Neural, Parallel, and Scientific Computations 21 (2013) 201-216

CLASSIFICATION OF CANCERS BY MICROARRAY GENE EXPRESSION DATA USING THE BEHRENS-FISHER STATISTIC

NABIN K. MANANDHAR SHRESTHA AND KANDETHODY M. RAMACHANDRAN

Department of Mathematics and Statistics
University of South Florida
Tampa, FL 33620 USA

ABSTRACT. Microarray expression experiments allow the recording of expression levels of thousands of genes simultaneously. Such data have been useful for classifying different types of cancers. Majority of literature on this topic assumes equality of variance between control and treatment samples. However the variance of the expression levels in different classes are generally different due to the nature and response of the mRNA at the different conditions, the classification methods should take account of this information. In this paper, we have proposed a new method of selecting informative genes based on the Bayesian Version of Behrens-Fisher distribution. We have found that the proposed method better to others because it selects the genes that are useful for classification and gives the better result. The efficiency of this method has been demonstrated by applying them in three real microarray data. We have compared our result with some of the other popular methods that are found in the literature.

AMS (MOS) Subject Classification. 39A10

1. Introduction

Classification of biological samples using the gene expression data is a broad area in functional genomic and has drawn much attention in the field of cancer classification that evolved from the Leukemia data set first analyzed by Golub *et. al.* [6]. Various methods have been suggested in building classifier. Microarray data is a special type of data, which is distinguished from other types of data by the fact that the number of genes (predictors) is usually much larger than the number of samples. More importantly, only a small fraction of genes are meaningful for classifying the samples. Thus identification of these genes not only makes the computation easier but it reduces the cost and time. Selecting differentially expressed genes between different classes can be taken as the marker genes for the classification. It is because these are the genes that play vital role that shows differential expression between samples. This idea can be extended if there are more than two conditions or classes. Golub *et. al.* [6] have used the weighted voting criteria for selecting the marker genes

Received April 2, 2013

for the classification, but the problem with this approach is the number of genes that is to be selected as informative. The maximum margin classifiers, like support vector machines [12] has been used for the classification. This method, being the sophisticated mathematical result, has shown the promising result on the field of pattern recognition and classification. The main problem with support vector machines is that it uses all the genes for the classification of samples. Guyon et. al. (2002) proposed the feature selection by recursive elimination method and showed that just few of the genes are useful for the classification.

In this paper, we briefly describe different methods used for the classification and propose our method of gene selection as the feature selection in the small sample high dimensional data for the classification .

The structure of the paper is as follows. Section 2 reviews the literatures in the microarray classification. In section 3, we propose our method - called Behrens Fisher (BF) method, for selecting genes that are used for classification. Section 4 discusses about the application of new method and resulting classification.

2. Review of classification Methods

In the case of microarray data, the number of genes p are far greater than the number of samples n. This creates problems for classification of samples into different classes. The prediction rule may not be able to be formed by using all of the p genes. Even if we could use all of the genes, the noise associated with the genes having little or no discriminatory power makes the classification process unsuitable. The generalization error does not decrease although the training error is small. Although different classification methods uses some or all of the genes, they do not classify the samples without error. Actually, the methods are data dependent. One method gives better result in one data, while other method gives the better classification in another data.

2.1. **Nearest Shrunken Centroids Method.** The purpose of discriminant analysis or classification is to assign samples to one of the several (G) classes based on a set of measurements $\mathbf{x} = (x_1, x_2, \ldots, x_p)$ measured from samples. In the case of supervised learning, the classes are predetermined from a set of samples, called *training samples.* These training samples are used to build a classifier. The classifier is then used to determine the class of a new sample. When a sample is misclassified, then an error is said to be incurred. The cost or loss associated with such an error is defined as

$$L(k, \hat{k}) = \begin{cases} 0, & \text{if } k = \hat{k}\ ; \\ 1, & \text{otherwise.} \end{cases} \tag{2.1}$$

where k is the correct group of the sample and $\hat{k}$ is the assignment made to that sample by the classifier. If the class conditional densities, $f_k(\mathbf{x})$, and the class priors,

π_k, of class k are known, then Bayesian optimal rule of classification to classify a new sample $\mathbf{x}^*$ is to minimize the risk

$$R(\hat{k}|\mathbf{x}) = \sum_{k=1}^{G} L(k, \hat{k}) \ \Pr(G = k|\mathbf{X} = \mathbf{x}) \tag{2.2}$$

Then the classification rule is:

$$\hat{k} = argmax_k \ f_k(\mathbf{x}^*)\pi_k.$$

But, the problem is we do not know the class conditional densities, $f_k(\mathbf{x})$, of each of the classes. So, many researchers assume that these densities are multivariate normal with densities,

$$f_k(\mathbf{x}) = (2\pi)^{-p/2}|\mathbf{\Sigma}_k|^{-1/2} \ \exp\left[\frac{-1}{2} \ (\mathbf{x} - \mu_k)'\mathbf{\Sigma}_k^{-1}(\mathbf{x} - \mu_k)\right]$$

