Feature Selection and Partial Least Squares Based Dimension Reduction for Tumor Classification

Hua-Long Bu, Guo-Zheng Li, Xue-Qiang Zeng
School of Computer Engineering & Science, Shanghai University, Shanghai 200072, China
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China

Jack Y. Yang
Harvard Medical School, Harvard University, Cambridge, Massachusetts 02140 USA

Mary Qu Yang
National Human Genome Research Institute National Institutes of Health (NIH) U.S. Department of Health and Human Services Bethesda, MD 20852 USA

Abstract---**Partial Least Squares (PLS) is one of the widely used dimension reduction methods for analysis of gene expression microarray data, it represents the data in a low dimensional space through linear transformation, the size of the reduced space by PLS is critical to generalization performance of classifiers. The previous works always determined the top fixed number of components or the top several components by cross-validation. Here we demonstrate the usage of feature selection for PLS based dimension reduction. As a case study, PLS is combined with two feature selection methods (Genetic Algorithm and Sequential Backward Floating Selection) to get more robust and efficient dimensional space, and then the constructed data from the selected components is used as input for the Support Vector Machine (SVM) classifier. We use the method for tumor classification on gene microarray data, experimental results illustrate that our proposed framework is effective both to reduce classification error rates and get compact dimensional space.**

Keywords-Feature Selection; Partial Least Squares; Microarray analysis

I INTRODUCTION

DNA microarray experiments are used to collect information from tissue and cell samples regarding gene expression differences for tumor diagnosis [1-3]. The wealth of this kind of data in different stages of cell cycles helps to explore gene interactions and to discover gene functions. Moreover, obtaining genome-wide expression data from tumor tissues gives insight into the gene expression variation of various tumor types, thus providing clues for tumor classification of individual samples. The output of microarray experiment is summarized as an $N \times P$ data matrix, where N is the number of tissue or cell samples; P is the number of genes. Here P is always much larger than N, which will hurts the generalization performance of most classification methods. To overcome this problem, we can construct K new components summarizing the original data as well as possible, with $K \ll P$, which is known as dimension reduction or feature extraction methods [4].

Dimension reduction projects the whole data into a low dimensional space and constructs the new dimensions (components) by analyzing the statistical relationship hidden in the dataset. Building a classification system under this framework mainly involves two steps, 1) extracting a number of features through linear or nonlinear transformation, and 2) training a classifier using the extracted features.

Choosing an appropriate set of features is critical when designing gene classification systems under the framework of supervised learning. Usually, a large number of features are extracted to represent the original data. As we known, the extracted features also contain noise or irrelevant message. Therefore, without employing feature selection, some of them would be either redundant or even irrelevant to the classification task [5].

Ideally, we only want to use the features, which have high separability power while ignoring or paying less attention to the rest. A compact yet salient feature set can simplify both the pattern representation and the classifiers consequently; the resulting classifier will be more efficient. But in most practical cases, relevant features are not known beforehand. Finding out which features to be used in a classification task is referred to as feature selection. There has been a great deal of work in machine learning and related areas to address this issue [6-8], but little attention has been paid to the emerging applications with Partial Least Squares at all.

1-4244-1509-8/07/$25.00 ©2007 IEEE

Partial Least Squares (PLS) is one of the widely used dimension reduction methods for analysis of gene expression microarray data [9-10], it represents the data in a low dimensional space through linear transformation, the size of the reduced space by PLS is critical to the generalization performance of classifiers, especially the initial several components of PLS contain more information than the others, but it is hard to decide how much tail components are trivial for discrimination. Some authors proposed to fixed the number of components from three to five [11]; some proposed to determine the size of the space by classification performance of cross-validation [12]. However each one has its own weakness. Fixing at an arbitrary dimensional size is not applicable to all data sets, and the cross-validation method is often obstructed by its high computation. An efficient and effective model selection method for PLS is demanded.

We propose and demonstrate the importance of feature selection after PLS's transformation in the tumor classification problems. As a case study, PLS is combined with the feature selection methods (Genetic Algorithm and Sequential Backward Floating Selection) to get more robust and efficient dimensional space, and then the constructed data from the original data is used with Support Vector Machine (SVM) for classification. We use the method for tumor classification on gene microarray data, we try to study whether feature selection selects proper components for PLS dimension reduction and whether only the top components are nontrivial for classification.

