BIOINFORMATICS ORIGINAL PAPER

Vol. 23 no. 8 2007, pages 972–979
doi:10.1093/bioinformatics/btm046

Gene expression

Improved centroids estimation for the nearest shrunken centroid classifier

Sijian Wang[1] and Ji Zhu[2,*]

[1]Department of Biostatistics and [2]Department of Statistics, University of Michigan, Ann Arbor, MI, 48109, USA

Received on November 17, 2006; revised on January 19, 2007; accepted on February 4, 2007

Advance Access publication March 24, 2007

Associate Editor: John Quackenbush

ABSTRACT

Motivation: The nearest shrunken centroid (NSC) method has been successfully applied in many DNA-microarray classification problems. The NSC uses 'shrunken' centroids as prototypes for each class and identifies subsets of genes that best characterize each class. Classification is then made to the nearest (shrunken) centroid. The NSC is very easy to implement and very easy to interpret, however, it has drawbacks.

Results: We show that the NSC method can be interpreted in the framework of LASSO regression. Based on that, we consider two new methods, adaptive L_∞-norm penalized NSC (ALP-NSC) and adaptive hierarchically penalized NSC (AHP-NSC), with two different penalty functions for microarray classification, which improve over the NSC. Unlike the L_1-norm penalty used in LASSO, the penalty terms that we consider make use of the fact that parameters belonging to one gene should be treated as a natural group. Numerical results indicate that the two new methods tend to remove irrelevant genes more effectively and provide better classification results than the L_1-norm approach.

Availability: R code for the ALP-NSC and the AHP-NSC algorithms are available from authors upon request.

Contact: jizhu@umich.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Class prediction with high-dimensional microarray data has recently received much attention in many fields, such as bioinformatics, machine learning, medicine and statistics (Alizadeh *et al*., 2000; Dabney, 2005; Dudoit *et al*., 2002; Eisen *et al*., 1998; Golub *et al*., 1999; Hastie *et al*., 2001; Khan *et al*., 2001; Liu and Shen, 2006; Pan, 2002; Shen *et al*., 2006; Wu, 2006; Zhang *et al*., 2006a). It is considered very helpful for medical research if one can classify and predict the clinical category of a sample based on its gene expression profile. The microarray classification problem is a very challenging task, however, because there are a huge number of variables (genes) but a much smaller number of samples. Hence, finding relevant genes that distinguish samples is also greatly desired in practice.

*To whom correspondence should be addressed.

Tibshirani *et al*. (2002) proposed the nearest shrunken centroid (NSC) method for class prediction in DNA-microarray studies. The NSC uses 'shrunken' centroids as prototypes for each class and identifies subsets of genes that best characterize each class. We describe the NSC algorithm briefly in the next subsection.

1.1 The nearest shrunken centroids method

Assuming we have n samples, and for each sample, we have expressions for p genes. Let x_{ij} be the expression for the jth gene and the ith sample. Each sample belongs to one of K classes $1, 2, \ldots, K$. Let C_k be the set of indices of the n_k samples in class k. The jth component of the centroid for class k is $\bar{x}_{kj} = \sum_{i \in C_k} x_{ij}/n_k$, the mean expression in class k for gene j; the jth component of the overall centroid is $\bar{x}_j = \sum_{i=1}^{n} x_{ij}/n$.

Let

$$\mu_{kj}^0 = \frac{\bar{x}_{kj} - \bar{x}_j}{m_k \cdot s_j}, \tag{1}$$

where, s_j is the pooled within-class SD for the jth gene:

$$s_j^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i \in C_k} (x_{ij} - \bar{x}_{kj})^2, \tag{2}$$

and $m_k = \sqrt{1/n_k - 1/n}$.