Then assuming the equal variance-covariance matrix of each class, $\mathbf{\Sigma}_k = \mathbf{\Sigma}$, the linear discriminant score

$$D_k^l(\mathbf{x}) = \mathbf{x}'\mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k'\mathbf{\Sigma}^{-1}\mu_k + \log \pi_k \tag{2.3}$$

and assuming classwise covariance matrices unequal, the quadratic discriminant score

$$D_k^q(\mathbf{x}) = -\frac{1}{2}\log|\mathbf{\Sigma}_k| - \frac{1}{2}(\mathbf{x} - \mu_k)'\mathbf{\Sigma}_k^{-1}(\mathbf{x} - \mu_k) + \log \pi_k \tag{2.4}$$

Generally, the maximum likelihood estimate are used to estimate the population mean and population variance. Furthermore, the empirical probability is used to estimate the class priors.

In the case of DNA microarray data, the number of covariates (genes), p, which are in the order of several thousands, are much greater than the number of samples, n_k, generally within hundreds, in each class. So, the sample covariance matrix is singular and this gives the unreliable estimate of covariance matrix because of high variability. Friedman [1] introduced the regularized discriminant analysis in which the the unequal variances were shrinked towards the common variance using the regularization, thus increasing the performance of the RDA classifier. The microarray problem is thus unique and challenging. Since the expression level of most of the genes are same in two different treatment samples, those genes contribute little in the case of classification. Thus it is important to identify the genes that actually contribute for the classification. Assuming that those genes which have common class means do not contribute for the classification, Tibshirani *et. al.* [2] proposed the Nearest Shrunken Centroids (NSC) method. They used the *shrinkage parameter*, Δ, for thresholding and declared those genes as non-contributing genes if the shrunken centroids for gene g in class k,

$$\bar{x}'_{gk} \quad \text{shrinks to the overall mean} \quad \bar{x}_g$$

when the

$$|d_{gk}| - \Delta \leq 0$$

where

$$s_g^2 = \frac{1}{n-G} \sum_{k=1}^{G} \sum_{j \in C_k} (x_{gj} - \bar{x}_{gk})^2, \qquad d_{gk} = \frac{\bar{x}_{gk} - \bar{x}_g}{\sqrt{1/n_k + 1/n} \cdot s_g^2}$$

The non-contributing genes are removed from the data, thus reducing the dimensionality of the gene-matrix. The discriminant score of the NSC classifier was defined as

$$D_k(\mathbf{x}^*) = \sum_{g=1}^{p} \frac{(x_g^* - \bar{x}'_{gk})^2}{s_g^2} - 2 \log \pi_k \tag{2.5}$$

In the classification process, the genes g which have each of the shrunken class-means $\bar{x}'_{gk}$ shrinks towards the overall class means $\bar{x}_g$ in each class $k = 1, 2, \ldots, G$ have the same $(x_g^* - \bar{x}'_{gk})^2$ values. So, the numerator of the above discriminant score (2.5) can be replaced by the square of differences of only those genes g for which

$$\bar{x}'_{gk} \neq \bar{x}_g, \qquad \forall k = 1, 2, \ldots, G$$

The optimal value of the shrinkage parameter, Δ, is chosen by the cross validation that minimizes the cross-validation error. The idea of cross-validation is to obtain the unbiased estimate of future prediction error associated with a particular observation and is obtained by removing it from the model. This gives the genes that are useful for classification.

2.2. **Weighted Voting Method.** Here we briefly review the methods of classification that are used by Golub *et. al.* To identify the genes which are truly expressed in the new samples, one can use the *weighted voting scheme* (WVS) method. This uses a weighted linear combination of the "marker" or "relevant" genes obtained in the training set to classify the new sample. In this method, the correlation between the expression values of a gene g in two classes is defined as

$$w_g = \frac{\mu_{g1} - \mu_{g2}}{\sigma_{g1} + \sigma_{g2}} \tag{2.6}$$

where μ_{gi} and σ_{gi} are the mean and standard deviations of gene g in the class i, $i = 1, 2$. The larger the absolute value $|w_g|$ is the more important the gene g is for prediction. The genes are ranked by their $|w_g|$'s and top ones are selected. These top selected genes are the *marker* or *informative* genes.

For each *informative* gene g in the training sample, let μ_{g1} and μ_{g2} be the means and σ_{g1} and σ_{g2} be the standard deviations respectively. Then the weight of gene g is determined by

$$w_g = \frac{\mu_{g1} - \mu_{g2}}{\sigma_{g1} + \sigma_{g2}} \tag{2.7}$$

This measure is also called the signal-to-noise ratio. This weighting factor reflects the correlation between the expression level of gene g and class distinction. The parameter b_g is calculated as

$$b_g = \frac{\mu_{g1} + \mu_{g2}}{2}, \tag{2.8}$$

which is the average mean of expression levels of two classes. Hence we define the parameters (w_g, b_g) for each *informative* gene in the training set. For a new sample $\mathbf{x}^*$ with x_g^* being the normalized log expression level of the gene g, we calculate the votes casted by each of the genes in the "informative" set. The vote of a gene g is

$$v_g = w_g(x_g^* - b_g) = \frac{\mu_{g1} - \mu_{g2}}{\sigma_{g1} + \sigma_{g2}}[x_g^* - \frac{\mu_{g1} + \mu_{g2}}{2}] \tag{2.9}$$

