculated as the uniformity of the cluster based on label information. Note that best technique in the case of purity is the technique that gives the highest purity, as opposed to entropy where lesser implies better.

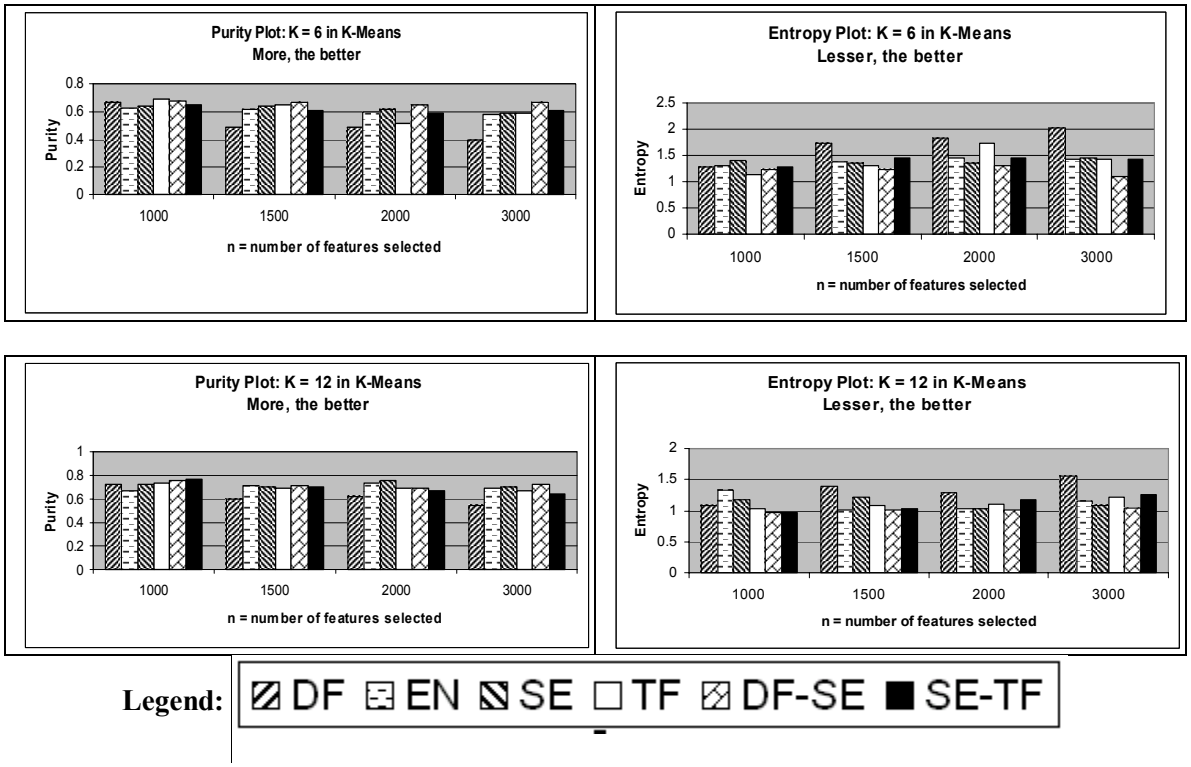
As we perform a host of experiments for varying values of *n* and *k* (in K-Means), we have an array of performance measures for every feature selection technique. To aid visual comparison of performances, we propose a single quality measure called *Sum of Deviations from Best* (*SDFB*) which is a per-technique score, aggregating the performances of the technique across experiments. SDFB for a particular feature selection technique (F_1) is computed as the sum of the absolute deviation of entropy (purity) of F_1 from the best entropy (purity) obtained (among all feature selection techniques for that experiment) over all experiments (varying *n* and *k*). Lower the value of SDFB, the better and consistent (for both entropy and purity SDFBs) the technique.

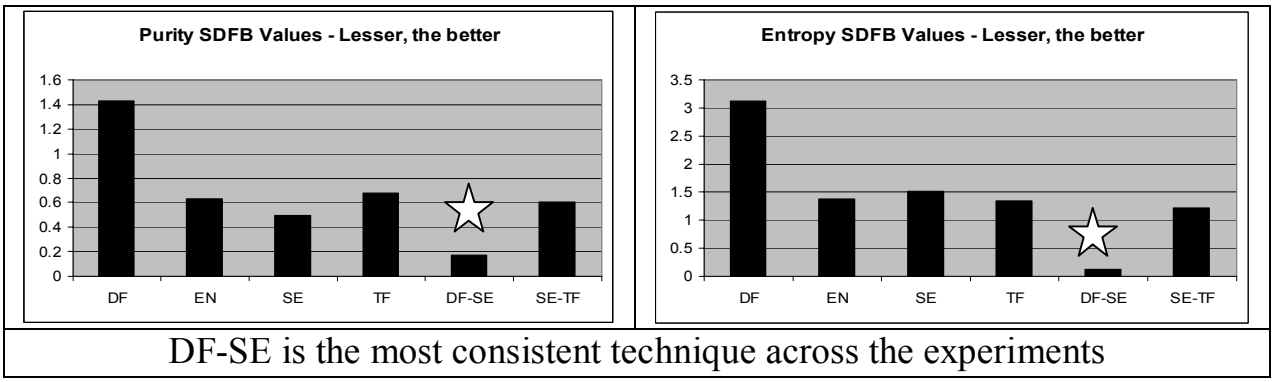
4.4 Results

We present an extensive set of charts from the R6 dataset experiments (some charts such as that for K=18 have been omitted due to space constraints) and a sample of the results for the Classic3 experiments.

4.4.1 Results on the R6 Dataset

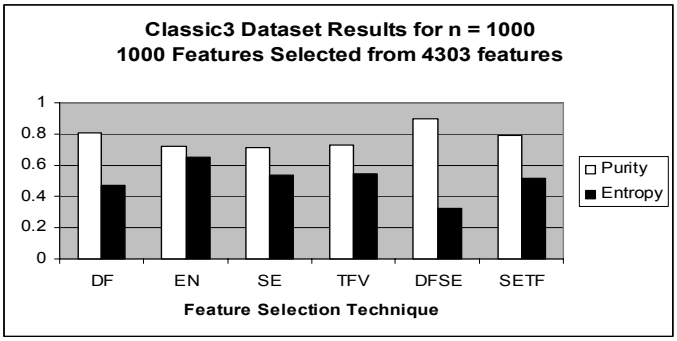
Table 2. Results on the R6 dataset for varying K and n





5.4.2 Results on the Classic3 Dataset

Table 3. Results on the Classic3 Dataset



4.5 Analysis

Many useful conclusions could be arrived at by a detailed analysis of the above charts. Firstly, DF shows a sharp decrease in performance with increasing *n* as depicted by the decrease in purity and increase in entropy as we move towards higher *n*. Secondly, DF is clearly inferior to the other techniques as illustrated by a huge pillar in the SDFB charts. Thirdly, the performance of SE compares well with those of EN and TF. Fourthly, and most importantly, DF-SE performs better than any of the other techniques and is the best performer in the majority of experiments with R6 resulting in a close-to-zero value for both Purity and Entropy SDFB. The performance on the Classic3 dataset further reinforces that DF-SE is very superior to all the other techniques. Fifthly, SE-TF doesn't seem to give too much of an improvement over its constituent feature selection techniques. The constituents of SE-TF weren't as 'different' as those of DF-SE (Ref: Section 3.2). This possibly, points to the fact that high-performing hybrid techniques could be obtained only by combining significantly different techniques.

5. Contributions and Future Work

In this work, we have laid down the AIC hypothesis that text data is very special in that occurrence of a term conveys more information than the absence of it. Based on this hypothesis we have proposed Scaled Entropy, a *different* feature selection technique which compares well in performance with existing techniques. We have proposed a measure to compare different feature selection techniques and based on our observation we have introduced the notion of combining different feature selection techniques to create better hybrid feature selection techniques.

Future work in this direction would be centered on the variability analysis of feature selection techniques and intelligent usage of it to generate hybrid feature selection techniques. Feature selection techniques could be subjected to variability analysis so as to handpick uncorrelated and good pairs to combine. In

fact, it may be possible to group existing feature selection techniques into few groups based on their mutual correlation. We hope that techniques which have similar heuristics will have higher correlation and hence would cluster together. Then techniques from different groups could be combined to obtain a better hybrid feature selection technique. Another possible extension of this work would be to devise specialized techniques for text using AIC hypothesis.

References

[1]. Lehmann, E. L. and D'Abrera, H. J. M. "Nonparametrics: Statistical Methods Based on Ranks*", rev. ed.* Englewood Cliffs, NJ: Prentice-Hall, 1998.

[2]. J. Han, M. Kamber, and A. K. H. Tung. Spatial clustering methods in data mining: A survey. In H. Miller and J. Han (eds.),Geographic Data Mining and Knowledge Discovery. Taylor and Francis, 2001.

[3]. A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. ACM Computing Surveys, 31(3):264-323, 1999.

[4]. Michael W Berry, "Survey of Text Mining: Clustering, Classification and Retrieval", Springer, 2004

[5]. Jain, Zongker, "Feature Selection: Evaluation, Application and Small Sample Performance", IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997

[6]. Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. Proc. of ICML-97 (pp. 412-420).

[7]. Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. Artificial Intelligence, 1(2), 245-271.

[8]. Bin Tang, Michael Shepherd, Evangelos Milios, Heywood, "Comparing and Combining Dimension Reduction Techniques for Efficient Text Clustering", Proceedings of the Workshop on Feature Selection for Data Mining, SIAM Data Mining, 2005

[9]. Liu, Liu, Chen, Ma, "An Evaluation of Feature Selection for text Clustering", Proceedings of the International Conference on Machine Learning, ICML-2003, 2003

[10]. Liu, Li, Wong, "A Comparative study on Feature Selection and Classification Methods Using Gene Expression Profiles and Proteomic Patterns", Genome Informatics, 2002

[11]. Wilbur, J.W., & Sirotkin, K. (1992). The automatic identification of stop words. Journal of Information Science, 18, 45-55.

[12]. I. S. Dhillon, J. Kogan, , and M. Nicholas. "Feature selection and document clustering". In M.W. Berry, editor, A Comprehensive Survey of Text mining. Springer, 2003.

[13]. Dash, M., & Liu, H. "Feature Selection for Clustering". Proc. of Pacific Asia Conference on Knowledge Discovery and Data Mining, PAKDD-2000, 2000

[14]. MacQueen, J. B. (1967). "Some methods for classification and analysis of multivariate observations." Proceedings of the Fifth Symposium on Math, Statistics, and Probability, Berkeley, CA

[15]. Bingham, Mannila, "Random Projection in Dimensionality Reduction: Applications to image and text data", Conference on Knowledge Discovery in Data (KDD 2001), 2001

[16]. Diaconis, P. 1988. "Group representation in probability and statistics". IMS Lecture Series 11, Institute of Mathematical Statistics.

[17]. Zhao, Karypis, "Criterion Function for Document Clustering: Experiments and Analysis", Department of Computer Science, University of Minnesota, TR#01-40

Classification and Prediction Using Empirical Distribution Functions

Ross Bettinger
SAS Institute, Inc.
Ross.Bettinger@SAS.com

Keywords

Classification, empirical distribution function, nearest neighbor, prediction, variable selection

Abstract

We describe a nonparametric algorithm based on the empirical distribution function which can be used for classification and prediction. The EDFs of at least ordinally-scaled variables are used to classify or predict the value of a target variable. The algorithm uses a function of the quantiles of the EDFs of independent variables to determine the nearest value of a dependent variable. The algorithm may be extended to binary-valued variables by assuming that there is a superior class and an inferior class, and that they may be labeled '1' and '0', respectively. The algorithm is robust with respect to missing data. There is a provision to perform variable selection to determine the best-performing subset of independent variables.

1. Introduction

We describe a robust nonparametric algorithm for classification and prediction that is based on the empirical distribution function (EDF). The EDFs of independent random variables that are at least ordinal in measurement scale are used to assign a class membership to an ordinal target variable or to predict the value of an interval-scaled target. The algorithm uses a weighted index of the quantiles of the EDFs of independent variables combined with a link function to determine the nearest value of a dependent variable. Binary-valued variables may be included by assuming that there is a superior class that may be labeled '1', and an inferior class that may be labeled '0'.

The empirical distribution function classification and prediction algorithm (EDFCAP) assumes that the ranks of the target variable are related to the ranks of the independent variables in a systematic manner that can be modeled by a functional relationship. The EDFCAP algorithm is robust in the presence of missing values since they do not impede the construction of the empirical distribution function. However, the EDFCAP algorithm assumes that the EDFs of the dependent and independent variables are "complete" in the sense that all possible values of the variables are present in their EDFs. Were this not to be the case, the gap in an EDF would create inaccuracies due to the omission of a representative value. Also, since outliers present no difficulty in formulating the EDF since only the ordering property of real numbers is used and not the distance between them, the results produced by the EDFCAP algorithm are not distorted by extreme values in the data.

2. Background

The cumulative distribution function (CDF) of a continuous random variable *x* with probability density function $f(x)$ is

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(t)dt \tag{1}$$

Some of the properties of the CDF are:

$$\begin{aligned} &F(x) = P(X \leq x) \quad -\infty < x < \infty \\ &0 \leq F(x) \leq 1 \end{aligned} \tag{2}$$

$F(x)$ is nondecreasing as *x* increases, i.e., if $x_1 < x_2$ then $F(x_1) \leq F(x_2)$ [1].

The EDF is the empirical analogue of the CDF. The EDF of a sample $\{x_i\}$, $i = 1,2,...,n$ is defined to be

$$\hat{F}(x) = \frac{1}{n}\#\{x_i \leq x\} = \frac{1}{n}\sum_{i=1}^{n} I(x_i \leq x) \tag{3}$$

where I is the indicator function. If *x* is a discrete ordinal random variable, the same definition holds with the probability mass function of *x* substituted for the probability density function of *x*. In the discrete case, $\hat{F}(x)$ is a step function with discontinuities at the interval boundaries of *x*. In the limiting case, $\hat{F} \to F$ as the number of discrete values increases because the minimum jump discontinuity is $1/n$. We use the definition of EDF for which the discontinuity occurs at the right-hand value of the data interval. $\hat{F}(x)$ defines a mapping from the domain of *x* to the range of values in [0, 1] such that $\hat{F}(x) = q$. The values *q* are the quantiles of the EDF, and represent the fraction of values x_i that are less than x_j for *i*<*j*. When expressed as percentages, the q are called "percentiles." We may interpret q as an estimate of the probability $P(X \le x) = F(x)$ by observing that $I(X \le x)$ is a Bernoulli random variable with parameter $p = F(x)$. Note that $E[I(X \le x)] = p$ and that $\text{var}[I(X \le x)] = p(1-p)$. Then by the definition of $\hat{F}(x)$ in (3), $q = \frac{1}{n}\sum_{i=1}^{n} I(x_i \le x)$ is the mean of n i.i.d. Bernoulli(p) random variables with mean $E[q] = p$ and $\text{var}[q] = p(1-p)/n$. Thus, q is an unbiased estimator of p. The interested reader is referred to [2] for a proof.

3. EDFCAP Algorithm

We may interpret the creation of the EDFs from non-missing x_i, $i = 1,...,m,\ m \le p$, and $\hat{F}_i(x_i) \mapsto [0,1]$ as a mapping $[0,1]^m \to [0,1]$ or as a projection of the EDFs of the X_i onto the EDF of Y. In this formulation, the $\hat{F}_i$ represent a varying-dimensional basis for spanning the quantized space of the non-missing X_i [1].

Let $(Y, X_1, \ldots, X_p)$ be a multivariate observation describing the output of some process Y, and simultaneously-observed independent random variables X_i associated with Y. Let $(Y_i, X_{i1}, \ldots, X_{ip}), i = 1, \ldots, n$ be a random sample of observations drawn from this process. We wish to use the EDFs of Y and the X_i to define a relationship between Y and X_i. Let $\hat{F}_i$ be the EDF for X_i. We can consider each $\hat{F}_i$ to be a one-dimensional table mapping X_i into q_i. Similarly for the dependent variable Y, $\hat{F}_Y(y) = q_Y$. By judiciously combining the q_i into a single result q_Y, we can approximately compute $\hat{F}_Y^{-1}(q_Y) = y$ by table lookup or interpolation. Since any EDF is a discontinuous step function, the standard definition of an inverse does not pertain. We can, however, find a y that is close to $\hat{F}_Y^{-1}(q_Y)$ by using a search technique. The estimand *y* then represents the value of Y most closely related to the associated variables X_i, as represented by the covariation of the quantiles q_Y and q_i.

3.1 EDFCAP Implementation

1. Compute $\hat{F}_Y(y)$ and $\hat{F}_i(x_i)$ for observed outcome Y and independent random variables X_i.
2. Compute $\hat{q}_Y = g(q_1, \ldots, q_p)$ to map the quantiles into a representative value. The function g is a link function that converts the q_i of the independent variables into an equivalent value q_Y of the dependent variable, Y.

---

[1] For example, if the observation is (Y, X_1, X_2, X_3) and X_3 contains a missing value, the observation (Y, X_1, X_2) is used in subsequent processing.

3. Compute $\hat{F}_Y^{-1}(q_Y) = y$ to determine the predicted class membership (binary or ordinal scale) or interpolated value (interval scale) of Y.

3.2 Link Function

The link function g which combines the q_i into a single representative value may have various formulations. For example, an obvious link function is g as a weighted average, or centroid, of the q_i. Thus,

$$g(q_1,\ldots,q_p) = \sum_{i=1}^{p} w_i q_i \Big/ \sum_{i=1}^{p} w_i \tag{4}$$

where the w_i are the Pearson correlation coefficients r_i between q_Y and q_i[2]. An alternative link function combines the X_i into the normalized weighted Euclidean distance from the origin ($\hat{F}_i(-\infty)$) to $\hat{F}_i(x_i)$. Thus,

$$g(q_1,\ldots,q_p) = \sqrt{\sum_{i=1}^{p} w_i^2 q_i^2} \Big/ \sqrt{\sum_{i=1}^{p} w_i^2} \tag{5}$$

where the w_i are the Pearson correlation coefficients as above. Any measure of association may be used, such as the asymmetric uncertainty coefficient, which was also used in computations[3]. Only q_i for non-missing X_i are used. The choice of link function is singularly important because it represents the mapping from the domain of the X_i to the range of Y. Link functions may be described as "direct" or "indirect."

3.2.1 Direct Link Functions

Direct link functions compute q_Y from the q_i in one step. They are of the form specified in (4) or (5). We assume that the function of the q_i for non-missing values of X_i represents the best relationship between X_i and Y. The weights w_i are specified in Table 1.

Table 1. Direct Link Function Weighting Factors

Description	Abbreviation	Weighting Factor, w_i
Equally-weighted	A	1
Correlation	C	$\lvert r_i \rvert$
Exponential	E	$e^{\lvert r_i \rvert} - 1$
Number of Discrete Steps	N	$n_i^{\lvert r_i \rvert - 1}$
Log of reciprocal	LR	$\log(1/(1-\lvert r_i \rvert))$
Reciprocal	R	$1/(1-\lvert r_i \rvert)$

[2] The q_i for each value of X_i was paired with q_Y for each corresponding value of Y so that there was a 1-1 pairing of (q_i, q_Y) for each (X_i, Y) in the data in the computation of Pearson's *r*. Ideally, each pair of (X_i, X_j) would be orthogonal so that $r_{X_i,X_j} = 0$. However, since true orthogonality is rarely seen in practice, multicollinear pairs of independent variables must be eliminated from the analysis to eliminate the redundant influence of two variables that are almost collinear with each other.

