Noise-based Feature Perturbation as a Selection Method for Microarray Data

Li Chen¹, Dmitry B. Goldgof¹, Lawrence O. Hall¹, and Steven A. Eschrich²

¹ Department of Computer Science and Engineering
University of South Florida

`{lchen2,goldgof,hall}@csee.usf.edu`

² Department of Interdisciplinary Oncology
H. Lee Moffitt Cancer Center & Research Institute
University of South Florida
Tampa, FL 33620, USA
`Eschris@ Moffitt.usf.edu`

Abstract. DNA microarrays can monitor the expression levels of thousands of genes simultaneously, providing the opportunity for the identification of genes that are differentially expressed across different conditions. Microarray datasets are generally limited to a small number of samples with a large number of gene expressions, therefore feature selection becomes a very important aspect of the microarray classification problem. In this paper, a new feature selection method, feature perturbation by adding noise, is proposed to improve the performance of classification. The experimental results on a benchmark colon cancer dataset indicate that the proposed method can result in more accurate class predictions using a smaller set of features when compared to the SVM-RFE feature selection method.

Key words: feature perturbation, microarray gene expression data, gene selection, classification

1 Introduction

Microarray technology makes it possible to measure thousands of gene expressions simultaneously. One of the major goals of microarray data analysis is the detection of differentially expressed genes across two kinds of tissue samples or samples obtained under two experimental conditions in this high-dimensional gene space. Due to the characteristics of microarray datasets, which usually have a small number of samples and a large number of gene expressions, feature selection methods are quite important to enable microarray classification. A large number of feature selection methods have been proposed to identify subsets of features, thereby reducing the probability of overfitting. In general, classifier-based gene selection can be performed using filter or wrapper approaches. Recent studies have demonstrated that wrapper methods often give satisfactory classification accuracy [1][2] since they evaluate a subset of features based on the

performance of a specific classifier. For example, Guyon et al. introduced a recursive feature elimination (RFE) scheme, in which features were eliminated successively according to their influence on a support vector machine (SVM) based evaluation criterion [3]. Some extended wrapper methods have been proposed based on SVMs [4]. In [5], a wrapper feature selection method was proposed which organized the feature combinations in a tree structure and the learning algorithm was used for evaluation and intelligently searching the tree structure to find the optimal subset of features for classification.

In the context of microarray data analysis, different feature selection methods have their own advantages and disadvantages. For example, the SVM-RFE method in [3] has been shown to be an effective gene selection method, however the classifier used in the feature selection procedure in this method is restricted to an SVM. The wrapper feature selection method [5] does the best search in the feature space for classification but the search procedure is very time consuming. In this paper, we propose a new feature selection method, feature perturbation by adding noise, to improve the performance of classification. This method is more general than the SVM-RFE method since it can be applied to any classifier. By using an adaptive feature elimination strategy, the running time can be dramatically shortened which makes the algorithm practical. Experimental results on the benchmark colon cancer dataset show that the proposed method can achieve higher prediction accuracy with a lower number of features (genes) when compared to the SVM-RFE method.

2 Method

2.1 Feature Perturbation

Feature selection methods are generally applied to all features to eliminate some of the original features and retain a minimum subset of features that yield good classification performance. The feature perturbation method proposed in this paper is a classifier-based, top-down feature eliminating method. The assumption of the feature perturbation method is quite straightforward: irrelevant features will have little influence on classification performance when perturbed by noise. Correspondingly, classification performance should be significantly impacted when important features are perturbed by noise. As a result, irrelevant features with little effect on classification under noise are removed and the subset of important features for classification will be retained.

Fig. 1 shows a flowchart of the feature perturbation method. The feature perturbation method starts with a set of training samples X with all features S as the surviving features. The method is an iterative procedure and the features are recursively eliminated. Each iteration consists of three stages. The first stage is training the classifier on the training samples X , based on the current surviving features S . Any classifier can be used to build the classification model. The second stage is called feature ranking, which computes the ranking criteria for the S features. The change of classification performance on the training samples before and after adding noise to each feature is defined as the ranking criterion

for this feature. Feature elimination removes the feature set R , which contains the features with the lowest ranks from S . For computational reasons, it will be more efficient to remove several features at a time rather than one. The whole procedure will be repeated until all features are removed. As a result, a ranked feature list will be generated and one can evaluate the selected feature set on the test samples using the resultant classification model.