A positive vote indicates that the sample belongs to class 1 and negative vote indicates it being in class -1. Then the total vote for the sample to be in class 1 is obtained by adding $V_1 = \sum_g \max(v_g, 0)$ and the total vote for sample to be in class -1 is $V_2 = \sum_g \max(-v_g, 0)$. Then the sample is assigned to that class corresponding to the higher total vote. Generally, we take the 5% of most positive and 5% most negative genes as the "informative" genes in the training set. But this number is a free parameter and depends on the user.

2.3. **Dudoit's Multi-class Classification Method.** Several proposals have been made for ranking the genes for multiclass classification. Dudoit *et al.* (2002) used the ratio of between-sum-squares to within-sum-squares of each gene for the multiclass classification. Explicitly, let there are G classes and the number of samples be n each of dimension p. Then, the samples in each class

$$n = n_1 + n_2 + \cdots + n_G$$

Let $\bar{x}_g$ be the mean of gene g over all classes. For each gene g, $g = 1, 2, \ldots, p$, let $\bar{x}_g^{(k)}$ be the mean in class k, $k = 1, 2, \ldots, G$. Then the ranking of genes are done using the ratio

$$\rho_g = |\frac{\sum_{k=1}^G n_k(\bar{x}_g^{(k)} - \bar{x}_g)^2}{(n - G)\sigma_g^2}|$$

where σ_g is the pooled within class standard deviation of gene g:

$$\sigma_g^2 = \frac{1}{(n-G)} \sum_{k=1}^{G} (n_k - 1)\sigma_g^{(k)2}$$

A new sample $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_p^*)$ is then classified into class k, if

$$k = \min_{k'} \; \|\mathbf{x}'^* - \bar{\mathbf{x}}^{(k')}\|$$

where $\| \cdot \|$ is the Euclidean norm, and $\mathbf{x}'^*$ and $\bar{\mathbf{x}}^{(k')}$ are the component vector and mean of class k' of only those component genes selected by the ranking procedure.

2.4. **Support Vector Machines (SVM) Methods.** In the SVM classification method of linearly separable samples, one finds the separating hyperplane

$$f(\mathbf{x}) = b + \mathbf{w}'\mathbf{x} \tag{2.10}$$

from the training samples $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$ where $y_i = \pm 1$ are binary class labels, that correctly classifies the training samples and maximizes the margin. The class of a new sample $\mathbf{x}$ is determined by the sign$[f(\mathbf{x})]$. All the training samples are classified correctly if

$$y_i(b + \mathbf{w}'\mathbf{x}_i) \geq 1 \qquad \text{for all } i$$

The hyperplanes

$$b + \mathbf{w}'\mathbf{x} = \pm 1$$

are called the canonical hyperplanes and the distance $1/\mathbf{w}$ between one of the canonical hyperplanes and separating hyperplane (2.10) is the margin. So, the optimization problem can be rephrased as

$$\min_{w} \|\mathbf{w}\|$$

subject to

$$y_i(b + \mathbf{w}'\mathbf{x}_i) \geq 1 \qquad \text{for all } i$$

For the non-separable case, we still maximize the margin but we allow some points on the wrong side of the hyperplane defining the slack variables $\xi = (\xi_1, \xi_2, \ldots, \xi_n)$. The optimization problem is

$$\min \|\mathbf{w}\| + \gamma \sum_{i=1}^{N} \xi_i$$

subject to

$$y_i(b + \mathbf{w}'\mathbf{x}_i) \geq 1 - \xi_i, \quad \xi_i \geq 0 \qquad \text{for all } i$$

where γ is the cost parameter that is determined by cross-validation. Using the Karush -Kuhn-Tucker condition, the optimal values of the parameters of the hyperplane are obtained as

$$\hat{\mathbf{w}} = \sum_{i \in SV} \alpha_i y_i \mathbf{x}_i$$

where $0 < \alpha_i \leq \gamma$. The sum $\sum_i \alpha_i y_i = 0$ corresponds to the *support vectors* $\mathbf{x}_i$. The bias parameter

$$\hat{b} = \frac{1}{n_0}\{\sum_{i \in SV} y_i - \sum_{i,j \in SV} \alpha_i y_i \mathbf{x}_i' \mathbf{x}_j'\}$$

where n_0 is the number of support vectors. Since $\alpha_i = 0$ for the non-support vectors $\mathbf{x}_i$, the summation indices i and j are only for the support vectors. The separating hyperplane is thus given by

$$f(\mathbf{x}) = \hat{b} + \hat{\mathbf{w}}'\mathbf{x} \tag{2.11}$$

and the corresponding decision rule for a new sample $\mathbf{x}^*$ is:

$$f(\mathbf{x}^*) = \hat{\mathbf{w}}'\mathbf{x}^* + \hat{b} \begin{cases} > 0, & \text{Classify to class 1;} \\ < 0, & \text{Classify to class 2.} \end{cases} \tag{2.12}$$