[3] We could have used the χ^2 statistic, but it is biased in favor of variables with large numbers of categories. We could also have used Spearman's rank correlation coefficient, but for large N, it is equivalent to Pearson's *r*. The asymmetric uncertainty coefficient is based on entropy considerations and represents a different approach to measuring strength of association.

The weights were chosen according to several heuristic "axes" of inquiry:

- Powers of *r*: reciprocal, equally-weighted, correlation, corresponding to the powers -1, 0, 1
- Exponential of $|r_i|$ and log of reciprocal of $|r_i|$
- Number of discrete steps of EDF of X_i weighted by $|r_i|$, normalized by n^{-1}.

Additional weighting schemes may be readily devised. This is a topic for further investigation.

3.2.2 Indirect Link Function and Variable Selection

The indirect link function computes q_Y based on the most significant individual variables among the X_i. Significance is measured by the magnitude of the measure of association. This is a two-stage process in which 1) the most significant independent variables are chosen, and 2) q_Y is then computed based on the value of the link function. An example of the indirect link function used in this paper is "choose the three variables X_i, X_j, X_k corresponding to the first three largest w_i where the w_i are as described above, and then let $q_Y = g(x_i, x_j, x_k)$ for those selected variables." We assume that the three variables X_i, X_j, X_k most highly correlated with Y represent the best relationship between the X_i and Y. The number of variables considered was used as a parameter in the experimental design applied to evaluate EDFCAP's performance. We considered up to five variables per performance run.

3.2.3 Refining the Estimated Quantile q_Y

After the q_i has been estimated, the estimate may be further refined by using the information contained in the quantiles of the original dataset. For each (X_i, Y) in the data, the quantile values are used in a cubic polynomial regression algorithm to relate the quantiles of the Y to the quantiles of the X_i. Cubic polynomial regression was chosen because a cubic polynomial has the flexibility to model points of inflection of an EDF and also because it can represent an inverse relationship between q_i and q_Y. Applying the regression equation to the q_i now changes the meaning of the q_i to $q_{X_i|Y}$ since q_i has now been redefined in terms of q_Y. The estimated quantile $q_{X_i|Y}$ then becomes the input to the link function as defined *supra*. The decision to use cubic regression was a parameter in the evaluation of the EDFCAP algorithm.

3.3 Classification and Prediction

After the link function has computed q_Y, the appropriate value of y is determined by table lookup (binary or ordinal variables) or interpolation (interval variables). There are alternative methods for performing the respective assignment of a value to y, depending on the measurement scale of Y.

3.3.1 Predicting Ordinal Class Membership

Three methods are available. Simple table lookup selects the first y corresponding to the first value of q_Y that satisfies $\hat{F}_Y^{-1}(q_Y) = y$. Nearest neighbor table lookup selects the y corresponding to the quantile nearest to q_Y. For example, if $q_Y = 0.1947$ and the nearest quantile-class label pairs were (.15, 3) and (.20, 4) then the class label assigned would be 4 since the pair (.20, 4) is closer than (.15, 3). Distance is computed along the q_Y axis. Linear interpolation followed by nearest neighbor interpolation linearly interpolates the value of y and follows with a nearest neighbor calculation to assign class membership.

3.3.1.1 Binary Dependent Variable

When Y is a binary variable, the outcome of a classification is the assignment of a label of '1' or '0' to Y. The EDF will consist of only two values: the mean of Y and 1. Using the mean as a cutoff probability will cause the label '1' to be assigned to the observation if q_Y is equal to or greater than the mean. Additional processing may be applied to improve the performance of the EDFCAP classifier, however.

After the EDFs for the X_i have been created, the training dataset is scored and the quantiles q_Y are regarded as estimates of actual probabilities. A receiver operating characteristic (ROC) analysis may be performed to evaluate the performance of the classifier with respect to correct and incorrect classifications of the actual outcomes Y according to the values of the estimated probabilities q_Y. Then, the value of q_Y that is closest to the optimal (0, 1) point[4] is that value of q_Y such that

$$q_Y = \arg\min_i \sqrt{(0 - x_i)^2 + (1 - y_i)^2} \tag{6}$$

where x_i is the false positive (FP) value and y_i is the true positive (TP) value for training dataset observation *i*. This point will have the highest percent of correct decisions where correctness = sensitivity (true positive) + specificity (true negative).

After the best q_Y has been found through ROC analysis, observations may be scored by directly comparing the q_Y produced from the evaluation of the link function to q_Y^c, the estimate of the cutoff probability. If $q_Y \geq q_Y^c$, the label '1' is assigned to the observation, otherwise the label '0' is assigned.

3.3.2 Interpolating Interval Values

Four methods are available. Simple interpolation is the same as simple table lookup. Nearest neighbor interpolation is the same as for ordinal class membership. Linear interpolation uses the bounding values of the quantiles bracketing q_Y to linearly interpolate the value of y. Linear interpolation followed by nearest-neighbor interpolation is the same as for ordinal class membership.

4. EXPERIMENTAL DESIGN

We used nine datasets from the UCI Repository [3] since they are available publicly and are known to the machine learning community. We selected them if they had binary, ordinal, or interval variables for which one could serve as a dependent variable. Since the dependent variable for the abalone dataset, rings, has 29 discrete levels it was used as an interval variable and also as an ordinal variable. We also used two datasets from actual projects. The task for the Bank dataset was to build a model predicting accountholder attrition, a binary-valued target. There were 667,569 observations of which 10,488 were attritors Since the target population was only 1.57% of the data, a stratified model was built using random sampling to create a training dataset consisting of 70% attritors and 30% non-attritors. The variables used were age of accountholder, length of time as a bank customer, marital status and residence status as nominal variables, and two binary triggers related to utilization of on-line bank services. The nominal variables were converted into ordinal variables by using information gain associated with the target variable to order them. The task for the Retail dataset was to predict the number of units of fragrance purchased, an interval-scaled target. There were 3,069 observations. The variables were US Census 2000 variables related to age, ethnicity, renter or homeowner status, and median income, and retail variables such as size of store, and units sold of selected products (skincare, cough/cold medication, toys).

We used only those observations that had non-missing values of the dependent variable so we would be able to accurately compare EDFCAP predictions to original values. A specified percentage of the dependent variable's values were set to missing to indicate that they were to be predicted, and the remaining non-missing variables were used in constructing the EDFs. We arbitrarily assigned 5% of the dependent variable's values to be missing in all cases. Missing values were randomly assigned according to the values of a uniform random variable. If a dependent variable was set to missing, the observation containing it was put into a holdout dataset and the observation was not used in constructing the EDFs for that set of data. For each dataset, we created four pairs of training and holdout datasets by varying the

[4] The (0, 1) point in ROC space is the point for which a classifier generates correct decisions 100% of the time. The probability associated with a (FP, TP) point is called a *cutoff* probability. A useful reference for ROC analysis is [4], and [5] is an excellent theoretical treatment.

random number seed used to generate missing values four times[5]. Classification and prediction accuracy results were collected and tabulated for evaluation. The four holdout datasets containing predicted class membership (binary or ordinal scale) or interpolated value (interval scale) were combined into one dataset and analyzed as a single experiment. The figure of merit for binary and ordinal variables was classification accuracy. The equivalent statistic for interval variables was the root mean square error (RMSE) of the matched pair (actual, predicted) for each set of parameters.

4.1 EDFCAP Scenarios

For a specified dataset, the relevant factors that could be varied were: cubic regression, interpolation method, link function, measure of association, and weighting scheme. If a dependent variable was binary, e.g., for the Adult, Bank, and Pima Indian data, ROC analysis was used.

Cubic regression was alternately applied and not applied. The link function used was chosen from a set of such functions, and the interpolation method used depended on the measurement scale of the dependent variable. The EDFCAP algorithm was applied to each dataset by varying the algorithmic parameters in full-factorial fashion. Given the "treatments" of {cubic regression, interpolation method, link function, measure of association, and weighting scheme}, a scenario was executed for each combination of ordinal classifications, and similarly for interval predictions.

4.2 Decision Tree and Regression Scenarios

The SAS Enterprise Miner™ Decision Tree modeling node was used to create classification and regression trees for scoring the holdout data. The Regression modeling node was also used to create classification and prediction models as an additional comparison technique using logistic and linear regression, respectively. The data exclusive of the holdout sample were randomly sampled into training and validation datasets in a 70%/30% split. Decision trees used the Gini criterion for splitting, and regression models minimized validation error. Trees up to 10 levels deep were built. Regression models were built using main effects and the interaction effects of all variables, and used stepwise variable selection. The performance metric to be minimized in all regression scenarios was the validation error (classification) or the variance (prediction). For logistic regression models, the validation error is the negative loglikelihood. For linear regression, the validation error is the error sum of squares.

5. EXPERIMENTAL RESULTS

We evaluated the EDFCAP algorithm on its classification accuracy (binary and ordinal variables) and predictive accuracy (interval variables). The results were compared to those of the Decision Tree modeling node and the Regression node in SAS Enterprise Miner[6].

5.1 Classification Accuracy

The measure of classification accuracy was the percent of predictions that exactly matched the original data. Table 2 reports classification accuracy for EDFCAP, decision tree, and regression models. The most accurate result for each dataset is highlighted in bold font.

Table 2. Classification Accuracy in Percent

Dataset	Dependent Variable	EDFCAP Centroid	EDFCAP Euclidean	Decision Tree	Logistic Regression
Abalone	Rings	25.30	29.12	**37.35**	28.04
Adult	Income	77.56	77.18	**83.47**	81.70
Bank	Attrition	**23.60**	**23.60**	7.39	3.87

[5] The only exception to this protocol was the abalone data used for classification testing. Sampling the relative scarcity of low and high values of the dependent variable created empty classes in the initial tests, so we decided to create four holdout datasets as usual and group all of the training datasets into one dataset for model building.

[6] SAS Enterprise Miner™ was used to create the models, which were scored with the same test data used for EDFCAP. The Decision Tree modeling node was used because it is robust in the presence of missing values. Imputation of missing values was used for the Regression node by replacing missing interval data with mean values and missing ordinal data with the mode of the distribution of the variable for which data were missing.

Import85	Symboling	**42.22**	**42.22**	26.67	26.67
Pima Indian	Class	77.85	78.48	75.32	**79.75**

While the decision tree algorithm is clearly the best classifier for two out of five datasets (Abalone, Adult), the EDFCAP algorithm is similarly superior for two (Bank, Import85). Logistic regression was the best classifier for Pima Indian data, with EDFCAP a close contender.

5.2 Prediction Accuracy

The measure of prediction accuracy used to compare the actual value to the predicted value was the root mean squared error (RMSE). Table 3 reports prediction accuracy for the best results in terms of RMSE. The most accurate result is highlighted in bold font.

Table 3. Prediction Accuracy as RMSE

Dataset	Dependent Variable	EDFCAP Centroid	EDFCAP Euclidean	Decision Tree	Linear Regression
Abalone	Rings	2.63	2.58	**1.77**	2.08
Auto Mpg	Mpg	3.69	3.76	3.69	**3.26**
BUPA	Drinks	**2.68**	2.78	2.82	2.79
Housing	MEDV	7.56	6.95	4.94	**3.95**
Machine	PRP	49.75	47.84	76.47	**45.91**
Retail	Fragrance	**59.90**	**59.90**	1566.63	1175.84

For prediction problems, OLS regression is a more accurate predictor than either the decision tree or EDFCAP for three out of six datasets. EDFCAP returns the best performance for the Retail problem. Appendix A contains detailed results of the EDFCAP centroid prediction performance for this dataset.

5.3 Best EDFCAP Scenario

A scenario was executed for each combination of parameters appropriate for the measurement scale of the target variable. For classification models, three datasets had binary target variables (Adult, Bank, Pima Indian), so ROC analysis was appropriate for them; the remaining two datasets (Abalone, Import85) had ordinal target variables for which ROC analysis was not relevant. For prediction models with interval target variables, the parameters were the same as for classification models, with the exception of interpolation, for which linear interpolation could be performed.

The parameter settings of the best scenarios are listed in Table 4. Where there is more than one entry, all of those settings listed produced the same result.

Table 4. Best Scenario Parameter Settings

Dataset	Dep Var Scale	Cubic Reg	Interpolation	Link Function	Measure of Association	Weighting Factor
Binary or Ordinal Target Variable						
Abalone	Ord	Y	S	Dist, I I5	U	C E LR R
Adult	Bin	N	LNN NN S	Dist, I2	U	A
Bank	Bin	Y	NN	Cent, I4	U	A C E LR N R
Import85	Ord	Y	NN	Cent, D I5	P U	C E LR R
Pima Indians	Bin	N	NN S	Cent, I3	P	LR
Interval Target Variable						
Abalone	Int	Y	LNN	Dist, I2	U	LR
Auto MPG	Int	Y	L	Cent, D	P	N
BUPA	Int	Y	NN	Cent, I3	U	N
Housing	Int	Y	LNN	Dist, I2	P	LR
Machine	Int	N	S	Dist, I4	P	N
Retail	Int	N Y	NN	Dist, D	P U	R

Cubic regression proved to be an effective enhancement to the link function for target variables regardless of measurement scale, and nearest neighbor interpolation by itself or with linear interpolation was likewise most effective. For binary or ordinal target variables, the centroid link function and uncertainty coefficient measure of association gave best classification accuracy, while for continuous target variables, the distance link function and Pearson's *r* were best. The best weighting factors for binary or ordinal target variables were correlation, exponential, and some form of reciprocal, and for interval target variables, some form of reciprocal and number of discrete steps in the EDF were most effective.

6. CONCLUSION

We described a robust nonparametric algorithm that uses a function of the quantiles of empirical distribution functions of independent variables to assign class labels to a binary or ordinal target variable or to predict the interval value of a continuous target variable. We compared its performance to a classification and regression tree algorithm and linear and logistic regression models and observed that the EDFCAP algorithm performed comparably to classification and regression trees and to logistic and linear regression on the datasets chosen for evaluation.

7. ACKNOWLEDGEMENTS

We thank Leonardo Auslender, Dave Duling, and Bob Lucas for their valuable and thoughtful comments.

8. REFERENCES

[1] DeGroot, M.H., *Probability and Statistics*, Addison-Wesley, Reading, MA, 1975.

[2] Lewis, P.A.W, and Oren, E.J., *Simulation Methodology for Statisticians, Operations Analysts, and Engineers, Volume 1*,Wadsworth, Inc., 1989.

[3] Blake, C.L. & Merz, C.J. (1998). UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

[4] Potts, William J.E., and Patetta, Michael J., *Predictive Modeling Using Logistic Regression Course Notes*, SAS Institute Inc., Cary, NC, 2000.

[5] Provost, Foster, and Fawcett, Tom, "Robust Classification for Imprecise Environments," *Machine Learning* 42, 203-231, 2001.

TRADEMARK CITATION

Appendix A – EDFCAP Centroid Link Function Prediction Performance for Retail Data

The results reported in section 5.2 are the best estimate of performance, but the range of results varied widely. The table below shows the variety of results computed by the EDFCAP centroid link function for the retail data.

Quantile	Estimate
100% Max	5581.6043
99%	5375.9315
95%	5104.1095
90%	4632.4862
75% Q3	3835.7879
50% Median	3453.1929
25% Q1	3384.2250
10%	2726.7356
5%	1274.4421
1%	60.3880
0% Min	59.9042

Using the Symmetrical Tau (τ) criterion for feature selection in decision tree and neural network learning

Fedja Hadzic and Tharam S. Dillon

Faculty of Information Technology, University of Technology Sydney, Australia

email: (fhadzic, tharam)@it.uts.edu.au

Abstract - *The data collected for various domain purposes usually contains some features irrelevant to the concept being learned. The presence of these features interferes with the learning mechanism and as a result the predicted models tend to be more complex and less accurate. It is important to employ an effective feature selection strategy so that only the necessary and significant features will be used to learn the concept at hand. The Symmetrical Tau (τ) [13] is a statistical-heuristic measure for the capability of an attribute in predicting the class of another attribute, and it has successfully been used as a feature selection criterion during decision tree construction. In this paper we aim to demonstrate some other ways of effectively using the τ criterion to filter out the irrelevant features prior to learning (pre-pruning) and after the learning process (post-pruning). For the pre-pruning approach we perform two experiments, one where the irrelevant features are filtered out according to their τ value, and one where we calculate the τ criterion for Boolean combinations of features and use the highest τ-valued combination. In the post-pruning approach we use the τ criterion to prune a trained neural network and thereby obtain a more accurate and simple rule set. The experiments are performed on data characterized by continuous and categorical attributes and the effectiveness of the proposed techniques is demonstrated by comparing the derived knowledge models in terms of complexity and accuracy.*

Keywords: feature selection, rule simplification, network pruning

1. Introduction

The data collected for various industrial, commercial or scientific purposes usually contains some features irrelevant to the concept of interest. When an induction algorithm is used to obtain a knowledge model about the concept the presence of these features interferes with the learning mechanism because of the noise introduced, and as a result the learned models tend to be more complex and less accurate. It is important to employ an effective feature selection strategy so that only the necessary and significant features will be used to learn the concept at hand. By concentrating on only the important aspects of the domain the derived knowledge models will be improved in terms of accuracy and comprehensibility.

Feature selection strategies can be roughly categorized into filter and wrapper based approaches. Filter approach is done independently of the learning algorithm and the irrelevant features are filtered out prior to learning. Common technique is to evaluate the features based upon their capability of predicting the target attribute and then to choose a subset of features with sufficiently high values. One such approach is the 'Relief' algorithm [6] that assumes two-class classification problems, and is inspired by instance-based learning. Relief detects those features statistically relevant to the target concept by assigning a relevance weight to each feature. It conducts in random sampling of the instances from the training set during which the relevance values are updated. The updating of relevance values is based on the difference between the selected instance and the two nearest instances of the same and opposite class. Another filter approach is "FOCUS" [1] which exhaustively examines all subsets of features and selects the minimal subset that is sufficient to determine the target concept for all instances in the learning set.

In a wrapper based approach [7] the feature selection algorithm exists as a "wrapper" around the induction algorithm. The algorithm conducts a search for a good subset of attributes using the induction algorithm itself as part of the evaluation function. The benefits of using the induction algorithm itself for evaluating feature subsets is that there will be no inductive bias introduced by a separate measure. On the other hand the major disadvantage is the computational cost associated with each call to the induction algorithm for evaluating the feature set [3]. More recently a hybrid algorithm named FortalFS [12] has been proposed, which uses results of another feature selection approach as the starting point in the search through feature subsets that are evaluated by the induction algorithm. In [4] a genetic algorithm SET-Gen was described for solving the problem of feature subset selection. A population of best feature subsets is kept and genetic operators are applied in order to create new feature subsets, which are evaluated according to the predefined fitness function. The fitness function favors those subsets that produce smaller decision trees, use less input features and retain predictive accuracy.