The NSC shrinks each μ_{kj}^0 to

$$\hat{\mu}_{kj} = \mathrm{sgn}(\mu_{kj}^0)(|\mu_{kj}^0| - \lambda)_+, \tag{3}$$

where λ is a tuning parameter, and the shrunken centroid in class k for gene j is constructed as:

$$\hat{x}_{kj} = \bar{x}_j + \hat{\mu}_{kj} \cdot m_k \cdot s_j. \tag{4}$$

Because many of the $\bar{x}_{kj}$ values are noisy and close to the overall mean $\bar{x}_j$, soft thresholding usually produces more reliable estimates of the true means, and if λ is large enough, some of the μ_{kj}^0 can be shrunken to zero, hence the corresponding $\hat{x}_{kj}$ are equal to $\bar{x}_j$.

For the classification of a test sample, suppose we have one with expression levels $x^* = (x_1^*, x_2^*, \ldots, x_p^*)$. We define the discriminant score for class k as

$$\delta_k(x^*) = \sum_{j=1}^{p} \frac{(x_j^* - \hat{x}_{kj})^2}{s_j^2} - 2 \log \pi_k, \tag{5}$$

where $\pi_k = n_k/n$, and the classification rule is given by

$$C(x^*) = k^*, \quad \text{where } k^* = \arg\min_k \delta_k(x^*). \tag{6}$$

So we can see, if for some j, all $\hat{\mu}_{kj}$, $k = 1, \ldots, K$, are zero, hence all $\hat{x}_{kj}$ are equal to $\bar{x}_j$, then the jth gene will not contribute to the discriminant score, and it can be removed.

The NSC classifier is very easy to implement and very easy to interpret; it has been shown to be very successful in many high-dimensional classification problems. However, it has drawbacks.

In this article, we re-derive the NSC method as a LASSO regression on gene expression profiles. This re-interpretation allows us to notice that the L_1-norm penalty used by NSC may not be the most effective way in analyzing microarray data. The L_1-norm penalty treats all centroids equally or 'flatly', but centroids for the same gene are naturally 'grouped' together and intuitively should be considered as a group. Also, centroids for different genes, say relevant ones and irrelevant ones, should be treated differently. Enlightened by these observations, we consider two different penalty functions different from the L_1-norm penalty to make use of natural grouping information within the data. As we will see in the numerical study, the methods we consider tend to remove irrelevant genes more effectively and provide better classification results.

The remainder of this article is organized as follows. In Section 2, we show how the NSC can be represented as a LASSO regression, and based on that, we consider two new methods: the adaptive L_∞-norm penalized NSC (ALP-NSC) and the adaptive hierarchically penalized NSC (AHP-NSC), which improve over the NSC. In Section 3, we derive algorithms for the two methods in detail. Numerical results are in Sections 4 and 5. A brief discussion is in Section 6.

2 METHODS

2.1 LASSO interpretation of the nearest shrunken centroids

Assume that the observation $x_i = (x_{i1}, \ldots, x_{ip})$ from class k follows a multivariate normal distribution: MVN(ν_k, Σ_k), where $\nu_k = (\nu_{k1}, \ldots, \nu_{kj}, \ldots, \nu_{kp})$ is the mean vector for class k, and Σ_k is the covariance matrix for class k. We further assume that $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_j^2, \ldots, \sigma_p^2)$, i.e. the covariance matrices are the same across different classes and are diagonal. Such assumption is common when one works with high-dimension low sample size data (Marron and Todd, 2002). Some theoretical justification for this assumption can be found in Bickel and Levina (2004).

We first center and scale each x_{ij} to be $y_{ij} = (x_{ij} - \bar{x}_j)/(m_k \cdot s_j)$, and consider the linear regression:

$$y_{ij} = \sum_{k=1}^{K} z_{ik}\mu_{kj} + \epsilon_{ij}, \tag{7}$$

where $\mu_{kj} = (\nu_{kj} - \bar{x}_j)/(m_k.s_j)$; z_{ik} is the indicator for whether the ith sample is in class k, i.e. $z_{ik} = 1$ if the ith sample belongs to class k, and $z_{ik} = 0$ otherwise; ϵ_{ij} are independent of each other and approximately follow $N(0, 1/m_k^2)$, if sample i belongs to class k.