The rest of this paper is organized as below: In Section II, we will show our proposed dimension reduction framework as well as the details of each component. In Section III, we perform the experiments and give a discussion on the results. Finally, our conclusions will be given in Section IV.

II COMPUTATIONAL METHODS

Partial Least Squares based Dimension Reduction (PLSDR) is a favorite method in gene analysis, but how to determine the number of extracted components for classifiers is a critical problem. In the previous works, the number is fixed as 3 or 5 top ones, or obtained by cross validation. These works assume that only the top several components are important. In fact the components are ranked from a statistical view; it may not the same rank according to their discriminative ability. Therefore, we propose to apply feature selection techniques to select components for classifiers. Fig. 1 illustrates the main steps of the approach employed here. The main difference from the traditional approach is the inclusion of a step that performs feature selection among the features extracted by dimension reduction. From Fig. 1, we can see that dimension reduction consists of two parts, PLSDR and feature selection, here PLSDR is performed by PLS, feature selection is performed by GA and SBFS and classifier is performed by SVM. They are explained in the following subsections.

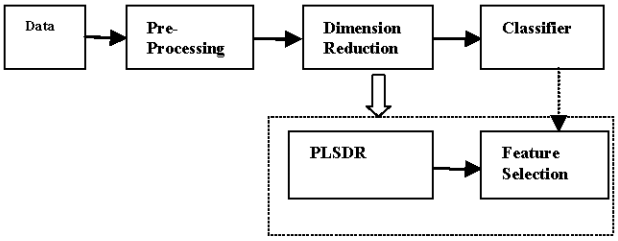

Figure I A framework of feature selection for PLSDR.

A. *Partial Least Squares (PLS)*

Partial Least Squares (PLS) was firstly developed as an algorithm which performing matrix decompositions, and then was introduced as a multivariate regression tool in the context of chemometrics. In recent years, PLS has also been found to be an effective dimension reduction technique for tumor discrimination, which denoted as Partial Least Squares based Dimension Reduction (PLSDR).

The underlying assumption of PLS is that the observed data is generated by a system or process that is driven by a small number of latent (not directly observed or measured) features.

The objective criterion for constructing components in PLS is to sequentially maximize the covariance between the response variable and a linear combination of the predictors. That is, in PLS, the components are constructed to maximize the objective criterion based on the sample covariance between y and Xw. Thus, we find the weight vector w satisfying the following objective criterion,

$$w_k = \arg\max_{w'w=1} \mathrm{cov}^2(Xw, y)$$

Subject to the orthogonal constraint

$$w_i' S w_j = 0 \quad \text{for all } 1 \le i < j .$$

where $S = X'X$.

To derive the components, $[\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_K]$, PLS decomposes X and y to produce a bilinear representation of the data:

$$X = t_1 w_1^T + t_2 w_2^T + \ldots + t_k w_k^T + E$$

$$y = t_1 q_1^T + t_2 q_2^T + \ldots + t_k q_k^T + F$$

where w are vectors of weights for constructing the PLS components. t, q are scalars, and E and F are the residuals. The idea of PLS is to estimate w and q by regression, and a basic algorithm to obtain w, q can be obtained in Ref. [13] Specifically, PLS fits a sequence of bilinear models by least squares, thus given the name partial least squares

The number of components k is the only parameter of PLS which need to be decided by user, here, we argue to use feature selection method to find it. With the increase of k, the explained variances of X and y are expanded, and all the information of original data are preserved when k reaches the rank of X, which is the maximal value of k.

Like PCA, PLS reduces the complexity of microarray data analysis by constructing a small number of gene components, which can be used to replace the large number of original gene expression measures. Moreover, obtained by maximizing the covariance between the components and the response variable, the PLS components are generally more predictive of the response variable than the principal components.

B. Feature Selection

Finding out which features to be used in a classification task is referred to as feature selection. Given a set of d features, the problem is selecting a subset of size m that leads to the smallest classification error. A number of feature selection methods can be found in Refs [14-17]. There are two main components in every feature subset selection system: the search strategy used to pick the feature subsets and the evaluation method used to test their goodness based on some criteria. We summarize both of them below.

a. Search strategies

Search strategies can be classified into one of the following three categories: 1) optimal, 2) heuristic, and 3) randomized. Exhaustive search is the most straightforward approach to optimal feature selection. However, the number of possible subsets grows exponentially, which make the exhaustive search impractical for even moderate size of features. The only optimal feature selection method which avoids the exhaustive search is based on the branch and bound algorithm. This method requires the monotonic property of the criterion function, which most commonly used criterion function do not satisfy.

Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS) are two well-known heuristic feature selection methods. Combining SFS and SBS gives birth to the "plus l take away r" feature selection method [18], which first enlarges the feature subset by adding l using SFS and then deletes r features using SBS. Sequential Forward Floating Search (SFFS) and Sequential Backward Floating Search (SBFS) [19] are generalizations of the "plus l take away r" method. They will be discussed next.

In randomized search, probabilistic steps or sampling process are employed. The relief algorithm [20] is the typical randomized search approaches. Based on their estimated effectiveness for classification, features are assigned weights exceed a user determined threshold are selected to train the classifier. Recently, GA has attracted more attention in the feature selection area, which will be discussed more detailed in Section c.

b. Search evaluation

Each of the evaluation strategies belongs to one of two categories: 1) filter and 2) wrapper. The distinction is made depending on whether feature subset evaluation is performed using the learning algorithm employed in the classifier design (i.e., wrapper) or not (i.e., filter). Filter approaches are computationally more efficient than wrapper approaches since they evaluate the goodness of selected features using criteria that can be tested quickly. This, however, could lead to non-optimal features, especially, when the features dependent on the classifier. As result, classifier performance might be poor. Wrapper methods on the other hand perform evaluation by training the classification error using a validation set. Although this is a slower procedure, the features selected are usually more optimal for the classifier employed.

c. Genetic Algorithm (GA)

Genetic Algorithm (GA) is a class of optimization procedures inspired by the biological mechanisms of reproduction. [21]. GA operate iteratively on a population of structures, each one of which represents a candidate solution to the problem at hand, properly encoded as a string of symbols (e.g., binary). Three basic genetic operators guide this search: selection, crossover, and mutation. The genetic search processes it iterative: evaluating, selecting, and recombining strings in the population during each iteration until reaching some termination condition. The basic algorithm, where P (t) is the population of strings at generation t, is given below:

```
t=0
Initialize P (t)
Evaluate P (t)
While (termination condition is not satisfied) do
Begin
    Select P (t+1) from P (t)
    Recombine P (t+1)
    Evaluate P (t+1)
    t=t+1
End
```

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

d. Sequential Backward Floating Search (SBFS)

SBFS is a well-known heuristic search method; it combines sequential forward search and sequential backward search to the "plus l-take away r" feature selection method and has been proved one of the best heuristic search methods [22]. The main algorithm is given below:

Input:

$X = \{X_i \mid i = 1,...,D\}$ //available measurements//

Output:

$Y_k = \{y_i \mid i = 1,...,k, y_i \in X\}, k = 0,1,...,D$

Initialization:

$Y_0 = \phi; X_0 = X; k = 0$

(in practice one can begin with k=2 by applying SBS twice)

Termination:

Stop when k equals the number of features required

Step 1(Conditional Inclusion)

$x^+ = \arg\max_{y \in Y_k} J(X_k + y)$

if $J(X_k + x^+) > J(X_{k-1})$ **then**

$X_{k-1} = X_k + x^+; k = k - 1; Y_k = Y_k - y;$

go to Step 1

else

go to step 2

end

Step2(Exclusion)

$y^- = \arg\max_{y \in X - Y_k} J(X_k - y)$

$Y_{k+1} = Y_k + y^-; X_{k+1} = X_k - y; k = k + 1$

C. *Support Vector Machine*

Support vector machine (SVM) is a primarily binary classifier that has been shown to be an attractive and more systematic approach to learn linear or non-linear decision boundaries [23]. Their key characteristic is their mathematical tractability and geometric interpretation.