When dealing with the feature selection for neural networks (NN) the problem is commonly referred to as network pruning and it is split into pre-pruning and post-pruning approaches. Pre-pruning is essentially the same as the filter approach and post-pruning approach trains a network to completion and then inspects the links between particular network units in order to determine the relevance between the two [9]. This approach is useful for rule simplification and for removal of attributes whose usefulness has been lost through the learning. Most of the methods for symbolic rule extraction from NN use some kind of pruning technique to increase the performance and produce simpler rules. The contribution of each unit in the network is determined and a unit is removed if the performance of network does not decrease after the removal. This is often referred to as sensitivity analysis in NN and is one of the common techniques for network pruning [9,11].

Symmetrical Tau (τ) [13] is a statistical measure for the capability of attribute in predicting the class of another attribute. Previously it has successfully been used as a feature selection criterion during decision tree construction. The τ criterion was reported to have many promising properties and in this paper we particularly want to demonstrate its capability to handle continuous attributes, Boolean combinations of attributes and the capability of measuring an attribute's sequential variation in predictive capability. We provide an experimental study of some different ways the τ criterion can be used to filter out the irrelevant features prior to learning (pre-pruning) and after the learning process (post-pruning).

The rest of the paper is organized as follows. In section 2 we describe the τ criterion and its promising properties as a feature selection criterion. The three experimental procedures are described in section 3 and experimental results are provided and discussed for each procedure. The paper is concluded in section 4.

2. Symmetrical Tau (τ)

There are many different feature selection heuristics used for various inductive learning methods and some of the common disadvantages are: bias towards multi-valued attributes, errors in the presence of noise, not handling of Boolean combinations and sequential variation in predictive capability [10]**.** Zhou and Dillon [13] have introduced a statistical-heuristic feature selection criterion, Symmetrical Tau (τ), derived from the Goodman's and Kruskal's asymmetrical Tau measure of association for cross-classification tasks in the statistical area. The τ criterion has successfully been used to remove the irrelevant features during decision tree induction and has the following powerful properties:

- Built-in statistical strength to cope with noise;
- Dynamic error estimation conveys potential uncertainties in classification;
- Fair handling of multi-valued attributes;
- Not proportional to the sample size;
- Its proportional-reduction-in-error nature allows for an overall measure of a particular attribute's sequential variation in predictive ability. This determines which attributes have become less useful for prediction and should be deleted (pruned).

- Middle cut tendency separating a node into two balanced subsets;
- Handles Boolean combinations of logical features.

The τ criterion is calculated using a contingency table, which is a table that provides a two-way classification, and may be used if each feature of the sample can be classified according to two criteria. As a result c1*c2 contingency table can be formed, where c1 and c2 are the values of two criteria. If there are I rows and J columns in the table, the probability that an individual belongs to row category i and column category j is represented as P(ij), and P(i+) and P(+j) are the marginal probabilities in row category i and column category j respectively. The Symmetrical Tau measure is defined as [13]:

$$\tau = \frac{\sum_{j=1}^{J}\sum_{i=1}^{I}\frac{P(ij)^2}{P(+j)} + \sum_{i=1}^{I}\sum_{j=1}^{J}\frac{P(ij)^2}{P(i+)} - \sum_{i=1}^{I} P(i+)^2 - \sum_{j=1}^{J} P(+j)^2}{2 - \sum_{i=1}^{I} P(i+)^2 - \sum_{j=1}^{J} P(+j)^2}$$

For the purpose of feature selection problem one criteria (A) in the contingency table could be viewed as a feature and the other (B) as the target class that needs to be predicted. The τ criterion has the following properties [13]:

- In most cases it is well defined;
- If P(ij) = 1 for some i and j, and all other cells have zero probability then the categories of A and B are known with certainty;
- If τ = 0, then the feature in question has no predictive ability for the category of another feature. For this to occur there must be no P(ij) = 1, and all non-zero probabilities are in a single row or column of the contingency table;
- If τ = 1, then the feature in question has the prefect predictive ability for the category of another feature. For this to occur there cannot be any P(ij)=1 and either: for each j there exists an i such that P(i,j) = P(+j), or for each i there exists a j such that P(ij) = P(i+);
- For all other cases τ falls between 0 and 1;
- τ is invariant under permutations of rows and columns.

3. Experimental Procedure and Results

In this section we provide our various experimentations done to demonstrate some different ways Symmetrical Tau can be used as a feature selection criterion. In each of the sections the approach taken is described and the experimental results are provided. For experimentation the decision tree algorithm used is C4.5, and for neural network testing we used the standard back-propagation algorithm with 2 hidden layers, learning rate - 0.3, learning momentum - 0.2 and the training time of 500 epochs. The training set was made up of 60% of the available data, and the rest was used as the testing set for the accuracy of the predicted model. Any attributes that serve as a unique identifier of an instance have been removed from the training set. We have used data of varying complexity and attribute characteristics, publicly available from the ‘uci’ machine learning depository [2].

4.1 Filter approach using the τ criterion for feature selection

Here the τ criterion is used to rank the existing attributes according to their capability in predicting the class of the target attribute. Only the attributes with sufficiently high τ values will form a part of the feature subset to be used by the learning algorithm. The relevance cut-off point chosen is where the difference amongst the τ values of the ranked attributes is sufficiently high. The aim is to remove the

irrelevant attributes without decreasing the accuracy of the derived knowledge model. Table 1 summarizes the results obtained when the described method was applied to domains characterized by categorical (top six) and continuous (bottom three) attributes.

	C4.5								Back-propagation NN	
	1 – unpruned		2- pre-pruned		3 -post-pruned		4 - pre- and post-pruned		Full feature set	τ-reduced feature set
Domains	Size	accurracy	Size	accurracy	size	accurracy	size	accurracy	accurracy (%)	Accuracy (%)
Postop	33	47.2	25	75	7	72.2	7	72.2	61.1	63.88
Breast-cancer	45	93.57	45	94.28	31	93.928	23	95	95.7	96.78
Voting	37	95.977	29	97.126	11	97.7	11	97.7	92.5	94.25
Lenses	7	70	5	70	7	70	5	70	70	70
Mushroom	29	100	27	100	29	100	27	100	100	100
Zoo	17	92.68	15	92.68	17	92.68	15	92.68	82.9	87.8
Wine	13	91.6	13	91.6	9	91.6	9	91.6	95.8	98.61
E-coli	51	76.29	51	76.29	43	78.51	43	78.51	71.8	75.5
Glass	51	67.4419	49	69.7674	51	69.7674	41	70.9302	59.3	61.62

Table 1 – Results of applying the τ criterion for filtering out the irrelevant attributes

The comparison of the decision tree results are displayed on the left where unpruned corresponds to the results obtained when the standard C4.5 algorithm is used, pre-pruned when the attribute set has been reduced according to the τ criterion and post-pruned when the post-pruning technique from C4.5 is used. The size and predictive accuracy (%) of the resulting decision trees were compared, and improvements occurred when the attribute set was filtered according to the τ criterion. For the unpruned version comparison (1 versus 2), the resulting decision tree was simpler in all cases except for breast-cancer domain where it remained the same. The decrease of tree complexity was not at the cost of a reduction in accuracy. In fact accuracy was either improved or kept the same. A significant improvement in accuracy was observed in 'post-operative patients' domain, with an increase from 47.2 % to 75 %.

For the pruned version (3 versus 4), the resulting decision tree was simpler in all but three cases where it remained the same. The accuracy increased for breast-cancer and glass domain and remained the same for the rest. When comparing the results obtained either by applying the τ criterion for pre-pruning or the post-pruning approach from C4.5 (2 versus 3) there were four cases in which pre-pruning achieved a simpler tree and other five for post-pruning. Accuracy increased through pre-pruning for two cases and by post-pruning for two (rest is same). Besides this similarity one advantage of pre-pruning is that irrelevant features are detected early in the learning process which avoids poor choices being made for test-nodes in the tree. As both approaches combined achieved the best results in all but one domain (postoperative patients), good practice would be to use a filtering method first followed by a post-pruning method which will detect and delete those attributes that have become useless and possibly interfering for the prediction task.

The value of the τ criterion at which the attributes were removed from the training set varied in most of cases, and it was not easy to determine a general cut-off point. The factors that affected the cut-off point for a certain domain appeared to be the attribute-set size and interrelationship between the attributes within the set. For example in the mushroom domain high difference amongst the τ values occurred high in the ranking, and bottom 15 attributes could be removed without affecting the accuracy. On the other hand in the lenses and post-operative patients domain this difference occurred low in the ranking and only the bottom two attributes could be removed. Furthermore some attributes having low τ values proved to be important independently from the attribute-set size due to their interrelationship with other attributes from the training set. In the post-operative patients domain all the attributes had very low τ values, and only when combined they provided high predictive power. All these observations indicate the importance of measuring the predictive capability for Boolean combinations of attributes, which is discussed next.

4.2 Measuring predictive capability for Boolean combinations of features

In order to calculate τ for combinations of features the input data was transformed for each n-combination by combining the attributes and the values that occur in each instance. The τ criterion was then calculated for all n-combinations and the one with the highest τ value was the attribute subset used by the induction algorithm. In some domains (mushroom, zoo, voting) attribute set was too large to calculate all possible combinations, in which case the attributes with low τ values were removed. It should be noted that this could potentially miss the best combination as sometimes an attribute that may have a low τ value could become useful when combined with another attribute. If forming all possible combinations is still infeasible, one could continue to remove combinations at each step by determining a cut-off point for each set of n-combinations formed. Only the promising combinations would be used for forming higher n-combinations, and the combinatory explosion problem could be alleviated to some extent.
The results of the experiment are provided in table 2. Note that the results of applying the C4.5 algorithm with post-pruning to the highest τ-valued combination are excluded from this table as they remain the same to when no post-pruning is done. Besides the domains obtained from the 'uci' depository, we have used a simple noise-free syntactic file for recognizing LED digits in order to check that the τ value will be equal to one for the necessary and sufficient attribute combination. Indeed the combination of five attributes was detected with value of 1 which is the minimal required attribute set to obtain perfect predictive accuracy for this domain. This can be seen from table 2 as the C4.5 algorithm achieves perfect accuracy with the used combination. However, the neural network was incapable of achieving perfect accuracy with this combination. As we are using a type of graph structure to represent the necessary information for τ calculation when a certain attribute combination has a value of 1 the knowledge about the target attribute is contained in the structure itself. Each child node of the attribute combination corresponds to the set of permissible values and the target vector associated with this node shows which class is implied by that particular combination of values. As these rules would involve all attributes from the combination a concept hierarchy formation technique [10] could be applied to obtain a comprehensible conceptual hierarchy for the domain. In this case there would be no need for the use of an inductive learning algorithm to obtain the knowledge model. LED domain is excluded from any further discussion.

	C4.5						Back-propagation NN	
	Unpruned		Post-pruned		Highest τ-combination		Full feature set	Highest τ-set
Domains	Size	Accurracy (%)	Size	Accurracy (%)	Size	Accurracy(%)	Accurracy(%)	Accurracy(%)
Breast-cancer	45	93.57	31	93.928	3	91.4286	95.7	89.64
Voting	37	95.977	11	97.7	3	96.55	92.5	96.55
Lenses	7	70	7	70	5	70	70	70
Mushroom	29	100	29	100	10	98.4923	100	98.49
Zoo	17	92.68	17	92.68	11	98.6829	82.9	82.9
LED	19	100	19	100	19	100	100	89.36

Table 2 – Comparison of results obtained when the highest τ-valued attribute combination is used

As it can be seen on the left of table 2, the size of the decision tree has been substantially reduced in all domains. However, in most cases this achievement was at cost of a small reduction in accuracy. An interesting observation is that out of all attribute combinations a single attribute had the highest value in breast-cancer (bare nuclei) and voting (physician-fee-freeze) domains. These attributes do indeed contain the most information for distinguishing the classes of the target attribute. The difference in the accuracy by using the full attribute set is only very small in comparison to the large reduction in tree size. In fact for the voting domain better accuracy was achieved by using single attribute rather than the full attribute set if no post-pruning was applied in the C4.5 algorithm, and the NN achieved better accuracy using only one attribute. The question still remains as to why the attribute combination that would increase the accuracy by this small amount did not have higher τ value than the single attribute. This is due to the fact that when using the τ measure for pre-pruning the value measures the 'total' predictive capability of

attributes and not sequential predictive capability, which is essentially what post-pruning is used for. Total predictive capability refers here to the measure calculated over all classes and instances. To measure the sequential variability in predictive capability of attributes the τ criterion would need to be calculated over a subset of classes and hence instances, which is done in the next section. In an attribute combination the extra information that the combined attributes provide may interfere with the main predicting attribute and hence the τ value is small. It would interfere until some class values are distinguished at which stage this interfering attribute may become useful. In other words some attribute with low τ value may have the necessary constraints to distinguish the remaining instances for which the high τ valued attributes did not have sufficient constraints. This claim is supported by the fact that only after post-pruning was applied for the voting domain the accuracy was higher than by using the single attribute. Furthermore, the attribute set in the voting domain had to be reduced for combining which may have missed some potentially useful combinations. Besides the fact that measuring predictive capability for Boolean combinations of features cannot capture the attributes that become useful once many classes have been distinguished, it can still be very useful to detect the most crucial attributes or combinations for a particular domain. Furthermore, in most of cases the difference in accuracy was not sufficiently high to discard the usefulness of the approach.

4.3 Using the τ criterion for rule simplification

The aim of this section is to demonstrate how the τ criterion can be used as a post-pruning approach for neural networks. Due to its capability of measuring attribute's sequential variation in predictive capability it is used to determine the relevance of an attribute to the rule extracted from a NN. In general the method could be applicable to any rule sets where there are clearly defined attributes values that imply a subset of target classes.

Self-Organizing Map (SOM) [8] is an unsupervised neural network that effectively creates spatially organized "internal representations" of the features and abstractions detected in the input space. It is based on the competition among the cells in the map for the best match against a presented input pattern. Existing similarities in the input space are revealed through the ordered or topology preserving mapping of high dimensional input patterns into a lower-dimensional set of output clusters. When used for classification purposes, SOM is commonly integrated with a type of supervised learning in order to assign appropriate class labels to the clusters. After the supervised learning is complete each cluster will have a rule associated with it, which determines which data objects are covered by that cluster.

For this experiment we have used a slight modification of the original SOM algorithm adjusted so that when used in domains characterized by continuous attributes, rules can be extracted directly from the networks links [5]. Once the rules have been assigned to each cluster the supervised learning starts where a cluster with smallest Euclidean distance to the input instance is activated. Each cluster has a target vector associated with it which is updated every time the cluster is activated. During this process the occurring input and target values have been stored for attributes which define the constraints of the activated cluster. The input values that are close to each other are merged together so that the value object represents a range of values instead. The information collected corresponds to the information contained in a contingency table between an input attribute and the target attribute for the instances captured by the cluster.

The τ criterion has been used for the purpose of removing the links emanating from nodes that are irrelevant for a particular cluster. These links correspond to the attributes whose absence has no effect in predicting the output defined by that cluster. The cluster attributes are ranked according to decreasing τ value. The relevance cut-off occurs at the attribute where the τ value is less than half of the previous attribute's τ value. Note that the τ criterion can only be calculated for cluster attributes that contain more than one value and whose cluster was activated for more than one target class. CSOM is then retrained with all the irrelevant links removed and the aim is that the newly formed clusters will be simpler in terms of attribute constraints.

Initial Clusters			Clusters after pruning and retraining		
C#	Constraints	Target Vector	C#	Constraints	Target Vector
1	0.35 ≤ SL ≤ 0.58 0.24 ≤ SW ≤ 0.58 0.54 ≤ PL ≤ 0.66 0.57 ≤ PW ≤ 0.62	Ivs - 6	1	0.3 ≤ SL ≤ 0.36 0.41 ≤ SW ≤ 0.41 0.58 ≤ PL ≤ 0.59 0.57 ≤ PW ≤ 0.58	Ivs - 2
2	0.63 ≤ SL ≤ 0.64 0.37 ≤ SW ≤ 0.41 0.57 ≤ PL ≤ 0.64 0.5 ≤ PW ≤ 0.54	Ivs- 3	2	0.41 ≤ SL ≤ 0.416 0.29 ≤ SW ≤ 0.29 0.67 ≤ PL ≤ 0.69 0.75 ≤ PW ≤ 0.75	Ivg - 2
3	0.194 ≤ SL ≤ 0.5 0.12 ≤ SW ≤ 0.41 0.33 ≤ PL ≤ 0.627 0.37 ≤ PW ≤ 0.58	Ivs -16	3	0.04 ≤ PW ≤ 0.16	Is - 33
4	0.49 ≤ SL ≤ 0.611 0.2 ≤ SW ≤ 0.41 0.64 ≤ PL ≤ 0.78 0.54 ≤ PW ≤ 0.75	Ivg – 8 Ivs - 1	4	0.25 ≤ SW ≤ 0.75 0.792 ≤ PW ≤ 1	Ivg - 19
5	0.44≤ SL ≤ 0.44 0.41 ≤ SW ≤ 0.5 0.64 ≤ PL ≤ 0.69 0.7 ≤ PW ≤ 0.7	Ivg – 2 Ivs – 1	5	0.36 ≤ SL ≤ 0.694 0.29 ≤ SW ≤ 0.41 0.52 ≤ PL ≤ 0.644 0.49 ≤ PW ≤ 0.58	Ivs - 8
6	0.52 ≤ SL ≤ 0.66 0.33 ≤ SW ≤ 0.58 0.69 ≤ PL ≤ 0.847 0.87 ≤ PW ≤ 1	Ivg -13	6	0.24 ≤ SL ≤ 0.42 0.12 ≤ SW ≤ 0.29 0.42 ≤ PL ≤ 0.576 0.36 ≤ PW ≤ 0.54	Ivs - 9
7	0.778 ≤ SL ≤ 1 0.25 ≤ SW ≤ 0.75 0.831 ≤ PL ≤ 1 0.7 ≤ PW ≤ 0.91	Ivg - 8	7	0.19 ≤ SL ≤ 0.22 0.37 ≤ PW ≤ 0.41	Ivs - 2
8	0.36 ≤ SL ≤ 0.47 0.29 ≤ SW ≤ 0.33 0.66 ≤ PL ≤ 0.69 0.62 ≤ PW ≤ 0.79	Ivg – 3 Ivs - 1	8	0.55 ≤ SL ≤ 0.55 0.2 ≤ SW ≤ 0.29 0.66 ≤ PL ≤ 0.67 0.7 ≤ PW ≤ 0.75	Ivg - 3
9	0.02 ≤ SL ≤ 0.417 0.41 ≤ SW ≤ 0.91 0 ≤ PL ≤ 0.15 0 ≤ PW ≤ 0.16	Is - 32	9	0.417 ≤ SW ≤ 0.5 0.625 ≤ PW ≤ 0.7	Ivg - 7

Table 3: Comparison of initially obtained clusters and clusters after pruning and retraining
Notation: SL – sepal_length, SW – sepal_width, PL – petal _length, PW – petal_width, Ivs – iris-versicolor, ivg – iris-virginica, is – iris-setosa.

