The First International Symposium on Optimization and Systems Biology (OSB'07)
Beijing, China, August 8–10, 2007
, pp. 232–241

Reducing Error of Tumor Classification by Using Dimension Reduction with Feature Selection

Hua-Long Bu[1,2,*] Guo-Zheng Li[1,2,†]
Xue-Qiang Zeng[1,2,‡]

[1] State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing 210093, China
[2] School of Computer Engineering and Science, Shanghai University
Shanghai 200072, China

Abstract Dimension reduction is an important issue for analysis of gene expression microarray data, of which principle component analysis (PCA) is one of the frequently used methods, and in the previous works, the top several principle components are selected for modeling according to the descending order of eigenvalues. While in this paper, we argue that not all the first features are useful, but features should be selected form all the components by feature selection methods. We demonstrate a framework for selecting good feature subsets from all the principle components, leading to reduced classifier error rates on the gene expression microarray data. As a case study, we have considered PCA for dimension reduction, genetic algorithms and the floating backward search method for feature selection, and support vector machines for classification. Experimental results illustrate that our proposed framework is effective to reduce classification error rates.

Keywords Feature selection; dimension reduction; support vector machines.

1 Introduction

DNA microarray experiments are used to collect information from tissue and cell samples regarding gene expression differences for tumor diagnosis [1-3]. The output of microarray experiment is summarized as an N*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 either select a small subset of interesting genes (gene selection) or construct K new components summarizing the original data as well as possible, with K<<P (dimension reduction, feature extraction).

Gene selection has been studied extensively in the last few years. The most commonly used procedures of gene selection are based on a score which is calculated for all genes individually and genes with the best scores are selected [4, 5].

*Email: fellowbhl@shu.edu.cn
†Email: gzli@shu.edu.cn
‡Email: stamina_zeng@shu.edu.cn

Gene selection procedures output a list of relevant genes which can be experimentally analyzed by biologists. These methods are often denoted as univariate gene selection, whose advantages are its simplicity and interpretalility. However, much information contained in the dataset is lost when genes are selected solely according to their individual capacity to separate the samples, since interactions and correlations between genes are omitted, as are of great interest in system biology recently.

Dimension reduction is an alternative to gene selection to overcome the problem of curse of dimensionality [6].Unlike gene selection, dimension reduction projects the whole data into a low dimensional space and constructs the new dimensions (components) by analyzing the statistical relationship hidden in the dataset. Building a tumor classification system under this framework involves two main steps (1) extracting a number of features, and (2) training a classifier using the extracted features. Principle components analysis (PCA) is one of the frequently used methods for dimension reduction of microarray data [7-8].

Choosing an appropriate set of features is critical when designing gene classification systems under the framework of supervised learning. Often, a large number of features are extracted to represent the original data. Without employing feature selection strategy, however, many of them could be either redundant or even irrelevant. Ideally, we would like to use the features which have high separability power while ignore or pay less attention to the rest. An appropriate feature set can simplify both the pattern representation and the classifiers consequently; the resulting classifier will be more efficient. In most practical cases, relevant features are not known a priori. Finding out what features to use in a classification task is referred to as feature selection. In the previous works, features of eigenvectors are chosen according to their corresponding eigenvalues, eigenvectors corresponding to the top several largest eigenvalues are selected for modeling, but eigenvectors of the tail component also contain information and may be more important for classifiers [9, 10].

In this paper, we propose using genetic algorithms to search the space of eigenvectors with the goal of selecting a subset of eigenvectors encoding important information. This is in contrast to the typical strategy of picking a percentage of the top eigenvectors to represent the original data. This approach has the advantage of simple, general, and powerful. A similar work can be found in Ref. [11], which use PCA and GA for object recognition

The rest of the paper is organized as follows: Section 2 introduces our proposed framework and the detailed techniques. In Section 3, numerical experiments on three real gene expression microarray data sets are performed and results are discussed. In Section 4, conclusions are given.

2 Computational Methods

As discussed above, there are three main steps in building a tumor classification system using supervised learning. 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 principle components extracted by

feature extraction. From Fig. 1, we can see that dimension reduction consists of two parts, feature extraction and feature selection, here feature extraction is performed by principle components analysis, and feature selection is performed by genetic algorithm and the backward floating search method. They are explained in the following subsection.