Using the non linear basis functions $\phi(\mathbf{x}_i)$, one can map the input space into a high dimensional feature space. Then the samples are classified by the linear boundaries in the feature space using the kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ which corresponds to the non linear boundaries in the input space. The separating hyperplane in the feature pace is

$$f(\mathbf{x}) = \hat{b} + \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) \tag{2.13}$$

Different kernels are obtained by

$$K(\mathbf{y}, \mathbf{x}) = \phi'(\mathbf{y})\phi(\mathbf{x})$$

where $\phi(\mathbf{y})$ and $\phi(\mathbf{x})$ can be any linear or non linear transformations of $\mathbf{y}$ and $\mathbf{x}$ and must satisfy the Mercer's Conditions [11]. But for this work, we use simple linear kernel,

$$K(\mathbf{y}, \mathbf{x}) = \phi'(\mathbf{y})\phi(\mathbf{x}) = \mathbf{y}'\mathbf{x} + 1$$

3. Behrens Fisher Statistic

3.1. Gene Selection by Behrens-Fisher Statistic.

Suppose there are G different classes in the population. Let n_k be the number of samples in class k, $(k = 1, 2, \ldots, G)$. Let $\mathbf{x}_{gk} = (x_{g_1}, x_{g_2}, \ldots, x_{g_{n_k}}) \overset{iid}{\sim} N(\mu_{gk}, \sigma^2_{gk})$ be the expression level (possibly log transformed) of the gene g in class k. For k=1,2,...,G, the density of $\mathbf{x}_{gk}$ can be written as

$$f(\mathbf{x}_{gk}) = \frac{1}{(\sigma_{gk})^{n_k}(2\pi)^{\frac{n_k}{2}}} \; exp\Big[-\frac{1}{2\sigma^2_{gk}} \{n_k - 1)s^2_{gk} + n_k(\bar{x}_{gk} - \mu_{gk})^2\}\Big] \tag{3.1}$$

where $\bar{x}_{gk}$ and s^2_{gk} are sample mean and sample variance of gene g in class k respectively.

Assuming the independency of location parameter μ_{gk} and scale parameter σ^2_{gk}, the joint prior for μ_{gk} and σ^2_{gk} can be written as

$$p(\mu_{gk}, \sigma^2_{gk}) = p(\mu_{gk})p(\sigma^2_{gk}) \tag{3.2}$$

Assume that the priors for μ_{g1} and μ_{g2} are flat priors and the priors for σ^2_{g1} and σ^2_{g2} are scaled inverse χ^2 distributions, *i.e.* $p(\sigma^2_{g1}) = I(\sigma^2_{g1}; \nu_0, \sigma^2_0)$ and $p(\sigma^2_{g2}) = I(\sigma^2_{g2}; \eta_0, \tau^2_0)$, where $\alpha = (\nu_0, \eta_0, \sigma^2_0, \tau^2_0)$ is the hyper-parameters that should be estimated from the data.

Let $\Delta\mu_g = \mu_{g2} - \mu_{g1}$. Then the statistic, called the BF-statistic

$$B = \frac{\Delta\mu_g - (\bar{x}_{g2} - \bar{x}_{g1})}{(\frac{\sigma_{g1}^2}{n_1} + \frac{\sigma_{g2}^2}{n_2})^{\frac{1}{2}}} = B_{x_2}\cos\theta - B_{x_1}\sin\theta \tag{3.3}$$

where

$\tan\theta = \frac{\sigma_{g1}/\sqrt{n_1}}{\sigma_{g2}/\sqrt{n_2}}, \qquad 0 \le \theta \le \frac{\pi}{2}$

$B_{x_1} = \frac{(\mu_{g1} - \bar{x}_{g1})}{\sigma_{g1}/\sqrt{n_1}}$

$B_{x_2} = \frac{(\mu_{g2} - \bar{x}_{g2})}{\sigma_{g2}/\sqrt{n_2}}$

and, B_{x_1} and B_{x_2} are independently distributed as t-statistics with v_{n_1} and v_{n_2} degrees of freedom respectively. Hence, the statistic B is distributed as the Behrens-Fisher distribution with

$$v_{n_1} = n_1 + \nu_0 - 1, \quad \texttt{and} \quad v_{n_2} = n_2 + \eta_0 - 1$$

degrees of freedom [5]. That is,

$$B \sim BF(v_{n_1}, v_{n_2}, \theta)$$

with pdf

$$f(\beta|\mu_{g1}, \mu_{g2}, \sigma_{g1}^2, \sigma_{g2}^2) = k\int_{-\infty}^{\infty}\Big[1 + \frac{(\alpha\cos\theta - \beta\sin\theta)^2}{v_{n_1}}\Big]^{-\frac{v_{n_1}+1}{2}} \times \Big[1 + \frac{(\alpha\sin\theta + \beta\cos\theta)^2}{v_{n_2}}\Big]^{-\frac{v_{n_2}+1}{2}} d\alpha$$

where

$$\alpha = B_{x_2}\sin\theta + B_{x_1}\cos\theta, \quad \beta = B_{x_2}\cos\theta - B_{x_1}\sin\theta$$