The CSOM was trained on the ‘iris’ domain available from the ‘uci’ depository and the comparison of initially obtained clusters and clusters after pruning and retraining is shown in table 3. Please note that the order in which the clusters are displayed in the right column does not reflect the clusters that have been simplified. Due to clarity issues and space limitations the clusters that are only triggered once during supervised learning are excluded from results as they are usually merged into other clusters or deleted due to noise suspicion. As can be seen from table 3 the use of τ criterion for network pruning was successful as the newly obtained clusters (rules) were simplified without increasing the misclassification rate. All the clusters are now implying only one target value and the minimal constraints have been found for certain target classes. Generally speaking a simplified network has better performance and simpler rules are expected to have better generalization power.

5. Conclusion

In this study we have demonstrated some different ways of effectively using the Symmetrical Tau (τ) measure to aid in the feature selection problem. The τ criterion proved to be useful as a filter type approach to feature selection, where in one experiment it was used to filter out single irrelevant attributes, and in other to select the most promising subset of features by determining the predictive capability of feature combinations. The study also gives an example of how the τ criterion can be used for post-pruning in neural networks. The approach simplified the extracted rule set and improved the accuracy by removing the attributes that are irrelevant for a particular output. The experimental results show the effectiveness of the proposed method and indicate its potential as a powerful feature selection criterion in other types of inductive learners.

6. References

[1] Almuallim, H., & Dietterich, T.G. 1991. "Learning with many irrelevant features", *Proceedings of the Ninth National Conference on Artificial Intelligence*, pp. 547-552, San Jose, CA: AAAI Press.

[2] Blake, C., Keogh, E. & Merz, C.J., 1998. "UCI Repository of Machine Learning Databases", Irvine, CA: University of California, Department of Information and Computer Science., 1998. [http://www.ics.uci.edu/~mlearn/MLRepository.html].

[3] Blum, A. & Langley, P. 1997. "Selection of relevant features and examples in machine learning", Artificial Intelligence, Vol 97, Issue 1-2, pp 245-271.

[4] Cherkauer, K.J. & Shavlik, W.J., 1996. "Growing simpler decision trees to facilitate knowledge discovery", *In Proceedings of Second International Conference on Knowledge Discovery and Data Mining*, AAAI Press.

[5] Hadzic, F. & Dillon, T.S., 2005. "CSOM: Self Organizing Map for Continuous Data", *3rd International IEEE Conference on Industrial Informatics (INDIN'05)*, 10-12 August, Perth.

[6] Kira, K. & Rendell, L.A., 1992. "A practical approach to feature selection", Proceedings of the Ninth International Conference on Machine Learning, pp 249-256.

[7] Kohavi, R. & John, G.H., 1997. "Wrappers for feature selection", *Artificial Intelligence*, Vol. 97, issue 1-2, pp 273-324.

[8] Kohonen, T., 1990. "The Self-Organizing Map", *Proceedings of the IEEE*, vol. 78, no 9, pp. 1464-1480.

[9] LeCun, Y., Denker, J. & Solla, S., 1990. "Optimal brain damage", In Touretzky, D.S., ed.: *Advances in Neural Information Processing Systems*, Vol. 2, pp 598-605, San Mateo, CA, Morgan Kauffman.

[10] Sestito, S. & Dillon, S.T., 1994. *Automated Knowledge Acquisition*, Prentice Hall of Australia Pty Ltd, Sydney.

[11] Setiono, R., Leow W.K. & Zurada, J.M., 2002. "Extraction of Rules From Artificial Neural Networks for Noninear Regression." *IEEE Transactions on Neural Networks*, vol. 13, Issue 3, May pp. 564 – 577.

[12] Souza, J., Japkowicz, N., & Matwin, S., 2005. "Feature Selection wuth a General Hybrid Algorithm", in *Proceedings of the Workshop on Feature selection for Data Mining: Interfacing Machine Learning and Statistics* held in conjunction with the 2005 SIAM International Conference on Data Mining, April 23, Newport Beach, CA.

[13] Zhou, X. & Dillon, T.S., 1991. "A statistical-heuristic feature selection criterion for decision tree induction", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no.8, August, pp 834-841.

Concept Identification in Web Pages

Zan Sun
Department of Computer Science
Stony Brook University, Stony Brook, NY 11794, U.S.A.
zsun@cs.sunysb.edu

Abstract

Many online applications (*e.g.,* e-commerce) typically involve a number of concepts. However, identifying those concepts in HTML-based Web pages is hard due to the nature of HTML that lacks semantical annotation of contents. In this paper, we present a technique to analyze Web pages and collect features that capture common observations of each concept including the textual information, structural presentation, and organization. We developed a statistical model coupling these features and our experiments demonstrate the effectiveness of the technique on a large collection of Web pages from various commercial sites.

Keywords: Concept identification, Statistical model, Semantic analysis

1 Introduction

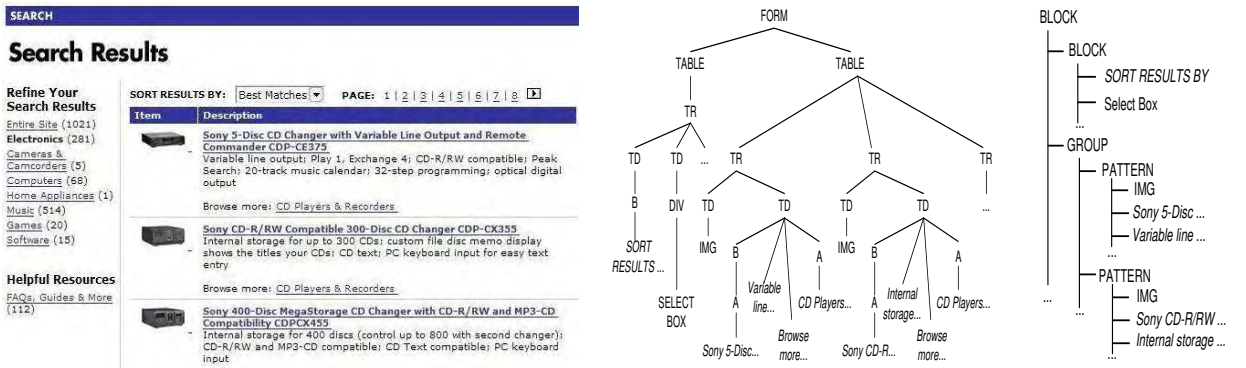
Semantic Web envisions a next-generation information network where content providers define and share machine processable data on the Web. A primary aspect of Semantic Web documents is that they contain metadata to express the meaning of their content. But an enormous amount of extant semantic data (such as product descriptions and pricing information, different categories of news, etc.) is still being encoded in "plain" HTML documents. Although RDF/XML has been widely recognized as the standard vehicle for representing semantic information on the Web, we can extend the reach of Semantic Web to HTML documents by identifying and annotating the (implicit) semantic concepts that are present in their content.

Early solutions [9, 7] to this problem were based on hand-crafted ontologies and graphical ontology/annotation editors that facilitated manual mapping of unlabeled document segments to ontological concepts. From an automation standpoint they are at the "low-degree-of-automation" end of the solution spectrum. The technique in [5] as well as our previous work [15] cover the middle ground wherein document segmentation is done automatically and assignment of semantic labels to these segments is done with manually-crafted ontologies and knowledge bases.

In this paper we describe a concept learning technique for identifying semantic content in Web documents. Our approach is built upon our previous work[14], where starting with a seed of hand-labeled instances of semantic concepts in a set of HTML documents, we bootstrap an annotation process that automatically identifies unlabeled concept instances present in other documents. It uses a combination of structural analysis of the page and machine learning. We enhanced its learning component by a carefully redesign of feature space to learn more robust statistical models of semantic concepts. This redesign constitutes the topic of this paper.

2 Feature based Concept Learning

Our solution consists of three key steps. (i) inferring the logical structure of a Web page via structural analysis of its content, (ii) learning statistical models of semantic concepts using light-weight features extracted from both the content as well as its logical structure in a set of training Web



(a) Search results (b) the segment of DOM Tree (c) partition tree

Figure 1: Search results from Bestbuy

pages, and (iii) applying these models on the logical structures of new Web pages to automatically identify concept instances.

2.1 Structural Analysis

The first step is an improvement of our previous work in [15, 17]. The main purpose of structural analysis is to convert Web pages into hierarchical structures, based on layout and presentation styles. Feature and concept extraction are then performed on the transformed structure.

Structural analysis (see [15] for details) is based upon the observation that semantically related items in content-rich Web pages exhibit consistency in presentation style and spatial locality. An example is shown in Figure 1(a), which is the search result section from Bestbuy (`www.bestbuy.com`) by searching keywords "cd player". Note that all the result items are listed together, and have a consistent presentation style. (*i.e.*, they all begin with an image, followed by the name, a short description and the category information.) Such properties are well captured in the corresponding Document Object Model (DOM) tree that is shown in Figure 1(b). Firstly, all the items are under the same node in the DOM tree; secondly, the similar elements (product images, item names, etc.) have the same root-to-leaf node sequences, where each node is represented by its HTML tag.

We associate *type* information with each node in the DOM tree to reflect the similarities in structural presentation. A *type* of a leaf node is defined to be a pair $< s, p >$, where s is the HTML tag sequence from the root to the leaf, p is the presentation style of the leaf extracted from the attributes of the tags. (*i.e.*, the font size, font color, etc.) An internal node also has a type, which is the sequence of its children's types. Two nodes are equal when their types are identical.

With the above definition, a pattern mining algorithm working bottom-up on DOM tree is used to aggregate the similar segments into subtrees. For each internal node, the algorithm tries to find the most frequently repeating substring (pattern) in the type sequence of the node, and partition the sequence accordingly. Otherwise the type sequence will be propagated to its parent and the same process will be executed at its parent level. For example, let us examine the right `TABLE` tag under the root node in Figure 1(b). Each child of this `TABLE` node has a type sequence of $T_1T_2T_3T_4T_5$, where T_i refers to its leaf child's type. (*e.g.*, for the left-most `TR` node, T_1 refers to the leaf child `IMG` under the node, and T_2 refers to "Sony 5-disc...", etc.). Each `TR` node refers to an item in the result list. Since no repeating pattern can be found here, the types of those `TR` nodes are propagated to the `TABLE` node, which has the type $T_1T_2T_3T_4T_5T_1T_2T_3T_4T_5$.... A pattern

$T_1T_2T_3T_4T_5$ is found now and the sequence is partitioned into sections corresponding to `TR` nodes whose type sequences are $T_1T_2T_3T_4T_5$. For the detail of the algorithm, please refer to [15].

The DOM tree is restructured after the pattern mining algorithm finishes. The restructured tree, also known as *partition tree*, contains three classes of internal nodes: (i) *group* - which encapsulates repeating patterns in its immediate children type sequence, (ii) *pattern* - which captures each individual occurrence of the repeat, or (iii) *block* - when its neither *group* nor *pattern*. Intuitively the subtree of a group node denotes homogenous content consisting of semantically related items. For example, Figure 1(c) shows the corresponding partition tree of the search result section of bestbuy, and observe that how all the items in the search results list in Figure 1(a) are rooted under the group node in the partition tree. The leaf nodes of the partition tree correspond to the leaf nodes in the original DOM tree and have content associated with them.

2.2 Feature Extraction

Our task can be viewed as a typical supervised machine learning problem – given a set of labeled training data, which is a set of subtrees of a concept, learn a classifier which can identify the subtrees of the concept from a new partition tree. Usually in machine learning techniques, each data is mapped to a feature vector $x \in \chi$, where the χ is a real vector space, namely *feature space*.

In our problem, given a subtree rooted at node p in the partition tree, we denote the feature vector $x_p = \langle n_{f_1,p}, n_{f_2,p}, n_{f_3,p}, \ldots \rangle$, where $n_{f_i,p}$ denotes the frequency of occurrence of feature f_i in p. We use three different types of features in the analysis:

2.2.1 Word features

The most intuitive feature for a concept is the textual information that the concept usually contains. For example, the "search result" concept usually has lexicons such as "result", "match" and "sort", etc. In our method, the word features are drawn from the text encapsulated within a partition tree node. For a leaf node in the partition tree, word features are drawn from its own text while for an internal partition tree node, the words present in all the leaves within the subtree rooted at it are aggregated. Stop words are ignored in both cases. $n_{f_i,p}$ is the number of times f_i occurs in the text of p. Word features have been demonstrated successfully in text categorization, which is in common with our problem that particular words are quite possible related to designated concepts.

The disadvantage of word features is that it can't decide boundaries of concepts. For example, the product detail usually contains important keywords such as "price", "specifications", etc. However, in most cases those words occur in a small segment of the Web page while the product details may contain other elements like product pictures, product name and related items. Therefore keywords alone are not enough to decide the proper segment, hence we expand the feature space as follows.

2.2.2 p-gram features

These are the features representing the visual presentation of content. In content-rich Web pages, it is often the case that the presentation of a semantic concept exhibits similarity across sites. For instance, in Figure 1(a), each item is presented as an image, followed by a link with the item name, a short text description, and ending with miscellaneous text information. Similar visual presentation can also be found on other sites. A p-gram feature captures these presentation similarities. The basic p-gram features are *link*, *text*, and *image* found in leaf partition tree nodes. Recall that, during structural analysis, pattern nodes aggregate every individual repeat in a type sequence. Since repeats are typically associated with similar visual presentation, complex p-gram features

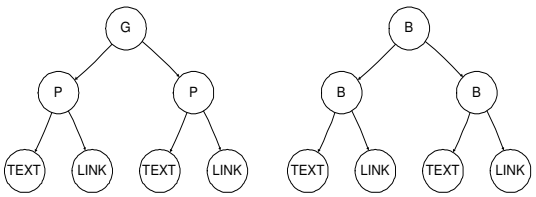


Figure 2: t-gram Features

are constructed only at pattern nodes by concatenating the p-gram features of their immediate children nodes. Internal nodes aggregate basic and possibly complex p-grams from the subtrees rooted at them. Like word features, $n_{f_i,p}$ is the number of times f_i occurs in the subtree rooted at p. For instance, in the left tree of Figure 2, the p-gram feature at the pattern node labeled "P" is $< text \cdot link >$ with the number of occurrences to be 1, while its parent, the group node labeled "G" has the same p-gram feature $< text \cdot link >$ with the occurrences to be 2.

2.2.3 t-gram features

While the visual presentation features are used in presenting the single item like product, the organization of these items are also useful to identify more complex concepts. For example, an instance of "search result" concept usually contains a list of items. However, "a list of items" is not captured using either word features or p-gram features. The t-gram features are used to represent such kind of ideas, i.e., the structure of the partition tree. Recall that internal partition tree nodes can be either group, pattern, or block while link, text, and image are the different classes of leaf nodes. The structural arrangement of these classes of nodes characterize the ideas of "a list of items" or " heterogenous contents". Given a partition tree node with N nodes in its subtree, the complete structural arrangement within the node can be described in terms of a set of subtrees of k ($2 \leq k \leq N$) nodes where each subtree is an arrangement of group, pattern, block, link, text, or image type nodes. Since enumerating all these subtrees has exponential complexity, we restrict our analysis to subtrees of 2 nodes[1]. When $k = 2$ the t-gram is essentially a parent-child feature. For instance, in Figure 2, when $k = 2$ the t-gram feature space of the left tree is $\{< G, P >, < P, Text >, < P, Link >\}$, and the right tree is $\{< B, B >, < B, Text >, < B, Link >\}$, where G and B are labels of group and block nodes respectively.

2.3 Concept Model

A concept model consists of two components: (i) a probability distribution on the frequency of occurrence of the word, p-gram, and t-gram features, and (ii) a probability distribution on the number of nodes present in the entire subtree of a partition tree node. A collection of partition trees whose nodes are (manually) labeled as concept instances serve as training set for learning the parameters of these distributions.

A maximum likelihood approach is used to model the distribution of a feature in a concept. Given a training set of L partition tree nodes identified as instances of concept c_j, the probability of occurrence of a feature f_i in c_j is defined using Laplace smoothing as:

$$P(f_i|c_j) = \frac{\sum_{p \in L} n_{f_i,p} + 1}{\sum_{i=1}^{i=|F|} \sum_{p \in L} n_{f_i,p} + |F|}$$

[1]Bigger sizes of subtrees can be used. From the practice we found 2 is sufficient for our identification problem.

Generic Concepts	Domain-Specific Concepts
Shopping Cart	Search Form
Add To Cart	Search Result
Edit Cart	Item List
Continue Shopping	Item Taxonomy
Checkout	Item Detail

Table 1: Concepts in Ontology.

where $n_{f_i,p}$ denotes the number of occurrences of f_i in partition node p and $|F|$ is the total number of unique feature including word, p-grams, and t-grams. The number of nodes within the subtree of a partition tree node for a concept c_j is modeled as a Gaussian distribution with parameters mean u_{c_j} and variance σ_{c_j} defined as:

$$\mu_{c_j} = \frac{\sum_{p \in L} |p|}{|L|}, \sigma_{c_j} = \sqrt{\frac{\sum_{p \in L}(|p| - \mu_{c_j})^2}{|L| - 1}}$$

For new partition trees, the probability $P(c_j|p)$ of a node p being an instance of concept c_j is proportional to $P(p|c_j)$ assuming an uniform distribution for $P(c_j)$. We use a modified multinomial distribution to model the likelihood $P(p|c_j)$:

$$P(p|c_j) = \left(\frac{\overline{N}!}{N_{f_1,p}! \cdots N_{f_{|F|},p}!}\right) \times \prod_{i=1}^{i=|F|} P(f_i|c_j)^{N_{f_i,p}}$$

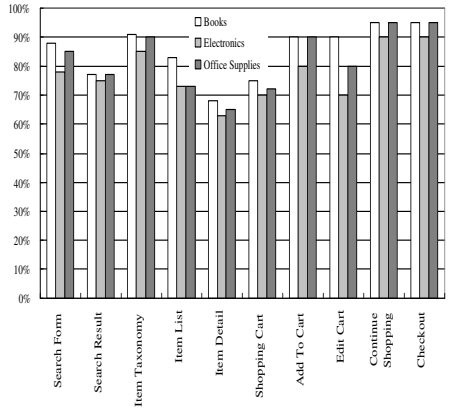
where $\overline{N} = K \times e^{(|p|-\mu_{c_j})^2/(2\sigma^2_{c_j})}$, with K being a normalized total feature frequency count, $|p|$ being the total number of partition tree nodes within the subtree rooted at p, and $N_{f_i,p}$ is a scaled value of $n_{f_i,p}$ such that $\sum_i N_{f_i,p} = \overline{N}$. Note that the above formulation of the likelihood takes into consideration both the *number of nodes* within p as well as the frequencies of the various features in the content encapsulated within p. This results in a tight coupling between content analysis and document structure during concept identification. The node with the maximum likelihood value is identified as the concept instance.

3 Experimental Results

We identified a set of commonly occurring concepts in Web pages, which are shown in Table 1. For each of these concept we built a statistical concept model. The concepts are from three domains, *books*, *electronics* and *office supplies*. The five general concepts in the left column of the table are for all the three domains whereas those in the right column are domain-specific. For instance the feature set of a list of books differs from that of consumer electronic items. We built one model for each concept in the left column of the table and three - one per domain – for each concept in the right column.

To build the model for each of the five generic concepts we collected 90 pages from 15 out of the 30 Web sites. For each of the domain specific concept we collected 30 Web pages from five Web sites that catered to that domain.