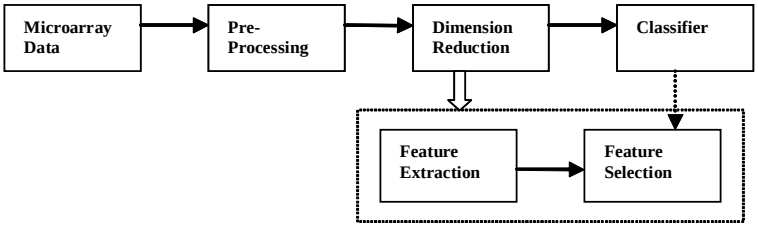

Figure 1: A framework of dimension reduction for tumor classification.

2.1 Feature Extraction Using PCA

PCA is a well known method of dimension reduction [12]. The basic idea of PCA is to reduce the dimensionality of a data set, while retaining the variation present in the original predictor features as much as possible. We summarize the main idea below:

Firstly the average sample ψ is computed: $\psi = \frac{1}{N}\sum_{i=1}^{N}\Gamma_i$, where N is the number of samples in the training set, Γ_i represent the ith sample.

Next, the difference ϕ of each sample from the average sample is computed: $\phi_i = \Gamma_i - \Psi$. Then the covariance matrix is estimated by $C = \frac{1}{N}\sum_{i=1}^{N}\phi_i\phi_i^T = AA^T$, where A= $[\phi_1, \phi_2, ..., \phi_N]$. The eigenspace can then be defined by computing the eigenvectors $\mu_i$of C.

Usually, we only need to keep a smaller number of eigenvectors $R_k$corresponding to the largest eigenvalues.

2.2 Feature Selection Using Genetic Algorithm

Genetic algorithm (GA) is a class of optimization procedures inspired by the biological mechanisms of reproduction. [13-14]. 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 (generation) 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.

2.3 Support Vector Machines

Support vector machines (SVMs) are primarily two-class classifiers that have been shown to be an attractive and more systematic approach to learn linear or non-linear decision boundaries [15]. Their key characteristic is their mathematical tractability and geometric interpretation.

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

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

SVMs aim 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-zeroa_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, SVMs implement 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 [16].

3 Numerical Experiments

3.1 Data sets

Six real microarray data sets are used in our studies which are briefly described as below.

1)Central Nervous System (CNS): Pomeroy et al. developed a classification system based on DNA microarray gene expression data derived from patient samples of Embryonal tumors of the central nervous system. The data set used in our study contains 60 patient samples with 7129 genes, 21 are survivors and 39 are failures.

2).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.

3) Leukemai: The acute leukemia data set was published by Golub et al. Training data set consists of 38 bone marrow samples (27 ALL and 11 AML), over 7129 probes from 6817 human genes. Also 34 samples testing data is provided with 20 ALL and 14 AML.

4) Breast Cancer (BC): The data set was published by Van't Veer et al. 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 24481

5)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.

6) Lung Cancer: The data set was published by Gordon et al. Classification between malignant pleural mesothelioma (MPM) and adenocarcinoma (ADCA) of the lung. There are 181 tissue samples (31 MPM and 150 ADCA). The training set contains 32 of them, 16 MPM and 16 ADCA. The rest 149 samples are used for testing. Each sample is described by 12533 genes

3.2 Experimental setting

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

The goal of using GA here is to use less features to achieve the same or better performance. Therefore, the fitness evaluation contains two terms: (1) classification error and (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 SVM and GA are set in default as in the software of The Spider and MATLAB [17-18].

3.3 Experimental results

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

1) SVM, This is a baseline method, SVM has achieved satisfactory results, and here it is used without any feature reduction on the data sets.

2) PCA+SVM, PCA is a feature extraction method, it is used for dimension reduction without feature selection and the classification of SVM is used. The size of top eigenvectors of PCA is obtained by validating the classifier on the validation data set, as is a traditional way.