Fig. 1. Flow chart of feature perturbation method

From the flowchart in Fig. 1 we can see that the most important aspect of the feature perturbation method is the feature ranking procedure. Assume that the active feature set is S and the training sample set is X . The performance of the classifier built on the current dataset is evaluated by the training accuracy represented by Acc . For each feature i in S , noise will be randomly generated and added to this feature for each of the samples in X , to generate a new dataset X' . Class predictions using the new dataset, where the feature i was perturbed by noise, will give us a prediction accuracy denoted as Acc_i . The ranking criterion for feature i can be calculated as the difference between Acc and Acc_i .

In the feature ranking procedure, noise is added to each feature to measure the importance of the features for classification. Here the problem is how to decide the characteristics of the noise. Intuitively, noise should be dependent on each particular feature if we assume that the features are independent of each other. However, it is not necessary for the noise to follow a certain distribution since the noise is only used for perturbing the features. In the following experiments, we use a uniform distribution to randomly sample noise. For each feature i the noise follows a uniform distribution with the parameters related to this feature, which can be represented by following Eq. 1:

$$Noise_i \sim Uniform(-c * sd_i, c * sd_i). \quad (1)$$

where, sd_i is the standard deviation of feature i across all training samples, and c is a constant factor which determines the noise level. Large amounts of noise were required to get the appropriate perturbation of the features in our experiments. This procedure was repeated 5 times in each iteration to get an average ranking criteria for each feature since the noise is randomly generated in the experiments.

Given the characteristics of microarray datasets which usually have small number of samples and a large number of gene expressions, the feature elimination procedure will be time consuming if features are eliminated one by one. Based on the assumption that there are only a few relatively important features for classification and most of the features are irrelevant, we used an adaptive feature elimination strategy in which the number of features removed is determined by the current number of surviving features. For example, half the features can be removed at a time rather than just one when there are likely to be a large number of irrelevant features. When the number of features is less than some threshold, one can eliminate the features individually to get more accurate results.

2.2 Feature Perturbation Method vs. SVM-RFE Method

An alternative algorithm for feature selection is the SVM-RFE method [3]. This approach eliminates the features successively according to their influence on the training SVM model. Fig. 2 shows the flowchart of the SVM-RFE method. Compared to the flowchart of the feature perturbation method in Fig. 1, we can see similarities except for the first two stages in each iteration. In the first stage of SVM-RFE, a SVM-based training model C is built on the training samples X

and current surviving features S . In the SVM-RFE feature ranking, the weight vector W is computed based on the training model and the ranking criteria for each feature i is determined by the corresponding weight w_i for this feature.

Fig. 2. Flow chart of SVM-RFE method

One distinct advantage of the feature perturbation method is that it is applicable to any classifier for feature selection and evaluation, while SVM-RFE is a method based only on support vector machines. Therefore, the feature perturbation method is more general than SVM-RFE. The computational complexity

of the feature perturbation method in each feature elimination procedure is $O(d^2 * n)$, while SVM-RFE requires $O(d * n)$ evaluation. Here, d is the number of active features and n is the number of training samples. The feature perturbation method is expected to be slower than the SVM-RFE method but we can use the adaptive feature elimination approach to speed up the computation time. The last characteristic of the feature perturbation method is randomness in selecting the features since the noise is randomly generated and added.

3 Experimental Results

3.1 Data and Parameters

We present results on a well-known benchmark microarray dataset in colon cancer which can be obtained from the website <http://microarray.princeton.edu/oncology/affydata/>. The colon cancer data consists of 62 tissue samples including 22 normal and 40 colon cancer tissues. Each sample has 2000 gene expression values. We use tenfold cross validation to evaluate prediction accuracy between the feature perturbation method and SVM-RFE method. Although the feature perturbation method is applicable to any classifier, support vector machines are used in our experiments in order to compare with the SVM-RFE method. The SVM used as the classifier is a modified version of libSVM [6]. Because no substantial difference in performance was observed in our preliminary classification using SVM with different kernels, a linear kernel with parameter $C = 1$ was used in the following experiments to reduce both training time and the probability of overfitting. The sequential minimal optimization (SMO) algorithm was the optimization algorithm used.