Note that pages containing more than one concept were shared during the building of the respective concept models. These models drive the concept extractor at runtime. Since the concept extractor simply chooses the highest scored node as the instance of the desired concept, we measured



Concept	$\theta = -1000$		$\theta = -2000$		$\theta = -3000$	
	Rec%	Prec%	Rec%	Prec%	Rec%	Prec%
Shopping Cart	56.3	100	63.6	91	69.2	84.8
Add To Cart	78.4	100	86.7	86.7	86.7	86.7
Edit Cart	76.6	97.4	79.1	82.3	80	80
Continue Shopping	88.6	100	92.5	92.5	92.5	92.5
Checkout	89.2	100	92	97.4	92.5	92.5
Search Form	63.5	95	74.8	89.2	78.3	86.7
Search Result	55.3	97.6	61.4	89.7	68.9	83.3
Item List	57.8	95	63.2	86.4	68.4	80
Item Taxonomy	65.6	100	73.2	98.8	79.3	93.1
Item Detail	42.3	97.6	49.7	91.8	55.6	83.4

Figure 3: (a) Recall for 3 domains. (b) Average Recall and Precision

the recall[2] of the concept extractor for each concept in the ontology. The precision[3] was also measured when we artificially set a threshold θ for each concept model, i.e., the node is an instance of the concept only if its score is the highest among all the nodes and greater than the threshold. Roughly 150 Web pages collected from all of these 30 Web sites were used as the test data. To label the concept, we use the log of the likelihood of each node as its score for more precision. Figure 3(a) shows the recall values for all of the 10 concepts in each of the three domains. Figure 3(b) lists the average recall and precision values over three domains when different thresholds were used.

An examination of the Web pages used in the testing revealed that the high recall rates (above 80% for "Item Taxonomy", "Search Form", "Add To Cart", "Edit Cart", "Continue Shopping" and "Checkout") are due to the high degree of consistency of the presentation styles of these concepts across all these Web sites. The low recall figures for the "Item Detail" (about 65% averaged over the three domains) and "Shopping Cart" (about 70%) are mainly due to the high degree of variation in their features across different Web sites. A straightforward way to improve the recall of such concepts is to use more training data. However even this may not help for concepts such as "Add To Cart" that rely on keywords as the predominant feature. Quite often these are embedded in a image precluding textual analysis. It appears that in such cases local context surrounding the concept can be utilized as a feature to improve recall.

It also shows that the thresholds have great impact on precision and recall values for the concepts that vary a lot from page to page. When the threshold is high, (e.g., $\theta = -1000$) the recall rates are quite low (around 50%). The high precision due to highly presentation consistency of the instances. Thus only those instances very similar to the training data can have scores above the threshold. Lowering the threshold can raise the recall rate but decrease the precision.

4 Related Works

The essence of the technique underlying our structural analysis module is to partition a page into segments containing "semantically" related items and classify them against concepts in the ontology.

[2]Recall value for a concept is the ratio of the number of correctly labeled concept instances in Web pages over the actual number of concept instances present in them.

[3]Precision value for a concept is the ratio of the number of correctly labeled concept instances in Web pages over the number of all the labeled concept instances in them.

Web page partitioning techniques have been proposed for adapting content on small screen devices [2, 3, 22], content caching [18], data cleaning [19, 21], and search [23]. The fundamental difference between our technique and all the above works is the integration of inferring a page's logical structure (*e.g.*, the partition tree in Figure 1(c)) with feature learning. This allows us to define and learn features, such as p-grams and t-grams, using partition trees.

Learning a concept model from training examples and using this model for detecting instances in documents is closely related to work done on categorization techniques, including Bayesian approaches [13, 12], and topic detection[1]. The fundamental difference between the problem of semantic annotation and text categorization is that in the former a single document can contain instances of multiple concepts while categorization assigns a single concept (class) to the entire document. Consequently, unlike any work in text categorization, in the annotation problem we will have to infer the presence of multiple concept instances in a single HTML document. Moreover our work is also concerned with inferring the *logical organization* of a HTML document - the concept hierarchy - which is not addressed in either text classification or topic detection. Also text categorization methods do not exploit the (presentation) structure of a document for inducing features (see [20] for a survey on feature selection in text categorization). We do that and as our experimental results indicate they are critical for boosting the precision of concept identification.

Concept identification in Web pages is also related to the body of research on semantic understanding of Web content. Powerful ontology management systems and knowledge bases have been used for interactive annotation of Web pages [9, 10]. More automated approaches combine them with linguistic analysis [16], segmentation heuristics [5, 6], and machine learning techniques [4, 8]. Our semantic analysis technique is an extension of our previous work [14] and, in contrast to all the above, does not depend on rich domain information. Instead, our approach relies on light-weight features in a machine learning setting for concept identification. This lets users define *personalized* semantic concepts thereby lending flexibility to modeling Web transactions.

It should also be noted that the extensive work on wrapper learning [11] is related to concept identification. However, wrappers are syntax-based solutions and are neither scalable nor robust when compared to semantics-based techniques.

5 Conclusion

In this paper we presented our method to learn and identify instances of concepts from Web pages. Our concept identification method has been successfully applied to our Guide-O system that is to facilitate online transaction under constraints. The method significantly improved the overall performance of the system by extracting only the related instances of concepts from Web pages. Other applications include information retrieval, assistive browsing and also data annotation for Semantic Web, etc.

References

[1] J. Allan, editor. *Topic Detection and Tracking: Event-based Information Organization.* Kluwer Academic Publishers, 2002.

[2] O. Buyukkoten, H. Garcia-Molina, and A. Paepcke. Seeing the whole in parts: Text summarization for web browsing on handheld devices. In *Intl. World Wide Web Conf. (WWW)*, 2001.

[3] Y. Chen, W.-Y. Ma, and H.-J. Zhang. Detecting web page structure for adaptive viewing on small form factor devices. In *Intl. World Wide Web Conf. (WWW)*, 2003.

[4] P. Cimiano, S. Handschuh, and S. Staab. Towards the self-annotating web. In *Intl. World Wide Web Conf. (WWW)*, 2004.

[5] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. Tomlin, and J. Yien. SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. In *Intl. World Wide Web Conf. (WWW)*, 2003.

[6] M. Dzbor, J. Domingue, and E. Motta. Magpie - towards a semantic web browser. In *Intl. Semantic Web Conf. (ISWC)*, 2003.

[7] D. Fensel, S. Decker, M. Erdmann, and R. Studer. Ontobroker: Or how to enable intelligent access to the WWW. In *11th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Canada, 1998.

[8] B. Hammond, A. Sheth, and K. Kochut. Semantic enhancement engine: A modular document enhancement platform for semantic applications over heterogenous content. In V. Kashyap and L. Shklar, editors, *Real World Semantic Applications*. IOS Press, 2002.

[9] J. Heflin, J. A. Hendler, and S. Luke. SHOE: A blueprint for the semantic web. In D. Fensel, J. A. Hendler, H. Lieberman, and W. Wahlster, editors, *Spinning the Semantic Web*, pages 29–63. MIT Press, 2003.

[10] J. Kahan, M. Koivunen, E. Prud'Hommeaux, and R. Swick. Annotea: An open rdf infrastructure for shared web annotations. In *Intl. World Wide Web Conf. (WWW)*, 2001.

[11] A. Laender, B. Ribeiro-Neto, A. da Silva, and J. Teixeira. A brief survey of web data extraction tools. *SIGMOD Record*, 31(2), 2002.

[12] D. Lewis, R. Schapire, J. Callan, and R. Papka. Training algorithms for linear text classifiers. In *ACM Conf. on Informaion Retrieval (SIGIR)*, 1996.

[13] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI Workshop on Learning for Text Categorization*, 1998.

[14] S. Mukherjee, I. Ramakrishnan, and A. Singh. Bootstrapping semantic annotation for content-rich html documents. In *Intl. Conf. on Data Engineering (ICDE)*, 2005.

[15] S. Mukherjee, G. Yang, W. Tan, and I. Ramakrishnan. Automatic discovery of semantic structures in html documents. In *Intl. Conf. on Document Analysis and Recognition*, 2003.

[16] B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff, and M. Goranov. Kim - semantic annotation platform. In *Intl. Semantic Web Conf. (ISWC)*, 2003.

[17] I. Ramakrishnan, A. Stent, and G. Yang. Hearsay: Enabling audio browsing on hypertext content. In *Intl. World Wide Web Conf. (WWW)*, 2004.

[18] L. Ramaswamy, A. Iyengar, L. Liu, and F. Douglis. Automatic detection of fragments in dynamically generated web pages. In *Intl. World Wide Web Conf. (WWW)*, 2004.

[19] R. Song, H. Liu, J.-R. Wen, and W.-Y. Ma. Learning block importance models for web pages. In *Intl. World Wide Web Conf. (WWW)*, 2004.

[20] Y. Yang and J. Pedersen. A comparative study on feature selection in text categorization. In *Intl. Conf. on Machine Learning (ICML)*, 1997.

[21] L. Yi and B. Liu. Eliminating noisy information in web pages for data mining. In *ACM Conf. on Knowledge Discovery and Data Mining (SIGKDD)*, 2003.

[22] X. Yin and W. S. Lee. Using link analysis to improve layout on mobile devices. In *Intl. World Wide Web Conf. (WWW)*, 2004.

[23] S. Yu, D. Cai, J.-R. Wen, and W.-Y. Ma. Improving pseudo-relevance feedback in web information retrieval using web page segnmentation. In *Intl. World Wide Web Conf. (WWW)*, 2003.

A Study of Multi-Objective Fitness Functions for a Feature Selection Genetic Algorithm

Márcio Porto Basgalupp, Karin Becker, Duncan D. A. Ruiz
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre – RS – Brazil
{basgalupp, kbecker, duncan}@inf.pucrs.br

Abstract: *The use of genetic algorithms for feature selection in classification problems is becoming widely accepted. One of the striking advantages of this approach is how easily different solution evaluation criteria can be combined and embedded in fitness functions, thus allowing the evaluation of the trade-offs with regard to different criteria. This paper reports a study on the contribution of three different classification-model evaluation criteria (i.e. accuracy, number of selected features and tree size). Although works in literature already propose different multi-objective fitness functions, the results reported in these works cannot be compared, since they were produced using different genetic algorithms and classifiers. The present work compares the results obtained by the use of different fitness functions, considering that all other genetic algorithm properties remain constant. For this purpose, a genetic algorithm was developed and tested with four fitness functions using classical datasets. Preliminary results revealed the contribution of each criterion, as represented by the respective fitness functions, displaying the advantages of a function that combines the three criteria.*

Key words: *genetic algorithm, model evaluation criteria, multi-objective fitness function.*

1. Introduction

In real-world classification problems, the relevant features (predictive attributes) for the determination of the class attribute are hardly known a priori. Therefore, many candidate features are introduced in the dataset in order to better represent the domain [1], many of which are redundant or irrelevant, hence jeopardizing the classification process. Feature selection is the process of identifying and removing irrelevant and redundant features, as much as possible, from a data set [2]. The striking advantages of feature selection are the improvement of the: a) quality of available data, because irrelevant and redundant features are removed; b) execution performance of data mining algorithms, as a consequence of the reduction of data dimensionality; and c) results yielded by the data mining step, because provided information is more significant with regard to models characterization [1][3][4].

In general, feature selection algorithms have two main components [1][3]: a search component, which generates candidate features subsets; and an evaluation component that measures the quality of each candidate subset. Feature selection algorithms are also controlled by a stopping criterion, which can act either over the search component (e.g. maximal number of generations) or the evaluation component (i.e. goodness of the candidate subset). According to [3], feature selection methods can be classified into two groups: filter and wrapper. In the classification context, addressed in this paper, wrapper methods make use of measures yielded by the classification process in the evaluation component. Different classification methods (e.g. decision tree, neural network) or classifier implementations of a same method (e.g. ID3 or C4.5 for decision tree) can be used.

Ideally, algorithms for the search component should generate all possible feature subsets in order to find the best one. However, this exhaustive process is impracticable even for a medium-size feature set. Therefore, the development of heuristic-based and random search methods aims at reducing the search computational complexity. In order to produce satisfactory results, such random/heuristic-based search components need to be well integrated with the evaluation component.

Genetic algorithms are random search algorithms based on the mechanisms of natural selection and genetics [5]. They are capable of evolving solutions of real world problems and are considered a very attractive approach for the search of sub-optimal solutions in optimization problems. Differently from

other methods based on simple random walks over the solution space, genetic algorithms use random choice to guide a highly explorative search. Relying in hazard to achieve good results may seem unusual, but nature has proven this is a successful approach. Figure 1 depicts the basic structure of a genetic algorithm. A possible candidate solution is referred to as a chromosome or individual. The algorithm establishes a loop, by generating a population and evaluating the "goodness" of each candidate solution using a fitness function. If the stopping criterion is not met, the population evolves using genetic operators (e.g. selection, crossover), thus generating new possible solutions. The most common stopping criteria are the number of evolutions and the identification of an individual with a satisfactory fitness value.

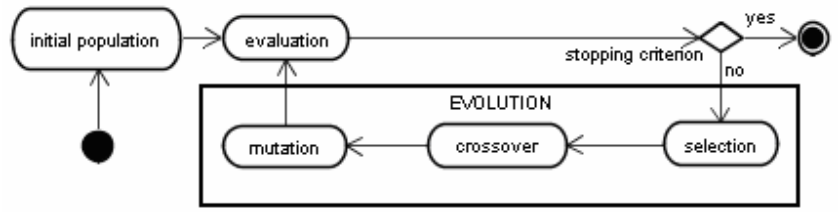


Figure 1: Activity diagram representing a basic structure of a genetic algorithm

Most works addressing the use of genetic algorithms for feature selection in classification problems propose wrapper methods [4][6][7][8]. One of the main advantages is how easily different evaluation criteria can be combined in fitness functions, thus allowing the evaluation of the trade-offs with regard to criteria such as classifier accuracy, number of selected features, model complexity, feature cost, among others. Although accuracy is a very important criterion, in many application domains other criteria need to be weighted with it in order to guarantee model interpretability [7], least features cost [4], etc.

The goal of this paper is to develop a study on the contribution of different classification model evaluation criteria, as these are combined in fitness functions. Although works in literature already propose different multi-objective fitness functions, the results reported in these works cannot be compared, since they were produced using different genetic algorithms and classifiers. The present work compares the results obtained by the use of different fitness functions, considering that all other genetic algorithm properties remain constant. For this purpose, a genetic algorithm was developed and tested with four fitness functions using classical datasets [9][10].

The remaining of this paper is structured as follows. Section 2 describes related work. Section 3 details the implementation of genetic algorithm developed, together with the four fitness functions. Section 4 discusses experimental results. Section 5 draws conclusions and presents future works.

2. Related Work

In this section, related work that uses genetic algorithm as the a wrapper method for feature selection in classification problems are described, emphasizing their main properties, namely fitness function, classifier type, chromosome representation, initial population generation method, and population evolution methods. A summarized comparison is displayed in Table 1.

Table 1 – Comparison of Genetic Algorithms for Feature Selection.

	Yang and Honavar	Sun et. al.	Cantú-Paz	Cherkauer & Growing
Encoding	Binary string	Binary string	binary string	Genome
Initial Population	not informed	Random	Filter	not informed
Population Evolution	Ranking, other genetic operations are not informed	Cross generational + uniform crossover	Tournament + uniform crossover	roulette + uniform crossover
Fitness Evaluation	accuracy + cost	Accuracy + Number of features	Accuracy	Accuracy + Number of features + number of Nodes
Classifier	Neural Network	Bayes, RN, SVM and LDA	Bayes	Decision tree

a. Cantú-Paz [8]

The wrapper method proposed in [8] uses a Bayes classifier. The fitness function is composed solely by *accuracy* criterion. An interesting contribution of this method is that it integrates also feature selection filtering methods to generate the initial population, thus reducing the number of redundant attributes. Chromosomes are represented by binary strings of dimension *n*, where *n* is the number of features, and bit value 1 corresponds to a selected feature. The selection method of individuals for population evolution is tournament. Experiments revealed it has outperformed the other methods used for comparison in terms of accuracy, the criterion used in the fitness function. However, with regard to the criterion *number of selected features*, it did not present the best performance. Other properties of the yielded classification models were not examined.

b. Sun et al. [6]

The feature selection genetic algorithm proposed in [6] uses four different classifiers for the gender classification from frontal facial images application, namely Bayes, neural networks, SVM (Support Vector Machine) and LDA (Linear Discriminant Analysis). The main goal is to improve the accuracy of the yielded classification model, and for that, two criteria are combined in a multi-objective fitness function, namely *accuracy* and *number of selected features*. The fitness function is given by:

$$fitness(x) = 10^4\, accuracy(x) + 0{,}4 zeros(x)$$

where $accuracy(x)$ is the accuracy rate generated by the classifier for the feature subset represented by chromosome x, and $zeros(x)$ is the number of features not selected in subset x. The goal is to choose the smaller feature subset x within subsets with similar accuracy, which justifies the weight given to the criterion number of features, in comparison to accuracy.

Chromosome representation is also a fixed length binary string. A random method for initial population is employed which minimizes the chance of consistently generating individuals representing approximately half of the possible attributes. In order to explore subsets of different number of features, the method generates randomly the numbers of 1's for each individual. Then, these 1's are randomly scattered in the chromosome. The selection strategy is cross-generational.

Experimental results compare the features selected by the genetic algorithms, considering the different classifiers, with the ones selected by an expert. Accuracy has significantly increased for all classifiers. The number of features also reduced significantly, and selected attributes were very relevant, varying from 32 to 67% of the relevant classification information, according to the expert's opinion.

c. Yang and Honavar [4]

A method that uses neural network-based classifier (DistAL) is presented in [4]. Chromosomes are also represented by fixed length binary strings. Nothing is mentioned about how the initial population is generated, and the selection method of individuals for population evolution is ranking. An interesting contribution of this work is the consideration of the *cost of attributes* during feature selection, by arguing that such criterion is important in many domains (e.g. in the medical domain, it is best to classify a disease based on blood samples, than on tomography results). The multi-objective fitness function is given by:

$$fitness(x) = accuracy(x) - \frac{\cos t(x)}{accuracy(x) + 1} + \cos t_{\max}$$

where $fitness(x)$ is the fitness of the feature subset represented by x, $accuracy(x)$ is the accuracy of the classification model yielded using the feature subset x, $cost(x)$ is the sum of the costs of feature subset x, and $cost_{max}$ is the sum of the costs associated with all of the features. Considering the weights assigned, it encourages the selection of a reasonable solution that yield high accuracy at a moderate cost ([4]). The presented results revealed the efficiency of the proposed approach, which improves both accuracy and quality of the classification model, measured in this case in terms of hidden nodes of the neural network.

The authors stress the difficulty of obtaining attribute costs, and experimental results were very limited due to this reason. The algorithm did not perform well with regard to the *number of selected features*.

d. Cherkauer and Growing [7]

SET-Gen [7] is a feature selection genetic algorithm targeted at improving the interpretability of decision trees built using C4.5 algorithm [11], without compromising *accuracy*. A specific chromosome representation named genome is used, in which each gene can represent a feature, given that the same feature can be represented in different genes of the same chromosome. This representation aims at: 1) slowing the potential loss of diversity that tends to occur during genetic search; 2) allows the definition of the maximum number of features to be selected, by the use of string length. The fitness function is also multi-objective, establishing trade-offs between *number of features*, *tree size*, and *accuracy*, as given by:

$$fitness(x) = \frac{3}{4}A(x) + \frac{1}{4}(1 - \frac{S(x) + F(x)}{2})$$

where *A(x)* is the accuracy of feature subset *x*, *S(x)* is the average size of the decision tree produced by C4.5 for the feature subset *x*, and *F(x)* is the number of features of *x*. The weights given to these criteria are not justified in [7].