This can be further approximated by scaled t-statistic [7]:

$$\frac{B}{a} \sim t(b) \tag{3.4}$$

where

$f_1 = \left(\frac{v_{n1}}{v_{n2}-2}\right)\cos^2\theta + \left(\frac{v_{n1}}{v_{n1}-2}\right)\sin^2\theta$

$f_2 = \frac{v_{n2}^2}{(v_{n2}-2)^2(v_{n2}-4)}\cos^4\theta + \frac{v_{n1}^2}{(v_{n1}-2)^2(v_{n1}-4)}\sin^4\theta$

$a^2 = \frac{(b-2)}{b}f_1$

$b = 4 + \frac{f_1^2}{f_2}$

$\cos^2\theta = \frac{\frac{\sigma_{g2}^2}{n2}}{\left(\frac{\sigma_{g2}^2}{n2} + \frac{\sigma_{g1}^2}{n1}\right)}, \qquad \sin^2\theta = 1 - \cos^2\theta.$

That is, B has approximately t-distribution with b degrees of freedom ($b \geq 1$) and scale parameter a. This statistic B can also be denoted as $B \sim t(0, a^2, b)$ and is valid only for $v_{n1}, v_{n2} \geq 5$.

3.2. Choosing the number of genes required for classification.

For this work we select those genes that are uniformly expressed in each of the classes. Since leave-one-out cross validation (LOOCV) error is almost unbiased estimate of generalization error [12], we use the leave-one-out cross validation on the training samples. The genes that are common in each of the cross validated training samples are the preliminary set of genes that have the power to discriminate between different classes. From this preliminary gene set, only those genes are selected that do not further decrease the cross validation error. This final set is the *optimal set* that is useful for the classification. Since the genes chosen by the t-test are the marker genes for the two different conditions, these can be used to classify samples into any one of the two classes. The genes selected by BF method may not be appropriate for the multi-class classification. To fit with the multi-class classification, we want to choose the marker genes that are useful. For this, we can use the *one-versus-all* method. In this method, we take one of the class as from normal (condition 1) and combine the rest of classes and take that combined classes as from diseased (condition 2). Then we use the leave-one out method in the training set to choose the genes that are useful for the classification. We simply leave one of the training sample and find the genes that are differentially expressed by the BF method in the rest of training samples that contains samples from both conditions. We repeat this procedure for all the samples in the training set. Then the differentially expressed genes that are common in each of the leave-one-out training set is taken as the marker genes, which we choose as the genes useful for classification. We repeat the same procedure for the rest of the classes (leaving the classes that were used) and get the differentially expressed genes. Finally, the *informative* genes are those common genes that is found expressed, thus uniformly expressed, in all the classes and leave one out training sets. Using this gene set, we classify one sample with the rest. In the above example, we classify the ALL sample with the rest. Then, this is repeated for all remaining classes. This method is the one-versus-all method.

4. Results

4.1. Datasets Pre-processing and Filtering.

All the data sets used in this paper are oligonucleotide microarray data and was pre-processed as in Dudoit *et. al.* (2002). The threshold was set with floor of 100 units and ceiling of 16,000 units. A ceiling of 16,000 units was chosen because it is at this level that we observe the flourescense saturation of the scanner; values above this can't be reliable measure. Similarly a floor of 100 units was chosen to minimize the noise and maximise the interpretation of marker genes due to the correlation of genes. We have filtered out (excluded) those low quality genes that have ratio (max / min) < 5 and (max $-$ min) < 500 across all of the samples. To make the data somewhat symmetrical, base-10 logarithm has been used for the transformation.

4.2. **MLL Leukemia Data.** MLL Leukemia data is Affymetrix oligonucleotide data and consists of 72 samples and 12,582 genes. There are 3 different classes - ALL, MLL, and AML. ALL has 20 training sample and 4 testing samples, MLL has 17 training samples and 3 test samples and AML has 20 training samples and 8 testing samples. After the preprocessing and filtering the low quality data, we are left with 8,681 genes.

Table 1 shows the comparison and performance of different methods for this data. The nearest shrunken centroid (NSC) method chooses only 12 genes but the performance of the model in the testing samples are not as good as in the other method. It makes four errors when classifying the training samples. The genes chosen by the Beherens-Fisher statistic have more discriminating power, as seen these genes used in weighted voting, Dudoit and Support Vector Machines methods. In all methods, misclassification occurs only in the training samples. The SVM method is seem to be the perfect classifier, since it makes no error in training and testing samples.