Given a set of points, which belong to either of two classes:

$$(x_1, y_1), (x_2, y_2), ..., (x_l, y_l), x_i \in R^N, y_i \in \{-1, +1\}$$

SVM aims at finding the hyperplane leaving the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyperplane. This is equivalent to performing structural risk minimization to achieve good generalization. Assuming there are l examples from two classes. Finding the optimal hyper-plane implies solving a constrained optimization problem using quadratic programming. The optimization criterion is the width of the margin between the classes. The discriminate hyperplane is defined as:

$$f(x) = \sum_{i=1}^{l} y_i a_i k(x, x_i) + b$$

where $k(x, x_i)$ is a kernel function and the sign of $f(x)$ indicates the membership of x. Constructing the optimal hyperplane is equivalent to find all the non-zero a_i. Any data point x_i corresponding to a non-zero a_i is a support vector of the optimal hyperplane.

Suitable kernels functions can be expressed as a dot product in some space and satisfy the mercer's condition. By using different kernels, SVM implements a variety of learning machines. The Gaussian radial basis kernel is given by

$$k(x, x_i) = \exp(-\frac{\|x - x_i\|^2}{2\sigma^2})$$

The Gaussian kernel is used in this study, since Gaussian kernel is used frequently and proved to be powerful to solve different problems [24].

III EXPERIMENTS

A. *Data sets*

Four real microarray data sets are used in our studies that are briefly described as below.

1). Colon: Alon et al. used Affy metrix oligonucleotide arrays to monitor expressions of over 6,500 human genes with samples of 40 tumor and 22 normal colon tissues. Expression of the 2,000 genes with the highest minimal intensity across the 62 tissues is used in the analysis.

2) Leukemia (LK): The acute leukemia data set was published by Golub et al. The training data set consists of 38 bone marrow samples (27 ALL and 11 AML), over 7,129 probes from 6,817 human genes. Also 34 samples testing data is provided with 20 ALL and 14 AML.

3) Breast Cancer (BC): Van't Veer et al. published the data set. The training data contains 78 patient samples; correspondingly, there are 12 relapse and 7 non-relapse samples in the testing data set. The number of genes is 24,481

4) Prostate: Singh et al. used microarray expression analysis to determine whether global biological differences underlie common pathological features of prostate cancer and to identify genes that might antipate the clinical behavior of Prostate tumors. The data set contains 77 prostate tumor samples and 59 non-tumor prostate samples with 12,600 genes.

B. *Experimental setting*

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

The goal of using GA here is to use fewer features to achieve the same or better performance. Therefore, the fitness evaluation contains two terms:

1) Classification error;

2) The number of features selected.

Between classification error and feature subset size, reducing classification error is our major concern. We use the fitness function shown below:

fitness=10^4*error+0.5*number_of_selected_features,

where error corresponds to the classification error on the validation data set.

The parameters of GA is set by default as in the software of MATLAB [25], and we set the parameters $\sigma = 0.1$, C=10 for SVM, which are tuned by our experiments.

C. *Experimental results*

In order to demonstrate the importance of feature selection of dimension reduction, we have performed four series experiments here:

1) PLS1+SVM, as a baseline method, we use all components of PLS, without any feature selection on the components, for SVM.

2) PLS2+SVM, some initial components of PLS are used. The size of top components is obtained by validating the classifier on the validation data set, as this is a traditional approach.

3) PLS+GA+SVM, beyond the baseline method, we proposed to use GA to select an optimum subset of components, since we consider not all the top components are useful for discrimination but the tail components also contain critical information for discrimination.

4) PLS+SBFS+SVM, the SBFS method is used to select significant components of PLS, similar to the above method.

The average error rates and the corresponding standard deviation values are shown in Table 1. We also show the number of features selected by each method in Table 2. Fig. 2 shows the comparison of distributions of components selected by GA and SBFS on four data sets.

TABLE I. STATITICAL CLASSIFICATION ERROR RATES ON FOUR DATA SETS (%)

Data Set	PLS_1 +SVM	PLS_2 +SVM	PLS+ GA+SVM	PLS+ SBFS+SVM
Colon	25.17(3.1)	23.33(4.2)	18.83(2.2)	22.67(2.7)
LK	10.71(4.3)	9.11 (4.1)	7.57 (3.8)	9.71 (4.2)
BC	36.32 (5.7)	36.21(5.3)	37.16(6.1)	34.84(6.4)
Prostate	18.81 (3.2)	16.67(3.9)	14.89(4.4)	13.63(3.7)
Average	22.75(4.1)	21.33(4.4)	19.61(4.1)	20.21(4.3)