SET-Gen algorithm uses traditional operations of mutation and crossover for population evolution, and introduces a new operator called feature removing, which eliminates all incidences of a specific feature. Chromosomes are selected by roulette method to form the next generations of the population. Initial population generation method is not mentioned. Experiments revealed that, in comparison with the classification models produced by C4.5, SET-Gen reduced significantly the complexity of created trees and the number of selected features. However, a slight accuracy improvement was detected, which was not statistically significant according *t*-test.

3. A Multi-Function Genetic Algorithm for Feature Selection

In the previous section, several wrapper feature selection methods were discussed. They stress the importance of different criteria, other than accuracy, to reach a satisfactory solution for the feature selection problem. All these works develop experiments that highlight the contribution of the evaluation criteria embedded in the fitness function. However, the fitness function alone is not responsible for the choice of a given feature subset solution by the genetic algorithm. Indeed, all proposals vary on the implementation strategies and configurations of the genetic algorithm, as well as on the classifiers used to produce data for evaluation. Hence, it is not possible to establish a comparison on the effects that each criterion has over the selection of a feature subset solution.

This work compares the results obtained by fitness functions that weights different criteria, considering that all other genetic algorithm properties remain constant. For this purpose, a genetic algorithm for feature selection was developed, of which the striking properties are described in the remaining of this section. A prototype was developed using Java programming language.

a. Encoding

As in [4][6][8], a chromosome is represented by a fixed length binary string, where its size is the total number of features. This encoding strategy makes easier the application of genetic operators.

b. Initial Population

The method of [6] was adopted, which is a random method that minimizes the chance of consistently generating individuals representing approximately half of the possible attributes.

c. Fitness Evaluation

The adopted criteria were *accuracy*, *number of selected features* and *tree size*. Although interesting, *cost* criterion could not be employed due to the difficulty on finding datasets for which the

costs associated to the attributes were available, which is one of the difficulties found in [4] to validate the proposed fitness function. These three criteria were arranged in four different fitness functions:

$$fitness_1(x) = accuracy(x) \quad (1)$$

$$fitness_2(x) = accuracy(x) - 0{,}2FeaturesSelected(x) \quad (2)$$

$$fitness_3(x) = accuracy(x) - 0{,}2TreeSize(x) \quad (3)$$

$$fitness_4(x) = accuracy(x) - 0{,}2FeaturesSelected(x) - 0{,}2TreeSize(x) \quad (4)$$

where *accuracy*(*x*) represents the classifier accuracy (10-fold cross-validation) considering chromosome *x*; *FeaturesSelected*(*x*) is the number of features in chromosome *x,* and *TreeSize*(*x*) is the number of nodes of the decision tree generated from *x*. The weights were assigned empirically, prioritizing accuracy.

d. Population Evolution

The techniques chosen for population evolution are classical and widely used in genetic algorithms [5]. The selection of the chromosomes to be used for next generation population is performed using roulette, a method in which the probability of selecting a given chromosome is directly proportional to its fitness. The new chromosomes are produced by 2-points crossover. Elitism theory was also employed, and the implementation inserts directly in the next population the best two chromosomes of the previous one (i.e. the ones with highest fitness). Mutations are performed over a chromosome according to mutation probability, by selecting a random number of gene pairs, and then, arbitrarily selecting genes exchange positions. The new mutated genes are also copied to the next population. The number of evolutions is used as stopping criterion. Crossover and mutation probabilities, together with maximal number of evolutions, are parameters provided by the user.

4. Experiments

The data sets for the experiments were selected from the UCI Machine Learning Repository [9] and UCI KDD Archive [10]. The classifier was the J48 algorithm available in Weka framework (version 3.5.1), which is an implementation of the C4.5 classifier [12]. The default parameters proposed by Weka were adopted. This implies on the use of a 10 fold cross-validation to produce accuracy values.

Initially, the selected datasets were classified using J48. Table 2 displays the properties of the selected datasets, together with accuracy and tree size of the corresponding classification models, which are the relevant model properties for our purposes. Then, each dataset was input to a specific configuration of the genetic algorithm, where a configuration represents one of the four fitness functions presented in Section 3. From now on, these are referred to as configurations $fitness_1$, $fitness_2$, $fitness_3$ and $fitness_4$., as described by formulae 1, 2, 3 and 4, respectively. We run each algorithm configuration with each dataset 5 times, in order to obtain the average of the measures analyzed. The measures captured were execution time, number of selected features selected, accuracy and tree size. The genetic algorithm was configured with population size 20, 10 evolutions, 0.9 probability of crossover and 0.05 probability of mutation.

Table 2: Data set properties and classification model results without feature selection.

Data Set	Instances	Number of Features	Classes	Accuracy without Feature Selection (%)	Tree Size
Anneal	898	38	6	98,44	47
Bands	540	39	2	70,19	7
Credit-g	1000	20	2	70,50	140
Credit-a	690	15	2	86,09	42
Labor	57	16	2	73,68	5
Colic	368	22	2	85,33	6
Autos	205	26	7	81,95	69
Arrhythmia	452	279	16	64,38	99

Tables 3 present the results obtained by genetic algorithm configuration *fitness*$_1$ and *fitness*$_2$, and Table 4, for *fitness*$_3$ and *fitness*$_4$. The tables display the mean values and standard deviation, considering the 5 executions. Execution time is not depicted in the tables because it did not revealed interesting results with regard to the comparison of the fitness functions. Figures in bold represent the best results according to a given criterion, comparing the results yielded by the 4 fitness functions. Tables 5, 6 and 7 compare these results for criteria *accuracy*, *number of selected features* and *tree size*, respectively.

Table 3: Results from *fitness*$_1$ (only accuracy) and *fitness*$_2$ (accuracy, number of features).

Data Set	*fitness*$_1$			*fitness*$_2$		
	Selected Features	**Accuracy**	**Tree Size**	**Selected Features**	**Accuracy**	**Tree Size**
Anneal	20,6 ± 3,78	**98,53 ± 0,44**	46,4 ± 11,57	**18,2 ± 6,02**	98,46 ± 0,92	44,6 ± 16,5
Bands	15,6 ± 3,51	79,96 ± 0,94	91 ± 9,46	19,8 ± 3,9	**80,85 ± 0,9**	114,6 ± 15,13
credit-g	7,8 ± 3,56	75,08 ± 0,64	66,8 ± 45,87	8,2 ± 2,59	**75,16 ± 0,92**	86,8 ± 13,99
credit-a	8,2 ± 1,1	**86,84 ± 0,81**	26 ± 16,17	6 ± 2,65	86,55 ± 1,02	19,2 ± 18,85
labor	7,4 ± 2,07	**89,12 ± 2,29**	5,8 ± 2,59	7,4 ± 2,7	88,07 ± 3,14	6 ± 2
colic	8,4 ± 3,44	86,03 ± 0,15	7,6 ± 1,52	11,4 ± 2,41	86,2 ± 0,45	10,6 ± 6,02
autos	15 ± 1	**84,1 ± 1,22**	71,2 ± 17,02	**11,8 ± 5,5**	83,61 ± 3,12	65,8 ± 6,83
arrhythmia	156,8 ± 12,36	**70,4 ± 0,96**	82,6 ± 4,56	33,6 ± 6,84	67,04 ± 3,04	85,4 ± 9,94

Table 4: Results from *fitness*$_3$ (accuracy, tree size) and *fitness*$_4$ (accuracy, number of features, tree size).

Data Set	*fitness*$_3$			*fitness*$_4$		
	Selected Features	**Accuracy**	**Tree Size**	**Selected Features**	**Accuracy**	**Tree Size**
Anneal	22,6 ± 2,3	98,35 ± 0,71	38,2 ± 5,54	20,2 ± 3,27	98,2 ± 1,17	**37 ± 7,97**
Bands	19,2 ± 3,56	70,44 ± 0,17	**3 ± 0**	**14,6 ± 6,66**	71,07 ± 1,1	5,4 ± 2,19
credit-g	**2 ± 1,58**	71,36 ± 1,35	5,8 ± 5,07	8,62 ± 4,65	70,34 ± 0,76	**2 ± 2,23**
credit-a	4,6 ± 1,67	85,57 ± 0,08	**3 ± 0**	**3,4 ± 0,55**	85,51 ± 0	**3 ± 0**
labor	7,6 ± 1,67	87,37 ± 1,92	7,2 ± 2,39	**4,6 ± 2,51**	89,12 ± 3,14	**5,6 ± 2,51**
colic	12,4 ± 1,95	85,98 ± 0,15	**6 ± 0**	**6 ± 1,87**	**86,41 ± 0,64**	7,6 ± 2,07
autos	15,4 ± 0,89	83,41 ± 1,72	55,2 ± 1,79	13,8 ± 2,77	81,17 ± 2,69	**50,4 ± 4,04**
arrhythmia	137,2 ± 36,3	70,27 ± 0,95	78,6 ± 6,23	**31,8 ± 11,97**	65,62 ± 3,99	**69,4 ± 12,76**

As depicted in Table 5, the feature selection genetic algorithm improved the *accuracy* of the resulting classification model in almost all cases, considering all fitness functions, when compared to the *accuracy* with no feature selection. This result confirms expected benefits from feature selection methods. In general, *fitness*$_1$ revealed the best accuracy results. The cases in which *accuracy* has not improved are related to the following datasets and algorithm configurations: anneal (*fitness*$_3$ and *fitness*$_4$), credit-g (*fitness*$_4$), credit-a (*fitness*$_3$ and *fitness*$_4$), and autos (*fitness*$_4$). Actually, these cases display very similar figures, and the difference was not considered as statistically significant, according to *t*-test. These results can be explained because *fitness*$_3$ and *fitness*$_4$ establish tradeoffs between accuracy and tree size. The latter yields more easily interpreted decision tree models, which is very important for many applications.

Configuration *fitness*$_2$ did not produce consistently the least *number of selected features*, as it can be seen in Table 6. Indeed, in general, the best results for this criterion were revealed by *fitness*$_4$, that in addition considers *tree size*. A possible explanation is the weight assigned to the *number of selected features* in *fitness*$_2$. Since *tree size* and *number of selected features* are somehow related, it is comprehensible that the weight indirectly assigned to this criterion is bigger in *fitness*$_4$ than in *fitness*$_2$. Nevertheless, even if *fitness*$_2$ did not yield the least number of selected features, the results for the corresponding *accuracy* are very satisfactory, and they are statistically comparable to accuracy considering all other fitness functions, according to *t*-test. As for *tree size* criterion, this fitness function yielded poor results, thus jeopardizing the interpretability of the model.

With regard to the *tree size* criterion (Table 7), configuration *fitness*$_4$ outperformed *fitness*$_3$. This result may also be explained by the indirect weight given to criterion *number of selected features* in

fitness$_4$. Configuration *fitness*$_3$ also presented a poor performance on the other criteria, displaying in general inferior results for number of selected features, and statistically comparable ones for accuracy.

Table 5: For each data set, accuracy obtained with each fitness function.

	Accuracy				
Data Set	**Without feature selection**	**fitness$_1$**	**fitness$_2$**	**fitness$_3$**	**fitness$_4$**
Anneal	98,44	**98,53 ± 0,44**	98,46 ± 0,92	98,35 ± 0,71	98,2 ± 1,17
Bands	70,19	79,96 ± 0,94	**80,85 ± 0,9**	70,44 ± 0,17	71,07 ± 1,1
credit-g	70,50	75,08 ± 0,64	**75,16 ± 0,92**	71,36 ± 1,35	70,34 ± 0,76
credit-a	86,09	**86,84 ± 0,81**	86,55 ± 1,02	85,57 ± 0,08	85,51 ± 0
labor	73,68	**89,12 ± 2,29**	88,07 ± 3,14	87,37 ± 1,92	89,12 ± 3,14
colic	85,33	86,03 ± 0,15	86,2 ± 0,45	85,98 ± 0,15	**86,41 ± 0,64**
autos	81,95	**84,1 ± 1,22**	83,61 ± 3,12	83,41 ± 1,72	81,17 ± 2,69
arrhythmia	64,38	**70,4 ± 0,96**	67,04 ± 3,04	70,27 ± 0,95	65,62 ± 3,99

Table 6: For each data set, number of selected features obtained with each fitness function.

		Number of Selected Features			
Data Set	**total**	**fitness$_1$**	**fitness$_2$**	**fitness$_3$**	**fitness$_4$**
Anneal	38	20,6 ± 3,78	**18,2 ± 6,02**	22,6 ± 2,3	20,2 ± 3,27
Bands	39	15,6 ± 3,51	19,8 ± 3,9	19,2 ± 3,56	**14,6 ± 6,66**
credit-g	20	7,8 ± 3,56	8,2 ± 2,59	**2 ± 1,58**	8,62 ± 4,65
credit-a	15	8,2 ± 1,1	6 ± 2,65	4,6 ± 1,67	**3,4 ± 0,55**
labor	16	7,4 ± 2,07	7,4 ± 2,7	7,6 ± 1,67	**4,6 ± 2,51**
colic	22	8,4 ± 3,44	11,4 ± 2,41	12,4 ± 1,95	**6 ± 1,87**
autos	26	15 ± 1	**11,8 ± 5,5**	15,4 ± 0,89	13,8 ± 2,77
arrhythmia	279	156,8 ± 12,36	33,6 ± 6,84	137,2 ± 36,3	**31,8 ± 11,97**

Table 7: For each data set, tree size obtained with each fitness function.

	Tree Size				
Data Set	**Without feature selection**	**fitness$_1$**	**fitness$_2$**	**fitness$_3$**	**fitness$_4$**
Anneal	47	46,4 ± 11,57	44,6 ± 16,5	38,2 ± 5,54	**37 ± 7,97**
Bands	7	91 ± 9,46	114,6 ± 15,13	**3 ± 0**	5,4 ± 2,19
credit-g	140	66,8 ± 45,87	86,8 ± 13,99	5,8 ± 5,07	**2 ± 2,23**
credit-a	42	26 ± 16,17	19,2 ± 18,85	**3 ± 0**	**3 ± 0**
labor	5	5,8 ± 2,59	6 ± 2	7,2 ± 2,39	**5,6 ± 2,51**
colic	6	7,6 ± 1,52	10,6 ± 6,02	**6 ± 0**	7,6 ± 2,07
autos	69	71,2 ± 17,02	65,8 ± 6,83	55,2 ± 1,79	**50,4 ± 4,04**
arrhythmia	99	82,6 ± 4,56	85,4 ± 9,94	78,6 ± 6,23	**69,4 ± 12,76**

As it can be seen in Table 4, the best results with regard to ensemble of the three criteria were yielded by configuration *fitness*$_4$. It outperformed on *number of selected features* and *tree size*, without a significant degradation on the *accuracy*. As mentioned, the trade-off between accuracy and interpretability is necessary in many domains. Perhaps with different weights, this fitness function would outperform with regard to all criteria.

Besides, in Bands dataset, the higher accuracy figures are related to very big trees (*fitness*$_1$ and *fitness*$_2$), which are due to the use of categorical attributes with many values, and which, apparently, have a strong influence on accuracy. This explains why even a slight reduction with regard to the original tree size has a strong, negative impact on accuracy (*fitness*$_3$ and *fitness*$_4$), when compared to the good results obtained by *fitness*$_1$ and *fitness*$_2$. Another issue is the high standard deviation observed in some cases for *number of selected features* and *tree size*. We believe that this issue is partly explained by the random

character of the solutions yielded by genetic algorithms and partly by the weight assigned to these criteria in the fitness function. This belief is grounded on the fact that this situation occurs when at least one of the criterion that negatively influences the function (i.e. *tree size* and *number of selected features*) has a high value. The assignment of relative weights to these criteria could be a solution to this problem.

5. Conclusions and future work

This paper developed a study on the effects of three different classification model evaluation criteria, namely accuracy, number of selected features and tree size, which were combined differently in multi-objective fitness functions. For this purpose, a genetic algorithm was developed and tested with four fitness functions using classical datasets. This algorithm employs classical implementation strategies. The initial population method proposed in [6] was also adopted. The contribution of genetic algorithms for feature selection applications was confirmed, and this study revealed how easily new evaluation methods can be incorporated, particularly by combining criteria in multi-objective fitness functions.

Experiments highlighted the contribution of each criterion, as represented by the respective fitness functions. In the experiments, fitness function $fitness_2$, which combines accuracy with number of selected features, displayed the worst overall performance. On the other hand, better results with regard to both tree size and number of selected features criteria were obtained when these two criteria were combined with *accuracy* ($fitness_4$). Although this fitness function did not present the best absolute performance with regard to *accuracy*, the trade-offs established are interesting, especially in contexts in which the interpretability of the classification model is important. In addition, the decrease on the observed accuracy rate was small, and is not statistically significant.

Obviously these experiments are limited in the sense that a single classifier implementation was adopted, and that weights have not been extensively experimented. Further experimentations with different classifiers and criteria weights need to be developed, as well as the consideration of other criteria (e.g. cost). Techniques for the detection of features redundancy are being integrated in the genetic algorithm. This work is part of a research that aims at using feature selection techniques targeted at business processes classification. In this domain, factors such as model interpretability, time constraints upon features and their cost and performance should be weighted with model accuracy.

References

[1] M. Dash and H. Liu, "Feature Selection for Classification". Intelligent Data Analysis - An International Journal, Elsevier, Vol. 1, No. 3, pages 131 - 156, 1997.

[2] S. B. Kotsiantis and P. E. Pintelas. On the selection of classifier-specific feature selection algorithms. Lecture Note of Intl. Conf. on Intelligent Knowledge Systems, V.1,N.1, August 2004, pp 153-160.

[3] H. Liu and H. Motoda. Feature Selection for Knowledge Discovery and Data Mining, Kluwer, 1998.

[4] J. Yang, V. Honavar. Feature Subset Selection Using a Genetic Algorithm. IEEE Intelligent Systems 13(2):44-49, 1998.

[5] D. E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. 412p., Massachusets: Addison-Wesley Co, 1989.

[6] Z. Sun, G. Bebis, X. Yuan and S. Louis. Genetic Feature Subset Selection for Gender Classification: A Comparison Study. WACV, 165-170, 2002.

[7] K. J. Cherkauer and J.W. Shavlik. "Growing Simpler Decision Trees to Facilitate Knowledge Discovery, " Proceedings of 2nd. Intl. Conf. on Knowledge Discovery and Data Mining, Portland, OR, pp. 315--318 (AAAI Press, San Mateo, CA), 1996.

[8] E. Cantú-Paz. Feature Subset Selection, Class Separability, and Genetic Algorithms. GECCO (1): 959-970, 2004.

[9] C. L. Blake, D. J. Newman, S. Hettich and C. .J. Merz. UCI repository of machine learning databases. 1998.

[10] S. Hettich and S. D. Bay. The UCI KDD archive. 1999.