TABLE 1. *Comparison of Classification Performance on MLL Leukemia Data.*

Method	Training Errors			Test Errors			Average Error		No. of
	ALL	*MLL*	*AML*	*ALL*	*MLL*	*AML*	*Train*	*Test*	Genes
Dudoit	0	1	1	0	0	0	2	0	23
Wt. Vote	0	1	1	0	0	0	2	0	17
SVM	0	0	0	0	0	0	0	0	17
NSC	0	4	0	0	0	0	4	0	12

4.3. **Golub Leukemia Data.** Golub Leukemia data consists of 7,129 genes and 72 samples. These samples are from two classes: Acute Lymphoid Leukemia (ALL) and Acute Myeloid Leukemia (AML). We have chosen 38 training samples: 27 ALL and 11 AML, and 34 testing samples : 20 ALL and 14 AML as in Golub *et al.* [6]. After pre-processing and filtering, 3571 genes are remained. For the classification, 33 genes were selected by the BF method using LOOCV error. These genes were used for the classification of ALL and AML samples. The classification error are shown in Figure 2. The genes selected by BF method seems optimal set in the sense that it makes very few error in classifying the samples. It makes no error while using the

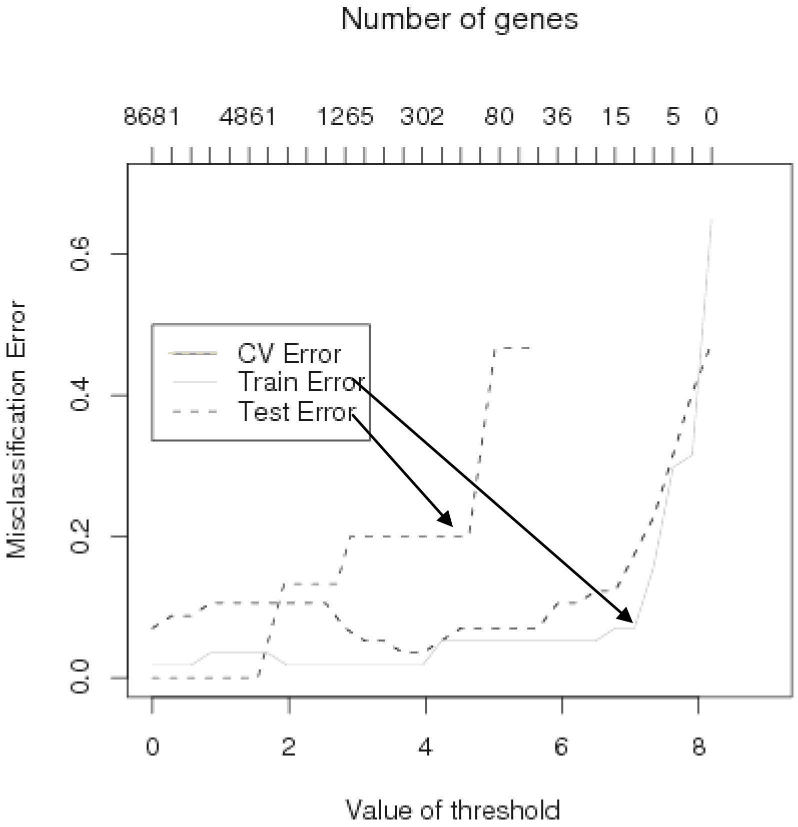

FIGURE 1. Cross-Validation Error of Nearest Shrunken Centroid Method for MLL data

SVM method, whereas it makes 1 error out of 38 training samples and 1 error out of 34 testing samples. As the number of genes increased, the error does not decrease when we select 80 genes. By taking 8 genes, it has been found that 3 errors were made in the training samples and 2 errors were made on the testing sample.

TABLE 2. *Comparison of Classification Performance on Golub Data.*

Method	Training Errors		Test Errors		No.of Genes
	ALL	*MLL*	*ALL*	*MLL*	
W. Vote	1	0	1	0	33
SVM	0	0	0	0	33
NSC	0	0	1	0	18

4.4. **Central Nervous System Embryonal Tumor Data.** This data consists of 12,625 genes and 327 samples. There are 7 classes and the samples are separated as training and test samples. For the comparison purposes, we have selected the same 215 training and 112 test samples as in Pomeroy *et. al.* (2001). These are : BCR (9 train, 6 test), E2A (18 train, 9 test), HYP (42 train, 22 test), MLL (14 train, 6 test),