TABLE II. AVERAGE PERCENTAGE OF FEATURES USED BY THE THREE METHODS ON FOUR DATA SETS (%)

Data Set	PLS_2+ SVM	PLS+GA +SVM	PLS+SBFS +SVM
Colon	72.96(2.6)	35.72(3.7)	25.90(2.3)
LK	37.63(2.2)	33.51(4.6)	25.78(4.2)
BC	22.98(4.7)	39.75(6.3)	25.25(5.1)
Prostate	34.79(2.9)	30.17(4.4)	18.80(3.8)
Average	42.09(3.1)	34.78(4.7)	23.93(3.8)

From Table 1, we can find PLS+GA+SVM improves the classification error rates with 1.71% against PLS1+SVM and 3.14% against PLS2+SVM, while PLS+SBFS+SVM improves the result with 1.12 against PLS1+SVM and 2.54 against PLS2+SVM in average on four data sets.

From Table 2, we can find feature selection techniques using GA and SBFS both do help in reducing features about 7.13% and 8.16% than PLS+SVM average on four data sets.

Fig .2 illustrates that the component subsets selected by GA were different from those by SBFS, they do not only contain top components, and they also contain tail components. As we have discussed above, different components seems to encode different kind of information for classification, Fig. 2 shows tail components also encode discriminative information.

D. *Discussion*

One of the difficulties of using PLS is how to select the components; we treat it as a feature selection problem. Here we take two kinds of typical feature selection methods to build a PLS+feature_selection+SVM framework for getting a simpler, gender and efficiency classifier. Observing the tables and figure shown in Section 3.3, several interesting comments can be made as follows:

1) The feature subsets selected by the GA approach improve classification performance. For all different data sets, GA makes a better performance than SBFS. We consider the reason is that SBFS makes local decision, while GA is a kind of random strategy.

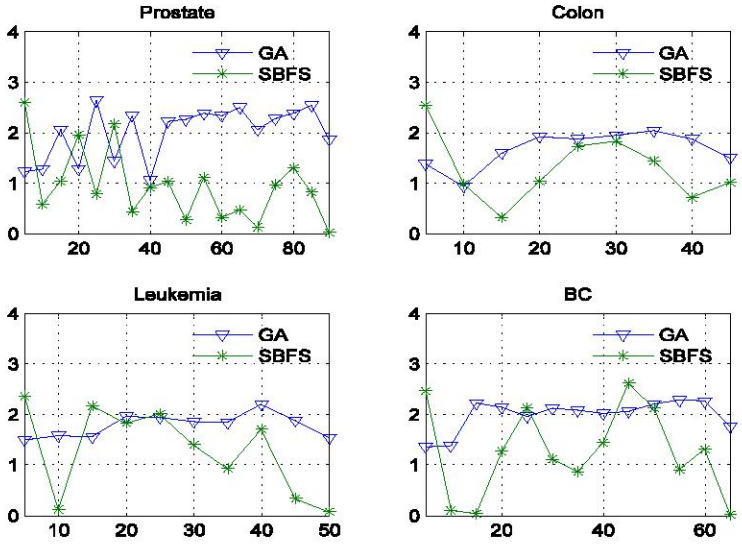

Figure II Comparison of distributions of eigenvectors selected by GA and SBFS on four data sets. (X-axis corresponds to the Eigenvectors, ordered by their eigenvalues and has been divided into bins of size 5. The y-axis corresponds to the average number of times an eigenvector within some bin was selected by GA/SBFS)

2) The GA/SBFS solutions are quite compact: The final feature subsets found by GA/SBFS are compact than traditional methods; the significant reduction in the number of components can also speeds up classification substantially.

3) Different features encode different information. SBFS and GA uses fewer features than traditional method and not all the top components are used, but SBFS and GA both can obtain better results than traditional methods, which shows that not all the top components are useful for classification, the tail component also contain discriminative information and top components maybe irrelevant or redundant.

IV Conclusion

We have investigated a systematic feature reduction framework by combing feature extraction with feature selection. To evaluate the proposed framework, we used four typical data sets. In each case, we used PLS for feature extraction, GA and SBFS as feature selection, and SVM for classification. Our experimental results illustrate that the proposed method improves the performance on the gene expression microarray data in the accuracy. Further study of our experiment indicates that not all the top of PLS's components are useful for classification, the tail component also contain discriminative information. Therefore, it is necessary to combine feature selection with feature extraction and replace the traditional feature extraction step as a new preprocessing step for analyzing high dimensional problems.