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

4) PCA+BFS+SVM, the backward floating search (BFS) method is used to selection significant eigenvectors of PCA, because BFS 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 [19] and has been proved one of the best heuristic search methods [20].

The average error rates and their 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 eigenvectors selected by GA and FBS on six data sets.

Table 1: Average classification error rates on six data sets

Data sets	SVM	PCA+SVM	PCA+BFS+SVM	PCA+GA+SVM
CNS	43.67(7.07)	42.46(4.45)	39.83(5.50)	40.69(6.16)
Colon	31.75(6.91)	29.83(6.22)	24.40(4.63)	23.61(3.42)
Leukemia	8.13(4.87)	6.83(5.34)	6.43(5.32)	4.17(2.10)
BC	36.75(7.05)	35.56(5.27)	30.67(6.27)	21.19(4.39)
Prostate	11.61(4.23)	17.04(5.28)	9.24(5.33)	7.65(2.61)
Lung	8.50(2.55)	14.00(4.79)	5.21(4.52)	1.67(1.06)
Average	23.40(5.45)	24.2(5.22)	19.26(5.26)	16.49(3.29)

From Table 1**,** we can find feature extraction using PCA get about the same result with SVM on the average value of six data sets, while feature selection further improves SVM by 4.41 percent in the case of BFS, and 6.94 percent in the case of GA than PCA+SVM.

Table 2: Average percentage of features used by the three methods on six data sets

Data sets	PCA+SVM	PCA+BFS+SVM	PCA+GA+SVM
CNS	65.54(4.73)	46.92(4.25)	40.31(5.66)
Colon	79.71(7.89)	28.21(5.71)	32.79(5.43)
Leukemai	76.00(8.22)	26.37(7.11)	29.81(4.82)
BC	92.72(2.12)	79.55(3.35)	45.95(7.29)
Prostate	87.93(3.48)	14.63(7.19)	47.47(7.39)
Lung	98.00(9.77)	76.42(4.23)	37.73(2.63)
Average	83.31(6.03)	45.35(5.30)	32.34(5.53)

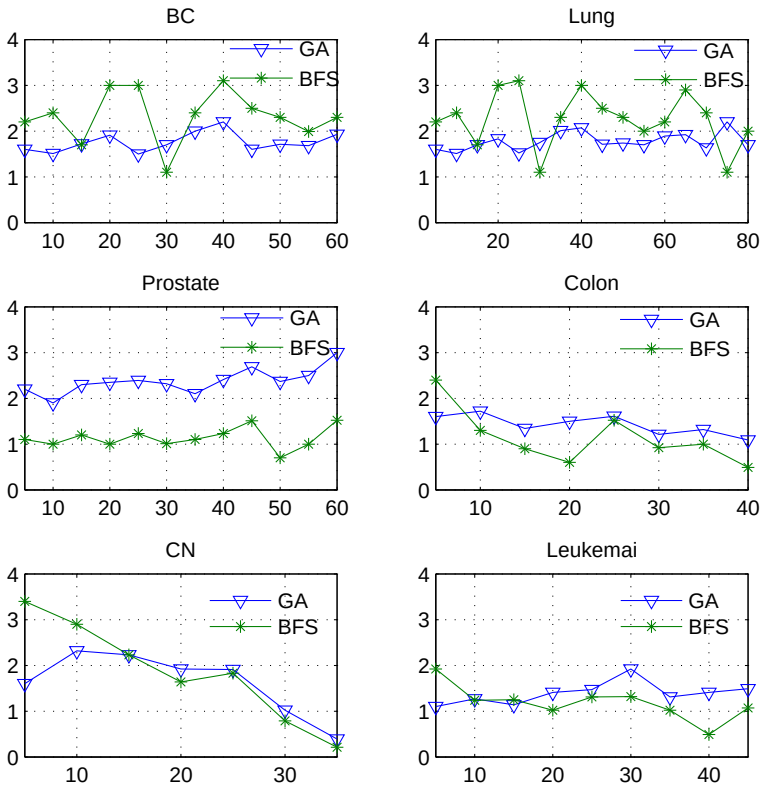

Figure 2: Comparison of distributions of eigenvectors selected by GA and FBS on six data sets. (X-axis corresponds to the Eigenvectors, ordered by their eigenvalues and has been divided into bins of size 5. The y-axis corresponds to the average number of times an eigenvector within some bin was selected by GA/BFS)

From Table 2, we can find feature selection of GA and BFS do great help in reducing features, they only use about one third eigenvectors while the optimized PCA uses more than two third top eigenvectors.