Since the samples are unequally distributed amongst both classes, weighted accuracy will yield a better estimate of how well the classifier performs than total accuracy [7]. Weighted accuracy is defined in Eq. 2:

$$Weighted\ Accuracy = \left(\frac{TP}{TP + FN} + \frac{TN}{FP + TN} \right) / 2. \quad (2)$$

where TP, FP, TN, and FN are the number of true positives, false positives, true negatives, and false negatives, respectively in a confusion matrix in the context of prediction with colon cancer, as shown in Table 1.

Table 1. Confusion matrix

	Predicted Cancer	Predicted Normal tissue
Actual Cancer	True Positive (TP)	False Negative (FN)
Actual Normal tissue	False Positive (FP)	True Negative (TN)

In the following experiments, all the results will be reported using weighted accuracy.

All the programs were implemented using the C language on a personal computer with a 2.8 GHZ CPU and 1 GB of RAM.

3.2 Experiments and Results

We have observed that 1522 of the 2000 features have p -values > 0.05 in the t -test, which validates our assumption that most of the features (genes) are uninformative for classification. Therefore we apply the following adaptive feature elimination strategy to speed up the run time for both the feature perturbation and the SVM-RFE method in the experiments. Starting from 2000 features, half of the features were removed each time when the number of features was greater than 200, after that, features were removed one by one until none remain. As a result, the evaluation will be performed for the feature sets with the size of 2000, 1000, 500, 250, 125, 124, ..., 1.

The procedure for the experiments was the following. First, as a baseline experiment, the SVM-RFE method was applied to the dataset using the original feature elimination strategy and the adaptive strategy to see if there was a significant difference between them. The dataset was randomly stratified into training and test datasets using a tenfold cross validation approach. The SVM-RFE method was utilized on the training set to recursively eliminate the features and test accuracies were predicted on the test set using the classifier built on the training data with selected features. This was done 10 times, once for each left out fold, and an average accuracy over the ten folds is reported. The whole procedure was repeated five times with different randomly chosen stratified sets of data in order to get more reliable results. The results reported are the average values of the five experiments.

Fig. 3 shows the comparison of average weighted accuracies of the SVM-RFE method using the two different feature elimination strategies across different numbers of features. In this figure, RFE-1 represents the original feature elimination approach where the features were recursively removed one by one. RFE-M represents the adaptive feature elimination approach. It can be seen from the results that there is very little difference in the performance for these two feature elimination approaches (p -value = 0.054 using the wilcoxon test). Note that the adaptive feature elimination approach has lower accuracies with less features than the original one, which indicates that some important features may be removed at the beginning using this approach. However the adaptive feature elimination approach is much faster than the original approach. The average running time across the five individual experiments for RFE-1 was approximately 150 minutes. On the other hand, the average running time across the five individual experiments for RFE-M was approximately 60 minutes. The experiment indicates that we can use the proposed feature elimination approach to speed up the running time while maintaining comparable performance.

The results reported in Fig. 3 are the average values of five individual cross-validation trials. Fig. 4 is a graph of the weighted accuracies for the best and worst cases in these five individual trials in the SVM-RFE-1 method with the original feature elimination approach. For the best one, the weighted accuracy

Fig. 3. Comparison of average weighted accuracies of SVM-RFE with different feature elimination approaches at different number of features

increases slightly when the number of features decreases, but it drops dramatically in the worst trial. Therefore, the average weighted accuracies are steadily lower than the initial ones with 2000 features and they vary smoothly across different number of features after taking the average.

Next, for the feature perturbation method, we tried different parameter values for c in Eq. 1 to see if it could significantly impact on the performance. c was chosen as 1, 2, and 3 respectively. Due to the limitation of running time, only the adaptive feature elimination approach was used in the algorithm. The dataset was randomly stratified as training and test datasets for a tenfold cross validation. The feature perturbation method was implemented on the training dataset to recursively eliminate the features, where the noise was randomly generated from the uniform distribution with the parameter c . The SVM classifier was built each time with the current active features and the test dataset with the corresponding active features was predicted to yield the test accuracy. This was done 10 times, once for each left out fold, and average weighted accuracy over the tenfold was reported. The whole procedure is repeated five times with the same five different randomly chosen stratified sets of data as in the SVM-RFE method in order to get more reliable results. The results reported are the average values of the five experiments for each value of c .