Acknowledgment

This work was supported in part by the Nature Science Foundation of China under grant no.20503015, Nature Science Project of Shanghai Municipal Education Committee under grant no.05AZ67 and open funding by Institute of Systems Biology of Shanghai University, China.

References

[1]. T. Golub, D. Slonim, P. Tamayo, et al. Molecular classification of cancer: Class discovery and class prediction by gene expression, Bioinformatics & Computational Biology, 286 (1999) , 531–537.

[2]. U. Alon, et.al, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, in Proceedings of the National Academy of Sciences of the United States of America, (1999), 6745–6750.

[3]. S. Dudoit, J. Fridlyand, and T.Speed, Comparison of discrimination methods for the classification of tumors using gene expression data, Journal of the American Statistical Association, 97 (457) (2002) . 77-87.

[4]. D.D. Lewis, Feature Selection and Feature Extraction for Text Categorization, Proc. Workshop Speech and Natural Language, (1992) 212-217

[5]. S. Watanabe, Pattern Recognition: Human and Mechanical, Wiley, New York, 1985.

[6]. Sun Zehang, Bebis George, Ronald Miller, Object detection using feature subset selection, Pattern Recognition 37, (2004) 2165-2176

[7]. P. Zhang, T. D. Bui, C. Y. Suen, Hybrid Feature Extraction and Feature Selection for Improving Recognition Accuracy of

Handwritten Numerals, Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition, 1520-5263/05

[8]. Jun Yan, et.al, Effective and Efficient Dimensionality Reduction for Large-Scale and Streaming Data, Preprocessing, IEEE transactions on knowledge and data engineering, 18(2006)

[9]. A.-L. Boulesteix, "Pls dimension reduction for classification of microarray data," Statistical Applications in Genetics and Molecular Biology, 3(1) (2004).

[10]. Danh V.Nguyen and David M.Rocke,Tumor classification by partial least squares using microarray gene expression data,Bioinformatics,18(2002)

[11]. Nguyen, D. V. and Rocke, D. M. Multi-class cancer classification via partial least squares with gene expression profiles. Bioinformatics, 18(9), (2002) 1216–1226.

[12]. Dai, J. J., Lieu, L., and Rocke, D.. Dimension reduction for classification with gene expression data. Statistical Applications in Genetics and Molecular Biology, 5(1), (2006)

[13]. H. Wold, Quantitative Sociology: International Perspectives on Mathematicaland Statistical Model Building, 1975, ch. Path models with latent variables: the NIPALS approach, p. 307.

[14]. M. Dash, H. Liu, Feature selection for classi1cation, Intelligent Data Anal. 1 (3) (1997) 131–156.

[15]. A. Blum, P. Langley, Selection of relevant features and examples in machine learning, AIJ. 97 (1997) 245–271.

[16]. R. Kohavi and G. John, Wrappers for Feature Subset Selection, Artificial Intelligence, 97(1-2), (1997) 273-324

[17]. Y. Yang and J.O. Pedersen, A Comparative Study on Feature Selection in Text Categorization, ICML, (1997) 412-420

[18]. S. Stearns, On selecting features for pattern classi1ers, The Third International Conference of Pattern Recognition, (1976) 71–75.

[19]. P. Pudil, J. Novovicova, J. Kittler, Floating search methods in feature selection, Pattern Recognition Letter 15 (1994)

[20]. K. Kira, L. Rendell, A practical approach to feature selection, ICML, (1992) 249–256.

[21]. D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Reading, MA: Addison and Wesley, 1989.

[22]. A. Jain, D. Zongker, Feature selection: Evaluation, application, and small sample performance, IEEE TPAMI 19 (1997) 153–158.

[23]. V. Vapnik, The Nature of Statistical Learning Theory, Springer, Berlin, 1995.

[24]. Chen Nian-Yi, Lu Wen-Cong, Yang Jie, Li Guo-Zheng, Support vector machines in Chemistry, Singapore, World Scientific Publishing Company, September 30, 2004

[25]. MATLAB R2006a, Version 7.2.0.232, produced by The Mathworks, Inc