Fig.2 illustrates that the eigenvector subsets selected by GA were different from those by BFS, they do not only contain top eigenvectors, and they also contain tail eigenvectors. As we have discussed in Section 2.2, different eigenvectors seems to encode different kind of information, Fig. 2 shows tail eigenvectors encode discriminative information.

3.4 Discussions

The difficulties of building a classifier for gene expression microarray data are dimension reduction. Here we use the PCA+GA+SVM framework to get a simpler, gender and efficiency classifier. Observing the tables shown in Section 3.3, several interesting comments can be made as below:

1) The feature subsets selected by the GA approach improve classification performance, all for the different data sets. GA can make a better performance than BFS: Since BFS makes local decision, while GA is a kind of random strategy, it's not surprised that GA improve the analysis performance.

2) The GA solutions are quite compact: The final feature subsets found by GA are very compact; the significant reduction in the number of eigenvectors speeds up classification substantially.

3) Features encoding irrelevant or redundant information have not been favored by the GA: comparing the BFS and PCA solely, we can find GA both reduces the average error rate and the number of features selected. This means many of the irrelevant or redundant information have been discarded and it improves the classifier performance.

4) BFS and GA uses only one third eigenvectors, and the optimized PCA uses two thirds eigenvectors, but BFS and GA obtain better results than PCA, this shows that not all the top eigenvectors are useful for classification, the tail eigenvector also contain discriminative information.

4 Conclusion

We have investigated a systematic feature reduction framework by combing feature extraction with feature selection. To evaluate the proposed framework, we used sixe typical data sets .In each case, we used PCA for feature extraction, GA and BFS 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 eigenvectors of PCA are useful for classification, the tail eigenvector also contain discriminative information. Therefore, it is necessary to combine feature selection with feature extraction for dimension reduction 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] Hedenfalk, et al. Gene expression profiles in hereditary breast cancer. New England Journal of Medicine, 24(2001), 539-548.

[5] M.Dettling, and P. Buhlmann, Boosting for tumor classification with gene expression data. Bioinformatics, 19(2003), 1061-1069.

[6] J. Jian. Dai, Lieu Linh, Dimension Reduction for Classification with Gene Expression Microarray Data Statistical Applications in Genetics and Molecular Biology 5（1）（2006）

[7] D.Ghosh. Singular value decomposition regression modeling for classification of tumors from microarray experiments. Proceedings of the Pacific Symposium on Biocomputing, 2002，11462-11467.

[8] Radka Stoyanova, Stoyanova et.al. Normalization of single-channel DNA array data by principal component analysis, Bioinformatics 20 (2004) 1772-1784

[9] Ho Pun-Mo, Wong Tien-Tsin, and Leung Chi-Sing, Compressing the Illumination-Adjustable Images with Principal Component Analysis, IEEE transactions on circuits and systems for video technology, 15 (3) (2005).

[10] Krishna K. Anaparthi, Balarko Chaudhuri, Bikash C. Pal, Coherency Identification in Power Systems Through Principal Component Analysis, IEEE transactions on power systems. 20 (3) (2005）

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

[12] Jolliffe, I.T. (1986). Principal Component Analysis. Springer, New York.

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

[14] M. Srinivas, L. Patnaik, Genetic algorithms: a survey, IEEE Computer. 27 (6) (1994) 17–26.

[15] Nello Cristianini & John Shawe-Taylor, An introduction to support vector machines and other kernel-based methods. Cambridge University Press, 2000

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

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

[18] The Spider, Version 1.71 – of the spider for Matlab, creators: Jason Weston, André Elisseeff, et al.

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

[20] A. Jain, D. Zongker, Feature selection: Evaluation, application, and small sample performance, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1997) 153–158.