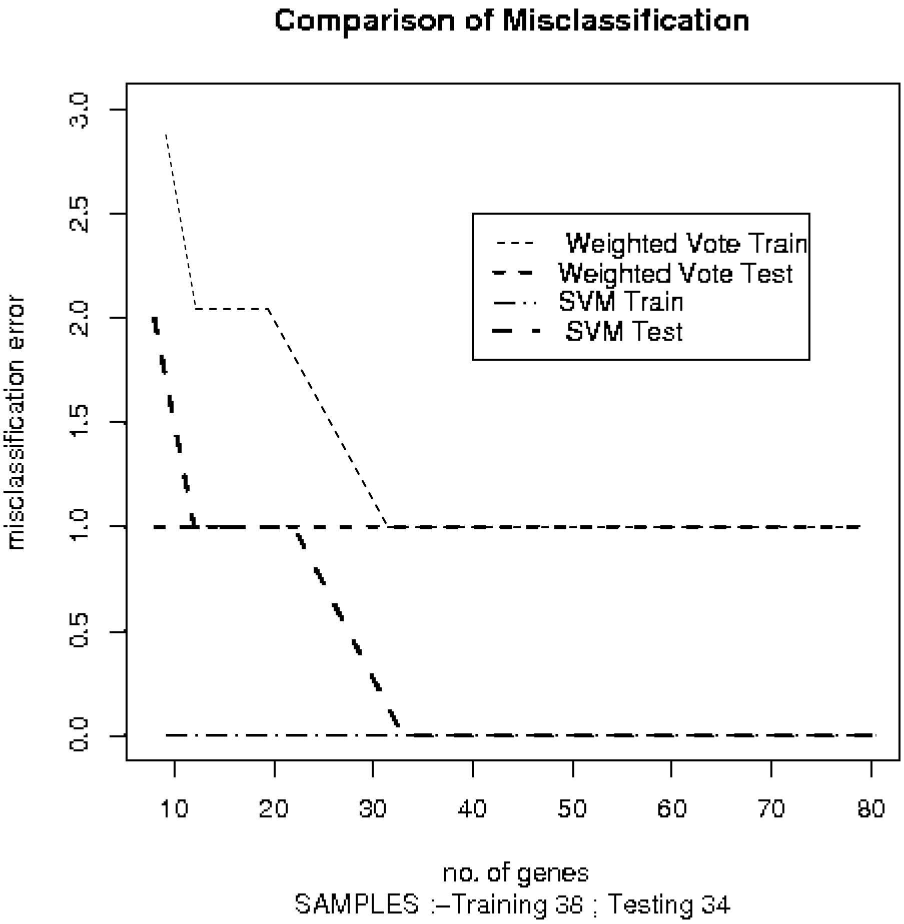

Figure 2. Comparison of Errors in Weighted Vote and SVM in Golub Data

T.ALL (28 train, 15 test), TEL.AML (52 train, 27 test), Others (52 train, 27 test). After applying the pre-processing and filtering steps, ther are 12,061 genes. For the sake of convenience, we call this data as ALL-7 data.

In this 7 classes case, the genes selected by the BF method has shown the promising result over the nearest shrunken centroid (NSC) method and shrunken centroid regularized discriminant analysis (SCRDA) method of Guo *et. al.* (2007). We have used the genes selected by the proposed BF statistic for the Weighted Voting method. For the Dudoit method, we have used the 150 genes selected by the BF statistic. The classification performed by both weighted vote and Dudoit method are better than the rest two methods. The overall error for the weighted vote method in the training samples is .0093 whereas it is 0.1209 and 0.0651 in NSC and SCRDA methods respectively. Similarly, the overall test errors in the weighted voting method is 0.0625, and that is 0.3482 and 0.1160 in NSC and SCRDA methods respectively. Another important fact to note is that the number of genes selected for the classification by BF method is also comparable to both of these methods.