Fig. 5 is a graph of the average weighted accuracies of the feature perturbation method with different parameter c at different number of features. The figure shows that in all cases where c equals 1, 2, and 3, the accuracy is stable or drops slightly with a large number of features and then it decreases dramatically when only a few features are retained. Specifically, the overall performance when $c = 2$ is better than the cases when $c = 1$ or $c = 3$. The average weighted accuracy was above 80% for $c = 2$ when using 15 features, as compared to 75.93% for c

Fig. 4. Weighted accuracies of SVM-RFE in the best and worst trials

$= 3$ and 77.34% for $c = 1$. The best average weighted accuracy can be achieved at 86.07% with 34 features when $c = 2$. The average running time across five individual experiments for the feature perturbation method with the adaptive feature elimination approach was approximately 800 minutes, which is more than 10x slower than the SVM-RFE method.

Fig. 6 shows the comparison of the average weighted accuracies of the feature perturbation method and the SVM-RFE method. The adaptive feature elimination approach was used to remove the features recursively. For the feature perturbation method, we chose the optimal parameter value of c as 2. As can be seen in this figure, the feature perturbation method outperformed SVM-RFE significantly in detecting differentially expressed genes for classifying the colon cancer dataset with the most number of features. The performance of the SVM-RFE method is only better than the feature perturbation method when the number of features (genes) is less than 5, where accuracy drops off dramatically in both approaches.

For a better comparison of the two different feature selection methods, statistical tests were applied to measure the significance of improvements in prediction accuracy. The feature perturbation method with different parameter values of c was compared to the SVM-RFE method. As the normality test fails, the Wilcoxon test [8] was used to test the significance of differences across numbers of features. Listed in Table 2 are the p-values of the Wilcoxon test, comparing the feature perturbation method (FPR) and SVM-RFE. The p-values indicate there is a significant improvement in accuracy using the feature perturbation method, as compared to the SVM-RFE method.

Fig. 5. Average weighted accuracy for feature perturbation method with different parameter values of c at different number of features

Fig. 6. Comparison of average weighted accuracy between feature perturbation method and SVM-RFE at different number of features

Table 2. Statistical test of performance improvement

Method 1	Method 2	Wilcoxon test p-value
FPR ($c=1$)	SVM-RFE	$< 2.2e - 16$
FPR ($c=2$)	SVM-RFE	$< 2.2e - 16$
FPR ($c=3$)	SVM-RFE	$< 2.2e - 16$

4 Conclusions

This paper presents a noise-based feature perturbation method as a selection method for microarray data. We began with the hypothesis that truly informative genes will cause a classifier to fail when perturbed by noise. We have described a recursive elimination procedure to remove features that change overall classification accuracy the least when modified by noise. This procedure is similar in form to the SVM-RFE method however it is not constrained to the use of support vector machines. The implementation details and computational complexity were described.

The noise-based feature perturbation method was compared to the SVM-RFE method through a series of experiments on the prediction of colon cancer vs. normal colon tissue. Experimental results showed that the feature perturbation method can achieve higher prediction accuracies with fewer number of features than the SVM-RFE method. However, several issues in the feature perturbation method need further investigation. The introduction of noise is based on the variability of the feature, however more work is required in identifying the optimal factor (the c parameter) to use. Additional strategies for speedup within the implementation will also be important to address.

Acknowledgments. This research was supported by the Department of Defense, National Functional Genomics Center project, under award number DAMD 17-02-2-0051. Views and opinions of, and endorsements by, the author(s) do not reflect those of the US Army or the Department of Defense.

References

1. Inza, I., Larranaga, P., Blanco, R., Cerrolaza, A.J.: Filter versus wrapper gene selection approaches in DNA microarray domains. *Artificial Intelligence Medicine*, 31 (2004) 91-103
2. Xiong, H., Swamy, M.N.S., Ahmad, M.O.: Optimizing the Kernel in the Empirical Feature Space. *IEEE trans. Neural Networks* 16 (2001) 460-474
3. Guyon, I., Weston, J., Barnhill S., Vapnik, V.: Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, 46 (2002) 389-422
4. Rakotomamonjy, A.: Variable Selection using SVM-based Criteria. *Journal of Machine Learning Research*. Vol 3. (2003) 1357-1370
5. Kohavi, R., John, G.H.: Wrappers for Feature Subset Selection. *Artificial Intelligence Archive*, 97 (1997) 273-324
6. Chang, C.C., Lin, C.J.: A Library for Support Vector Machines, *libsvm*. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
7. Witten, I.H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufman Publishers. 2000
8. Lehmann, E.L.: *Nonparametrics: Statistical Methods Based on Ranks*. Francisco, CA: Holden-Day. 1975