[11] J. R. Quinlan. C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc, 1993.

FEATURE SELECTION WITH A PERCEPTRON NEURAL NET

Manuel Mejía-Lavalle, Enrique Sucar[1], Gustavo Arroyo
Instituto de Investigaciones Eléctricas, Reforma 113, 62490 Cuernavaca, Morelos, México
[1]INAOE, L.E.Erro 1, 72840 StMa. Tonantzintla, Puebla, México
mlavalle@iie.org.mx, esucar@inaoep.mx, garroyo@iie.org.mx

Abstract. An exploratory research on the utilization of a Perceptron neural network as a method for feature selection is carried out. The basic idea consists on training a Perceptron in the context of supervised learning. The interconnection weights are used as indicators of which attributes could be the most relevant, under the assumption that a interconnection weight close to zero indicates that the associated attribute can be eliminated because it does not contribute to the class separator hyper-plane. The experiments realized with real and synthetic data show that this schema results in a good trade-off among performance (generalization accuracy), efficiency (processing time) and feature reduction.

Keywords: Data mining, Supervised learning, Feature selection, Perceptron neural net.

1 Introduction

Data Mining has been applied successfully for knowledge discovery in databases. Nevertheless, with the fast growth of databases, traditional data mining algorithms become obsolete because they cannot process huge information volumes. Different solutions have been proposed, including feature selection before applying the data mining phase, with the hope to reduce the load over the mining algorithm. Diverse researches have found that feature selection not only reduces processing time, but also often improves the miner's accuracy [1]. Although many feature selection techniques exist, no one performs well in any domain. In this work an exploratory research is presented, in which a trained Perceptron interconnection weights are utilized like a measure of attribute importance. This idea is inspired in part in the Principal Component Analysis technique (PCA) and in the Support Vector Machine (SVM) variant for feature selection (SVM-FS), that have in common to eliminate the attributes whose associated scale factors are close to zero. The advantage of this proposal is that it is very fast and as we will show, competitive with more sophisticated feature selection techniques.

2 Related Work

Although there is many feature selection algorithms reported in the literature, none of them are perfect. Some of them are effective, but very costly in computational time (e.g. wrappers methods), and other are fast, but less effective in the feature selection task (e.g. filter methods). Wrapper methods, although effective in eliminating irrelevant and redundant attributes, are very slow because they apply the mining algorithm many times, changing the number of attributes each time of execution as they follow some search and stop criteria [2]. Filter methods are more efficient; they use some form of *information gain* measurement between individual attributes and the class [3]; however, in general they measure the relevance of each isolated attribute, they cannot detect if redundant attributes exist, or if a combination of two (or more) attributes, apparently irrelevant when analyzed independently, indeed are relevant.

An alternative strategy is using the scale factors produced by, for example, principal component analysis (PCA), support vector machine (SVM) variants for feature selection (SVM-FS) and neural network (NN) paradigms. These approaches have in common to eliminate the attributes whose associated scale factors are close to zero. Within NN, there exists several methods for post learning by pruning interconnection weights, for example Optimal Brain Damage or Optimal Brain Surgeon [4]. The objective of these approaches is to obtain a simplified NN, conserving good or similar classification power of the complete NN, and therefore, there not directly focused on the feature selection task. Brank et.al. [5] conducted a study to observe how scale factor feature selection methods interact with several classification algorithms; however, no information about processing time and feature reduction is presented.

3 Perceptron Feature Selection

We explore the Perceptron as a strategy for relevant feature selection. We propose to use a "soft" or relaxed Perceptron (similar to [6]), in the sense that it can accept some percentage of misclassified instances, where the train-stopping criterion is when no accuracy improvement is obtained. We use the generalization accuracy (*Acc*) and Balanced Error Rate (*BER*) as criteria to evaluate the solution quality. To obtain the Perceptron output *S* we use the equation:

$$S = U\{ \Sigma_i W_i \; E_{ij} \; \} \tag{1}$$

where W_i are the *i* interconnection weights; E_{ij} is the input vector (with *i* elements) that form an instance *j*; and *U* is a step function that outputs 1 if $\Sigma_i W_i E_{ij} > \theta$ and 0 otherwise. θ is the Perceptron threshold. To train the Perceptron we apply the following equations:

$$W_i \; (t+1) = \; W_i \; (t) \; + \; \{ \; \alpha (T - S) E_{ij} \; \} \tag{2}$$

$$\theta \; (t+1) = \; \theta \; (t) \; + \; \{ \; -(T - S) \; \alpha \; \} \tag{3}$$

where *T* is the desired output and α is the learning rate, a user parameter that takes values between 0 and 1. The overall feature selection process we apply is the following:

FS-Perceptron (FS-P) Procedure

Given a numeric dataset with *D* attributes previously normalized [0,1], and *N* randomize instances,

1. Let *AccOld* = 0 (generalization accuracy), WithoutImprove = *ni* (numer of accepted epochs without improve)
2. While *AccNew* better than *AccOld* (*ni* times)
    a. Train a (soft) Perceptron (initial weights in zero)
    b. Test after each epoch, and obtain *AccNew*
    c. If *AccNew* better than *AccOld*: save weights and do *AccOld = AccNew*
3. Drop attributes with small absolute interconnection weights
4. Use the *d* remain attributes ($d < D$) to create a model as the predictor for the J4.8 classifier [7].

Although there exist more sophisticated procedures in the area of neural network pruning [4], we choose this *naïve* idea because of its simplicity (that implies efficiency) and direct application to feature selection (because of the direct relation between each feature and its Perceptron interconnection weight). With the Feature Selection Perceptron (FS-P) we expect:

a) To use less amount of memory, because a Perceptron only requires storing as many interconnection weights as "n" attributes the database has, as opposed to PCA that builds a "n^2" matrix.
b) To reduce the processing time because, as opposed to the SVM-FS that involves solving a quadratic optimization problem, the Perceptron converges fast to an approximate solution.
c) To avoid carrying out an exhaustive exploration (or close to exhaustive), that is to say, without having to evaluate multiple attribute subset combinations, as the wrapper and some filter methods.
d) Implicitly capture the inter-dependences among attributes, as opposed to filter-ranking methods, that evaluate only the importance of one attribute against the class.
e) The Perceptron can be used as a classifier too, with the possible advantage to improve accuracy because this link between the *feature selector - classifier* algorithms, that allows a implicit wrapper schema.

Then, the objective of this paper is to validate experimentally these hypotheses.

4 Experiments

We conducted several experiments with 15 synthetic and real datasets to empirically evaluate if FS-P can do better in selecting features than other well-known feature selection algorithms, in terms of accuracy, processing time and feature reduction. We choose synthetic datasets in our experiments because the relevant features of these datasets are known beforehand.

4.1 Experimentation details

In the first phase we used 10 synthetic datasets, each of them with different levels of complexity. To obtain the 10 datasets we use the functions described in [8]. We used also the *corrAL* synthetic dataset [9]. Additionally, we test our method with two real databases. The first one is a database with 24 attributes and 2,770 instances; this database contains information of Mexican electric billing costumers, where we expect to obtain patterns of behavior of illicit customers. The second is the Ionosphere dataset taken form the UCI repository [10] with 34 attributes and 351 instances. Finally, we consider two dataset taken from the NIPS 2003 feature selection challenge[1]. These datasets have very high dimensionality. The *Madelon* database has 500 features and 2,000 instances and *Gisette* dataset has 5,000 features and 6,000 instances.

In order to compare the results obtained with FS-P, we use Weka´s [7] implementation of ReliefF, OneR and ChiSquared feature selection algorithms. These implementations were run using Weka´s default values, except for ReliefF, where we define 5 as the neighborhood number, for a more efficient response time. Additionally, we compared with several of Elvira´s [11] filter-ranking methods. To select the best ranking attributes, we use a threshold defined by the largest gap between two consecutive ranked attributes, according to [9] (e.g., a gap greater than the average gap among all the gaps).

In the case of FS-P, we set the learning rate α to 0.6, the maximum epochs equal to 500, and the number of epochs without accuracy improvement, *ni*, to 15, for all the experiments. All the experiments were executed in a personal computer with a Pentium 4 processor, 1.5 GHz, and 250 Mbytes in RAM.

4.2 Experimental results

The results of applying FS-P to 10 synthetic datasets are shown in Table 1. We can observe that the averaged processing time (column 2) and epochs (column 3) is acceptable. The generalized accuracy obtained for FS-P is not good (column 4) but the resulting averaged accuracy of using the selected features with the J4.8 classifier (with 10 fold cross validation) is good (column 5). In columns 6 and 7 we can see that the features selected by FS-P are equal or near to the perfect attributes (oracle column). In almost all cases, except for datasets 3 and 5; the average number of features selected are similar.

Table 1. FS-P with 10 Synthetic Databases.

Synthetic Database	FS-P Time(secs)	FS-P Epoch	FS-P Acc(%)	FS-P + J4.8 Acc (%)10-fCV	FS-P Attr.Selected	Oracle
1	3	40	47	100	3-7	3
2	2	24	55	100	1-2-3	1-3
3	2	18	61	68	4	3-4
4	2	17	63	84	1-3	1-3-4
5	3	34	65	82	9	1-3-9
6	4	47	66	99	1-2-3	1-2-3
7	6	59	100	98	9-1-2	1-2-9
8	4	39	100	100	1-2-4	1-2-4
9	4	48	100	97	9-1-2-4	1-2-4-9
10	3	37	99	99	4-8-7-1-2	1-2-4-7-8-9
Avg.	**3.3**	**36.3**	**75.6**	**92.7**	(2.7)	(3)

[1] http://www.nipsfsc.ecs.soton.ac.uk/datasets/

Next, we use the selected features obtained by several feature selection methods as input to the decision tree induction algorithm J4.8 included in the Weka. We use 10-fold cross validation in order to obtain the average test accuracy for each feature subset (in all cases, we obtain similar results using *BER* as quality measure criterion). The results are shown in Table 2 (only averages due to paper space limit). The column "Oracle/All" represents a perfect feature selection method (it selects exactly the same features that each dataset function uses to generate the class label and, in this case, is equal to the obtained accuracy if we use all the attributes). From Table 2 we can see that the FS-P averaged accuracy is better than most feature selection methods, except ReliefF.

Table 2. J4.8´s accuraccy (%) using the features selected by each method (10 Synthetic Datasets).

Synthetic Database	Method											
	Oracle/All	**ReliefF**	**FS-Percep**	**ChiSquar**	**Bhattach**	**Mut.Infor**	**Kullback Leibler-1**	**Matusita**	**OneR**	**Kullback Leibler-2**	**Euclidean**	**Shannon**
Avg.	**99.3**	**96.1**	**92.7**	**92.4**	**91.2**	**90.5**	**89.8**	**89.2**	**84.9**	**84.1**	**83.6**	**79.6**

Processing time is shown in Table 3. We observe that, although FS-P is computationally more expensive than ChiSquared and other filter-ranking methods, these algorithms cannot detect good relevant attributes or some attribute inter-dependencies. On the other hand, FS-P was faster than ReliefF and maintained good generalized accuracy. To have a better idea of the FS-P performance, we can compare the results an exhaustive wrapper approach. In this case, if the average time required to obtain a classification tree using J4.8 is 1.1 seconds, and if we multiply this by all the possible attribute combinations, 12.5 days would be required to conclude such a process.

Table 3. Averaged processing time for each method in seconds (10 Synthetic Datasets).

Exhaustive wrapper	ReliefF	OneR	FS-P	ChiSquared and Elvira
1,085,049 (12.5 days)	573 (9.55 mins.)	8	3.3	1

When we test with the corrAL synthetic dataset, FS-P was the only that can remove the redundant attribute (Table 4); results for FCBF and Focus methods were taken from [9]. Because the corrAL is a small dataset, processing time in all cases is near to zero seconds, and thus omitted.

Table 4. Features selected by different methods (corrAL dataset).

Method	Features selected
FS-Perceptron	A0, A1,B0, B1
ReliefF	R, A0, A1, B0, B1
OneR	R, A1, A0, B0, B1
ChiSquared	R, A1, A0, B0, B1
Symmetrical Uncertainty	R, A1, A0, B0, B1

Method	Features selected
$FCFB_{(log)}$	R, A0
$FCFB_{(0)}$	R, A0, A1, B0, B1
CFS	A0,A1,B0,B1,R
Focus	R

With the Electric Billing database, FS-P obtains similar accuracy as Kullback-Leibler-2 (97.29 vs. 97.5%), but with less processing time (3 vs. 6 secs.). Testing over the Ionosphere database, FS-P obtains similar accuracy as ReliefF (92.5 vs. 92.8%), but with less processing time (0.1 vs. 4 secs.) and good feature reduction (5 vs. 6 features).

Finally, we experimented with the Madelon and Gisette NIPS 2003 challenge datasets. In these cases we can not apply Weka or Elvira feature selection tools because they ran out of memory; so, for comparison, we use the results presented by Chen et.al [12]. They apply SVM with a radial basis function kernel as

feature selection method. Table 5 shows results for Madelon and Gisette datasets (N/A means information not available).

Table 5. Accuracies (%) and *BER* using the features selected by each method (Madelon and Gisette).

Database	Method	Features Total (%)	Accuracy (%)	BER	Pre-process. time
Madelon	FS-Perceptr	21 (4. 2%)	58.35	0.4165	48 secs.
	SVM	13 (2. 6%)	N/A	0.4017	N/A
Gisette	FS-Perceptr	64 (1. 3%)	94. 5	0.0549	3. 3 mins.
	SVM	913 (18. 2%)	N/A	0.0210	N/A

From Table 5 we can observe that the obtained *BER* using FS-P is similar when SVM is applied; on the other hand both, accuracy and *BER*, are poor. The reason for this bad result is because Madelon is a dataset with clusters placed on the summits of a five dimensional hypercube, so, in some sense, is a variation of the XOR problem, a non-linear separable classification problem. Thus, FS-P and SVM (still with a kernel function) fail with this database. In the case of Gisette, that contains instances of handwritten digits "4" and "9", we can see that SVM obtains a superior *BER*, but FS-P achieves an acceptable *BER* and accuracy, using fewer attributes (64 vs. 913).

5 Conclusions and Future Work

According to our experiments, FS-P results in a good trade-off among generalization accuracy, processing time and feature reduction. We observed that FS-P´s memory requirements increase linearly, its generalization accuracy and processing time is competitive against other methods, finds some attribute inter-dependencies, and obtains acceptable feature reductions.

On the other hand, we found some FS-P limitations. First, the Perceptron algorithm can only classify linearly separable classes, so this could affect when the database is non-linearly separable. Second, we observed (with additional experiments not included in this paper) that, sometimes, different learning rates (α) conduct to different relevant attributes; in this case it is necessary to realize experiments with different learning rates, to verify if attributes´ ranking remains stable. Also, due to the early stopping criterion (we stop with few epochs, for efficiency) the Perceptron is a bad classifier. In general, we can conclude that FS-P represents a useful addition to the feature selection existing methods.

Future work includes: a) perform experiments with more datasets, b) apply kernel functions to overcome the linear separability limitation, c) try other stopping criteria, searching, and d) use a metric (e.g. *F-score*) to do first attribute elimination, and then apply FS-P, following [12].

References

1. Guyon, I., Elisseeff, A., An intro.to variable and FS, J.of Mach.Learning R., 3, 2003, pp. 1157-1182.
2. Kohavi, R., John, G., Wrappers for feature subset selection, AI Journal, 1997, pp. 273-324.
3. Molina, L., et.al., A., FS algor., a survey and exp. eval, IEEE Int.conf.d.m., Japan, 2002, pp. 306-313.
4. Jutten, C., et.al., Pruning methods: a review, European symp. on ANN, April 1995, pp. 129-140.
5. Brank, J., et.al., Interaction of FS methods and linear class.models. Proc.of the ICML-02, Sydney, AU, 2002.
6. Gallant, S.I.: Perceptron-Based Learning Algorithms, in IEEE Transactions on NN, 1, 1990, pp. 179-191.
7. www. cs.waikato.ac.nz/ml/weka, 2004.
8. Agrawal, R., et.al., DB mining: a performance perspective, IEEE Trans. KDE, Vol. 5, no. 6, 1993, pp. 914-925.
9. Yu, L., Liu, H., Efficient FS via analysis of relevance and redundancy, J.of ML R. 5, 2004, pp. 1205-1224.
10. Newman, D.J. [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, 1998.
11. www. ia.uned.es/~elvira/ , 2004.
12. Chen,Y., Lin,C., Combining SVMs with various feature selection strategies. To appear in the book "Feature extraction, foundations and applications", Guyon, I. (ed), 2005.

A Continuous Variable Response Driven Transformation for Use in Predictive Modeling

Talbot Michael Katz,
TopKatz@msn.com

Abstract: A new transformation of a continuous-valued predictor for a binary-valued target partitions the range of the predictor variable into separate bins, creates a spline with a knot at the endpoint of each bin, and assigns to each point the value of the derivative of the spline function. The bins are chosen to minimize the sum of squared residuals from fitting weighted target values with polynomial or other suitable functions. The binary target values are initially rearranged to create a cumulative density function of the predictor; this cumulative density function is fit with a regression spline after the bins / knots have been chosen by the minimization procedure. The derivative of the regression spline becomes the transformation. This transformation is most useful in cases where the variation of the target is non-monotonic (and non-random) with respect to the predictor.

Keywords: Optimal, Spline, Knots, Transformation, Derivative

Introduction

Univariate spline functions provide a method of curve fitting by splitting the domain into subintervals, fitting a pre-specified component function (usually a low-degree polynomial) within each subinterval, and joining together the subintervals in a continuous fashion at the subinterval endpoints (called knots). Splines have proven very useful in regression modeling, and several software packages have facilities for computing spline regressions. The most common type of splines fit cubic polynomials on each subinterval, because these can be stitched together in a continuously differentiable manner with a minimal amount of variation between sample points.

The main decisions for creating splines are how many subintervals / knots to use, and where to place them. Happily, there is some evidence that location of knots is less important than the number of knots, and the number of knots usually need not be very large [1]. Nevertheless, this paper describes a pseudo-optimal criterion for choosing the number and location of knots, and then builds new features for binary prediction on the selected subintervals. The method first constructs a continuous target variable from the original binary variable, so the method of choosing the number and location of knots could be applied to create a standard spline transformation for a continuous target variable. For a binary target variable, after building the spline transformation for the constructed continuous outcome, take the derivative to produce the final desired transformation.

Amount and Location of Knots for Continuous Target Variable

The method for determining the amount and placement of knots for the spline transformation closely follows the algorithm for determining the optimal number of bins described in [2]. First suppose that both the target variable and predictor variable under consideration are continuous-valued. The idea will be to choose an appropriate sized sample of points, fit the data to the component function (e.g., cubic polynomial) on each sufficiently large subinterval, and pick the subinterval partition with the best overall fit. Then construct the regression spline (using the

same component function) on this partition. I call this pseudo-optimal because, although it employs an optimization technique, namely integer programming, it is not the same as finding the optimal spline transformation (which would appear to require exhaustive enumeration).