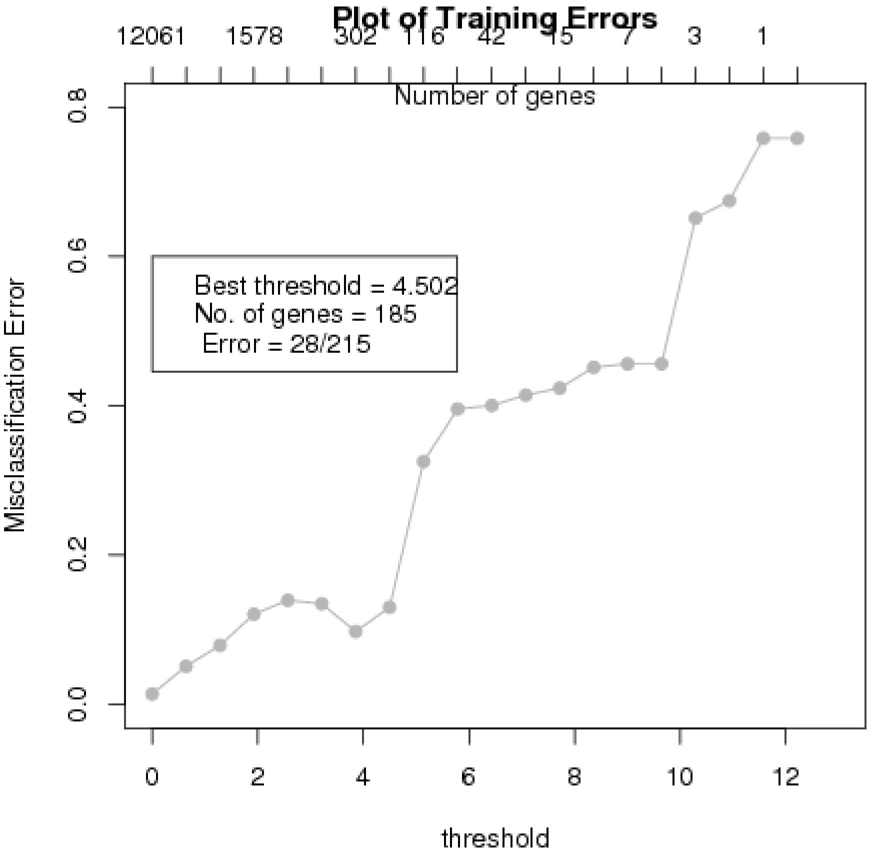

FIGURE 3. Training Error in ALL -7 Class Data using NSC Method

TABLE 3. *Comparison of Classification Performance on ALL-7 Data.*

	Method	Classes							Total	No.of
		BCR	*E2A*	*HYP*	*MLL*	*T.ALL*	*TEL.AML*	*Others*	Error	Genes
Tr	W. Vote	0	0	0	0	0	0	2	2	370
a	Dudoit	2	0	1	0	0	1	16	20	150
i	NSC	9	0	7	8	0	0	2	26	185
n	SCRDA	0	0	2	0	0	0	12	14	543
T	W. Vote	0	0	3	0	0	0	4	7	370
e	Dudoit	1	0	1	0	0	1	3	6	150
s	NSC	6	0	21	6	0	0	6	39	185
t	SCRDA	1	0	2	0	0	0	10	13	543

The confusion matrix for the different methods are shown in Table 4. and Table 5. It is the matrix of number of samples classified by the method, and shows explicitly how many samples are misclassified and in which class they were assigned by the

TABLE 4. *Confusion matrix for the MLL-3 training data by Dudoit method, and Golub Test Data by NSC Method.*

Class	**MLL Data**			**Golub Data**	
	Predicted			Predicted	
True	ALL	AML	MLL	ALL	AML
ALL	20	0	0	20	0
AML	0	19	1	1	13
MLL	0	1	16		

classifier. Since the weighted vote method and SVM method are two class classifier, confusion matrix can not be calculated for this classifier.

TABLE 5. *Confusion matrix for the ALL-7 test data by SCRDA method.*

Class	Predicted						
True	BCR	$E2A$	HYP	MLL	$T.ALL$	$TEL.AML$	$Others$
BCR	5	0	0	0	0	0	1
$E2A$	0	9	0	0	0	0	0
HYP	0	0	20	0	0	0	2
MLL	0	0	0	6	0	0	0
$T.ALL$	0	0	0	0	15	0	0
$TEL.AML$	0	0	0	0	0	27	0
$Others$	4	0	3	0	0	3	17

We compare the performance of the classifiers by the accuracy of it to classify the test samples. The estimated classification or accuracy rate,

$$\text{Accuracy Rate} = \frac{\text{sum of truely classified test samples in different classes}}{\text{total no. of testing samples}}$$

In the case of support vector machines, we have used one-against all method to get the accuracy. This method uses all the genes in the samples. It is seen that the genes chosen by the Behrens Fisher distribution is actually useful for the classification.

TABLE 6. *Accuracy Rate for the ALL-7 test data*

Data	no. of Genes	no. of Classes	no. of samples		Accuracy				
			Train	Test	WV	Dudoit	NSC	SCRDA	SVM
Golub	7,129	2	38	34	97.06	94.11	97.06	97.06	79.41
MLL	12,582	3	57	15	100	100	100	100	100
ALL-7	12,625	7	215	112	93.75	94.64	65.17	88.40	84.82

REFERENCES

[1] Friedman, J. H.: **Regularized discriminant Analysis**, *JASA*, 1989.

[2] Tibshirani *et. al.*: **Class prediction by Nearest Shrunken Centroids with Application to DNA microarrays**, *Statistical Sciences*, 2003.

[3] Fox and Dimmic: **A two-sample *t*-test for Microarray data**; BMC *Bioinformatics*, Vol. **7**, 2006.

[4] Zhang, Aidong: **Advanced Analysis of Gene Expression Microarray Data**, *World Scientific*, 2006.

[5] Manandhar Shrestha, Nabin and Ramachandran K.: **Behrens-Fisher's Distribution for Selecting Differentially Expressed Genes**, *NPSC*, Vol. **16**, 2008.

[6] Golub *et. al.*: **Molecular Classification of Cancer: Class Discovery and Class Predictions by Gene Expression Monitoring**; *Science*, Vol. **286**, 1999.

[7] Patil, V. H. : **Approximation to the Behrens-Fisher Distributions**; *Biometrika*, Vol. **1**, 1965.

[8] Tusher, Tibshirani and Chu: **Significance Analysis of Microarrays applied to the Ionizing Radiation Response**; *PNAS*, Vol. **98**, 2001.

[9] Best and Rayner: **Welch's Approximate Solution for the Behrens Fisher Problem** ; *Technometrics*, Vol. **29 (II)**, 1987.

[10] Webb, Andrew: **Statistical Pattern Recognition**; *John Wiley and Sons, 2nd Edition*, 2002.

[11] Herbrich, Ralf: **Learning Kernel Classifiers**; *MIT Press, Cambridge, Massachusetts*, 2002.

[12] Vapnik, Vladimir: **Statistical Learning Theory**; *John Wiley and Sons*, 1998