Choose a sample of *N* points, sorted by ascending order of the continuous predictor variable under investigation. Let *x[i]* be the value of the predictor variable and *y[i]* be the corresponding value of the target variable for *1 <= i <= N*. Our goal will be to partition the *N* points into disjoint subsets such that each subset contains a contiguous sequence of all points *k* with *i <= k <= j* for some pair *i* and *j*, i.e., sub-segments or subintervals. Let *f[k;i,j]* be the fitted value corresponding to *y[k]* for *i <= k <= j*, and let *s[i,j]* be the sum of squares of the residuals *(y[k] - f[k;i,j])* for *i <= k <= j*. Define the binary optimization variables *v[i,j]* for each pair of points *1 <= i <= j <= N*; *v[i,j] = 1* will mean that the sub-segment determined by *i* and *j* has been chosen, otherwise *v[i,j] = 0*. The objective will be to minimize the function *c[i,j] * v[i,j]*, where *c[i,j] = s[i,j] + C* for some constant *C*. The choice of *C* will be critical. If *C = 0*, the optimization will want to put each point in its own sub-segment, because *s[i,i] = 0* (at least, when *x[i]* is unique, which is likely for continuous variables); if *C* is very large, the single segment containing all *N* points will be preferred.

The *v[i,j]* variables are subject to the following conditions / constraints :
(required) This says that every point must be in exactly one sub-segment. For each point *k*, the sum of *v[i,j]* over all sub-segments containing *k (i <= k <= j)* is equal to 1.
(optional) If there is a lower bound, *LG*, on the number of sub-segments, then the sum of *v[i,j]* over all pairs of points *i* and *j* (including *i = j*) is >= *LG*.
(optional) If there is a hard upper bound, *UG*, on the number of sub-segments, then the sum of *v[i,j]* over all pairs of points *i* and *j* (including *i = j*) is <= *UG*.
(required) If *LP* is the lower bound on the number of points per sub-segment, then eliminate variables *v[i,j]* with *j+1-i < LP*. For a polynomial component function, *LP* must be at least one more than the degree of the polynomial.
(optional) If there is an upper bound, *UP*, on the number of points per sub-segment, then eliminate variables *v[i,j]* with *j+1-i > UP*.

If the sample data is unevenly distributed, it may also be desirable to add constraints to guarantee that the difference between sub-segment endpoint values is bounded below and / or above. Like the bounds on the number of points per sub-segment, bounds on the differences between endpoints serve to eliminate variables.

The choice of the constant *C* creates a soft upper bound of *1 + (s[1,N]/C)* on the number of sub-segments. (The optimization will pick some number of sub-segments no larger than that value.) The "default" value of *C = s[1,N] / (N - 1)* makes the single sub-segment solution and the solution consisting of all individual point sub-segments equally likely.

Transforming a Binary Target Variable into a Continuous Target Variable

As mentioned above, for a continuous target variable, the spline constructed on the subintervals chosen by the method above could be used as a new candidate feature. But fitting a spline directly to a binary outcome may not yield anything useful, since low-degree polynomials do not produce good fits for binary outcomes. In this case we first transform the target variable, then choose the knots by the above algorithm, then fit the spline to the continuous target, and finally

take the derivative of the resulting spline to use as the new candidate feature for the binary target.

Since individual predictor variable values may occur several times in a sample, suppose there are *M* <= *N* distinct values of the predictor variable, let *n[k]* be the number of times the *k*-th value occurs in the sample, and let *r[k]* be the number of positive responses (assuming possible response values of *0* and *1*) at the *k*-th value for *1* <= *k* <= *M*. Let *D* = {sum of *r[k] / n[k]* over all distinct predictor values, *1* <= *k* <= *M* } and let *d[j]* = {sum of *r[k] / n[k]* over all distinct predictor values, *1* <= *k* <= *j* }. Note that if all the predictor values are unique, then *n[k] = 1* for all *k*, *r[k] = 0* or *1* for each *k*, and *d[j]* is just the cumulative sum of responses up to the *j*-th point, in sorted order. Then *d[j] / D* will be the transformed value of the target variable at the *j*-th distinct value of the predictor; each of the *n[j]* points in the sample with the same predictor value will get the same transformed target value, *d[j] / D*, although they did not necessarily all have the same original value of the target. Notice that for subintervals with a high density of response, the transformed target variable will have a high average derivative value with respect to the predictor, and for subintervals with a low density of response, the transformed target variable will have a low average derivative value with respect to the predictor, so the derivative of the spline will provide a continuously varying analog to the behavior of the response.

Optimization Considerations

Because the number of variables and constraints grows with the sample size, the number of points that can be used in a sample is limited by the power of the solver. This method works readily using the SAS ® PROC LP solver with a sample of 100 to 200 points, which should be adequate to pick up the essential behavior of most continuous variables for modeling purposes. From an optimization standpoint, the key feature is that the integer solution is the same as the LP-relaxation.

Over-fitting

The optimization procedure custom tailors the transformation to the sample it is based on. As noted above, if the objective function constant multiplier, *C*, is set equal to 0, the optimization will attempt to make each sample point its own sub-segment. The easiest way to fight this is to do two things. First, set the value of *C* to a reasonable level, such as the default, which was chosen to be “equidistant” from the single-point-groups and entire-range-group solutions. Second, make sure that each sub-segment has sufficient support by choosing a lower bound on the number of points in each sub-segment. It would be hard to feel comfortable with intervals supported by fewer than ten points. Unfortunately, even ten points is rather small, but it’s difficult to guarantee 25 or 30 points, because the overall sample needs to be kept from growing too big for the optimizer to deal with. So, the next level of protection would be to generate several samples, run the optimization procedure on each of them, and determine a solution based on the combination of all the sample runs; one way to do this would be to compute the objective functions for each solution on each of the samples, and choose the solution which has the best sum of objective values for all the samples.

References:

[1] *Regression Modeling Strategies*, Frank E. Harrell, Jr., 2001, Springer-Verlag, New York, ISBN 0-387-95232-2

[2] *An Optimal Binning Transformation for Use in Predictive Modeling*, Talbot Michael Katz, FSDM 2005

Features in Data Processing vs Features in Data Mining

Tsau Young ('T. Y.') Lin
Department of Computer Science San Jose State University
San Jose, CA 95192, USA, tylin@cs.sjsu.edu

Abstract

In traditional data processing (DP), feature or attribute values represent the human perceived properties, characteristics and so forth. In other words, DP requires full background knowledge support (by DP professionals). On the other hand, data mining(DM), as an automated system, cannot carry out these human perceived properties. Each DM algorithm requires different level of background knowledge system.

Keywords*: attributes, feature, data mining, granular, data model*

1 Introduction

In traditional data processing (DP), feature or attribute values represent the human perceived properties, characteristics and so forth. In other words, DP requires full background knowledge system support (by DP professionals). On the other hand, data mining(DM), as an automated system, cannot carry out these human perceived properties. Data mining algorithms can only process those notions that are encoded in the *data* and *implemented/stored background knowledge*. This paper presents some "surprised" observations to the core techniques of data mining [1].

In this paper, a Relational Table is regarded as a knowledge representation of real world entities. Each entity is represented by one tuple.

1. Data processing requires full background knowledge system support (by DP professionals): (1) A one-to-one semantic preserving correspondence of two tables may preserve the meaning of data. (2) Features can only be preserved by semantic preserving feature transformation. (3) Feature represents the human perceived properties, characteristics and so forth.

2. Association mining requires *no* background knowledge support: (1) A one-to-one correspondence of two tables preserves the association(rule)s (frequent itemsets). (2) Features can be preserved by any one-to-one feature transformation. (3) A feature is a partition of entities

3. Clustering requires the metric of ambient space. (1) An isometry(one-to-one correspondence that preserves the metric) preserves the clusters. (2) Features can be preserved by metric preserving feature transformation (3) A feature is a topological partition under the topology induced by the metric of ambient space.

4. Classification requires different levels of support depending on the applications; some are similar to that of association mining, some are that of clustering.

2 Understanding the Data

2.1 A convention - "word" and "symbol"

First we need to precisely define some key terms.

- A *symbol* is a string of ”bit and bytes” that has no real world meaning. For example, the term ”Yellow” (as a color) is intended to represent what human perceives in his/her optical nerve, but such a meaning is not implemented in a computer system. To a computer system, ”Yellow” is merely a character string; the human feelings and/or physical properties of yellow light are not implemented. A *word* is more than a symbol. A symbol is termed a *word*, if the intended real world meaning *does participate* in the formal processing or computing.

2.2 Data Processing vs Data Mining

To understand the nature of a data, we examine how the data is created: In traditional data processing, (1) we select a set of features/attributes, called relational schema. Then (2) a set of entities is (knowledge) represented by a table of words, in terms of the features/attributes. $K_{map} : V \rightarrow K_{word}$; $v \longrightarrow k$ where K_{word} is a table of words (this is actually the usual relational table). Each word, called an attribute value, represents a real world fact; however the real world meaning is not implemented.

In traditional data processing environment,

- DBMS processes/computes these data under *human commands*, and hence *carries* out the human perceived-semantics. We will term such a processing ”*Computing with Words*.” In other words, the meaning of the symbols does influence/participate in the computing process with human supports.

However, when the same relational table is used in Association Mining (AM). The table of words K_{word} is processed as a table K_{symbol} of symbols. In association rule mining K_{word} has been ”forgotten” into K_{symbol}. In summary,

- The data (relational table) in Data Processing is a *table of words*. The interpretations of symbols has a ”complete knowledge systems” (supported by human being).
- The data (relational table) in Association Mining is a *table of symbols*. There is no knowledge system to support the interpretation.
- The data (relational table) in Clustering is a *table of symbols*. However, the interpretation of symbols requires the knowledge of ambient space. In other words, clustering needs a small knowledge system support.

3 Understanding the Features geometrically

In this section we will consider numerical tables; each tuple is a point in Euclidean space/plane.

3.1 Rotations and Expansions

Let us consider a simple kind of feature transformations (coordinate transformation), namely, the rotations and expansion of the X-Y coordinates. Assume we have a table of 5 points in X-Y-plane. For convenience, we use polar coordinate systems; see Table 1. Noted that the first column of each table (2A, 2B and 2C) indicates the expansion/shrinking in longitude. The second column indicates the angle it rotates; they are differ by a fixed value, θ. So it is obvious that there is a respective one-to-one correspondence between columns. So all these θ-rotated tables are isomorphic to the original table; see [3].

- Rotations and expansions are non-trivial feature transformation from the data processing point of view, but
- Rotations and expansions is an "identity" transformation in association mining. In other word, rotations and expansions do not transform the table out of its isomorphic class; see Section 3.3. However,
- Rotations with *no expansions* induces an "identity" transformation in clustering, that is, it maps clusters to clusters. However a "drastic" expansion may change the clusters; see Section 3.2

3.2 Rotations and Expansions in Clustering

It should be very clear, if θ, say 10,000, get very large, the 5 points in Table 1 may shrink into a cluster; in this case, the distances between 5 points are in the order of 0.00001 (less than 0.00002). Note that 0.087 is 5 degree(5*3.14159)/180=0.087) and $(\theta * 3.14159)/180 = \alpha$. Table 1 illustrate the rotations and expansions/shrinking.

S#	Length Length	Direction	Length Length	Direction	Length Length	Direction	Length Length	Direction
S_1	$1/(2.0+0.0)$	0	$1/(2.0+0.087)$	5	$1/(2.0+\alpha)$	$0+\theta$	$\{S_1,S_2,S_3,S_4,S_5\}$	S_1
S_2	$1/(2.0+0.0)$	30	$1/(2.0+0.087)$	35	$1/(2.0+\alpha)$	$30+\theta$	$\{S_1,S_2,S_3,S_4,S_5\}$	S_2
S_3	$1/(2.0+0.0)$	45	$1/(2.0+0.087)$	45	$1/(2.0+\alpha)$	$40+\theta$	$\{S_1,S_2,S_3,S_4,S_5\}$	S_3
S_4	$1/(2.0+0.0)$	60	$1/(2.0+0.087)$	65	$1/(2.0+\alpha)$	$60+\theta$	$\{S_1,S_2,S_3,S_4,S_5\}$	S_4
S_5	$1/(2.0+0.0)$	90	$1/(2.0+0.087)$	95	$1/(2.0+\alpha)$	$90+\theta$	$\{S_1,S_2,S_3,S_4,S_5\}$	S_5
	Polar coordinate Table 2A		Rotates -5 degree Table 2B		Rotates -θ degree Table 2C		same data encoding	

Table 1. Five points in polar coordinate and rotated coordinate; three table are isomorphic, but not isometric. If the rotation angle θ becomes very large the 5 points will squeeze together very closely and hence form a cluster

3.3 Rotations and Expansions in Association Mining

Table 1 also shows that there are isomorphisms among the rotated tables; and this effect is clearly shown in last column; All tables are identical.

4 Understanding the Association Mining

4.1 Semantic Issues and Knowledge Systems

Let us discuss a hypothetical environment. Assume a data processing department maintains two databases: one for the part department (Table **??**) and one for the human resource (Table 2) We stress here that from data processing point of view, the two table are completely different; One table is about hardware, the other is about human being.

However, data mining department (concentrated on association rules) finds that the two table are isomorphic, that is, there is a one-to-one correspondence between the attribute values of the corresponding columns:

1. Column one: $S_n \leftrightarrow P_n, n = 1, 2, \ldots, 9$

2. Column two: Thirty $\leftrightarrow$ 30; Twenty $\leftrightarrow$ 20; Ten $\rightarrow$ 10.

3. Column three: April $\leftrightarrow$ Hammer; Mar $\leftrightarrow$ Screw; February $\leftrightarrow$ Nail; Jan $\leftrightarrow$ Pin;

4. Column four: NY $\leftrightarrow$ Steel; SJ $\leftrightarrow$ Brass; LA $\rightarrow$ Alloy.

and hence there is one-to-one correspondence between two sets of association rules (support ≥ 2):

1. Length one: Ten $\leftrightarrow$ 10, Twenty $\leftrightarrow$ 20, Mar $\leftrightarrow$ Screw, SJ $\leftrightarrow$ Brass, LA $\leftrightarrow$ Alloy

2. Length two:

   (a) (TWENTY, MAR) $\leftrightarrow$(20, Screw),

   (b) (Mar, SJ) $\leftrightarrow$(Screw, Brass),

   (c) (TWENTY, SJ) $\leftrightarrow$ (20, Brass).

Therefore, it only needs to conduct association mining in one table. The second set of association rules can be found by replacing the symbols from the other set of association rules. So as far as the data mining department is concern, there is only one table to do the association mining.

U	K	($S\#$	Business Amount (in m.)	Birth Day	CITY)	($P\#$	Weight	Part Name	Material
u_1	$\longrightarrow$	(S_1	TWENTY	MAR	NY	(P_1	20	SCREW	STEEL
u_2	$\longrightarrow$	(S_2	TEN	MAR	SJ	(P_2	10	SCREW	BRASS
u_3	$\longrightarrow$	(S_3	TEN	FEB	NY	(P_3	10	NAIL	STEEL
u_4	$\longrightarrow$	(S_4	TEN	FEB	LA	(P_4	10	NAIL	ALLOY
u_5	$\longrightarrow$	(S_5	TWENTY	MAR	SJ	(P_5	20	SCREW	BRASS
u_6	$\longrightarrow$	(S_6	TWENTY	MAR	SJ	(P_6	20	SCREW	BRASS
u_7	$\longrightarrow$	(S_7	TWENTY	APR	SJ	(P_7	20	PIN	BRASS
u_8	$\longrightarrow$	(S_8	THIRTY	JAN	LA	(P_8	30	HAMMER	ALLOY
u_9	$\longrightarrow$	(S_9	THIRTY	JAN	LA	(P_9	30	HAMMER	ALLOY

Table 2. A RelationalTable K and K'

However, in the data analysis department, it has different opinion. Though the two relations, Tables 2, are isomorphic, but their meaning (human interpretations) are completely different. They have very non-isomorphic semantics:

1. (TWENTY, SJ) means the business amount at San Jose is likely 20 millions. However, its isomorphic association (20, Brass) has no meaning at all; weight 20 has no meaning to Brass.

2. (SCREW, BRASS) means the screw is most likely made from Brass. However, its isomorphic association (Mar, SJ) has no real world meaning at all; the association means the supplier was born in March and work at SJ.

- So we are forced to conclude that association rules are not very "meaningful" unless there is a knowledge systems to support the semantics. Hence we need semantic oriented association mining

4.2 Data Encoding for Association Mining

The following discussions are adopted from ([2], pp 702). Let us consider the bitmap indexes for K (see Table 2) the attribute WEIGHT would have 9 bit-vectors. The first, for value 10, is 011100000 because the second, third, fourth tuple have F=10. For value 20 and 30, they are 100011100 and 000000011 respectively. A bitmap index for other attributes can be proceeded similarly.

Next, we note that a bit vector can be interpreted as a subset of V, called an elementary granule. For example, the bit vector, 100011100, of F= 20 represents the subset $\{e_1, e_3, e_4, e_5\}$. Let us summarize the discussions in the Table 3.

V	K	($P\#$	Weight	Part Name	Material	$P\#$	Weight	Part Name	Material
v_1	$\longrightarrow$	(P_1	20	SCREW	STEEL	($\{e_1\}$	$\{e_1, e_5, e_6, e_7\}$	$\{e_1, e_2, e_5, e_6\}$	$\{e_1, e_3\}$
v_2	$\longrightarrow$	(P_2	10	SCREW	BRASS	($\{e_2\}$	$\{e_2, e_3, e_4\}$	$\{e_1, e_2, e_5, e_6\}$	$\{e_2, e_5, e_6, e_7\}$
	$\longrightarrow$	...	...	...	...	...			
v_7	$\longrightarrow$	(P_7	20	PIN	BRASS	($\{e_7\}$	$\{e_1, e_5, e_6, e_7\}$	$\{e_7\}$	$\{e_2, e_5, e_6, e_7\}$
v_8	$\longrightarrow$	(P_8	30	HAMMER	ALLOY	($\{e_8\}$	$\{e_8, e_9\}$	$\{e_8, e_9\}$	$\{e_4, e_8, e_9\}$
v_9	$\longrightarrow$	(P_9	30	HAMMER	ALLOY	($\{e_9\}$	$\{e_8, e_9\}$	$\{e_8, e_9\}$	$\{e_4, e_8, e_9\}$

Table 3. A Relational Table K'and its Granular Table

Table 3 is the so called vertical representation.

- Note that granules of a column are mutually disjoints and forms a partition of V. So a feature is a partition (= equivalence relation) and an "attribute value" is an equivalence class.

It should be clear that the bitmap table and granular table are isomorphic to *the original table*, hence as far as association mining is concerned

- It is adequate to conduct association mining in granular table. (some data miners called this vertical representation)

5 Conclusions

This paper concludes that a "correct" approach to data mining is knowledge based data mining. In other words, data mining systems need respective knowledge systems to support the semantic of data; the knowledge system defines the meaning of a feature or an attribute.

References

[1] Margaret H. Dunham, Data Mining Introduction and Advanced Topics Prentice Hall, 2003, ISBN 0-13-088892-3

[2] H Gracia-Molina, J. Ullman. & J. Windin, J, Database Systems The Complete Book, Prentice Hall, 2002.

[3] T. Y. Lin "Attribute (Feature) Completion– The Theory of Attributes from Data Mining Prospect," in: the Proceedings of International Conference on Data Mining, Maebashi, Japan, Dec 9-12, 2002, pp.282-289.