Now we consider an LASSO-(Tibshirani, 1996) type estimator for μ_{kj}:

$$\hat{\mu}_{kj} = \arg\min_{\mu_{kj}} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{k=1}^{K}\frac{z_{ik}}{n_k}(y_{ij} - \mu_{kj})^2 + \lambda\sum_{j=1}^{p}\sum_{k=1}^{K}|\mu_{kj}|. \tag{8}$$

After some algebra, one can show that the solutions to (8) are

$$\hat{\mu}_{kj} = \text{sgn}\left(\sum_i z_{ik}y_{ij}\right)\left(\left|\frac{\sum_i z_{ik}y_{ij}}{\sum_i z_{ik}}\right| - \lambda\right)_+ \tag{9}$$

$$= \text{sgn}(\mu_{kj}^0)(|\mu_{kj}^0| - \lambda)_+, \quad k = 1, \ldots, K;\ j = 1, \ldots, p \tag{10}$$

which matches exactly with (3). Therefore, the shrunken centroids used in the NSC can be considered as the solutions to (8). We acknowledge that Wu (2006) presented a similar interpretation of NSC, but used different values of λ for different classes.

We can see from (8) that by using the L_1-norm penalty, the NSC shrinks μ_{kj} continuously towards zero, and shrinks some of the fitted μ_{kj} to be *exactly* zero when making λ sufficiently large. In order to remove the jth gene, we require all μ_{kj}, $k = 1, \ldots, K$, to be zero. However, we can also see from (8) that the L_1-norm penalty treats all μ_{kj} the same, i.e. it does not use the information that μ_{kj} and $\mu_{k'j}$ are associated with the same gene j. Intuitively, they belong to one 'group' and should be treated differently from $\mu_{kj'}$, which is associated with a different gene j'. In the next two subsections, we consider two different penalty functions, i.e. the L_∞-norm penalty and the hierarchical penalty that incorporate this information into the modeling procedure. In general, we consider

$$\hat{\mu}_{kj} = \arg\min_{\mu_{kj}} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{K}\frac{z_{ik}}{n_k}(y_{ij} - \mu_{kj})^2 + \lambda \cdot J(\Omega)$$

$$= \arg\min_{\mu_{kj}} \ell(\Omega) + \lambda \cdot J(\Omega), \tag{11}$$

where $\Omega = \{\mu_{kj}, k = 1, \ldots, K;\ j = 1, \ldots, p\}$, and $J(\Omega)$ is a penalty function.

2.2 Method I: the adaptive L_∞-norm penalized NSC (ALP-NSC)

For the ALP-NSC, we consider to estimate μ_{kj} by

$$\hat{\mu}_{kj} = \arg\min_{\mu_{kj}}\left(\ell(\Omega) + \lambda\sum_{j=1}^{p}\max_k(|\mu_{1j}|, \ldots, |\mu_{kj}|, \ldots, |\mu_{Kj}|)\right), \tag{12}$$

where $\max(|\mu_{1j}|, \ldots, |\mu_{Kj}|) = \|(\mu_{1j}, \ldots, \mu_{Kj})\|_\infty$. Different from penalizing every μ_{kj} individually, the L_∞-norm penalizes the maximum absolute value of μ_{kj}, $k = 1, \ldots, K$, for the jth gene. If the maximum of $|\mu_{kj}|$, $k = 1, \ldots, K$, is shrunken to zero, all μ_{kj} are automatically shrunken to zero. The L_∞-norm penalty has also been used in Zhang *et al.* (2006b), Zhao *et al.* (2006) and Zou and Yuan (2006) for other supervised problems.

To further improve the model (12), we borrow the adaptive idea from Shen and Ye (2002) and Zou (2006), i.e. to penalize different genes differently. We consider

$$\min_{\mu_{kj}}\left(\ell(\Omega) + \lambda\sum_{j=1}^{p} w_j \cdot \max_k(|\mu_{1j}|, \ldots, |\mu_{kj}|, \ldots, |\mu_{Kj}|)\right), \tag{13}$$

where w_j are pre-specified weights. The intuition is that if the jth gene is relevant for distinguishing different classes from each other, we would like the corresponding w_j to be small, hence the jth gene is lightly penalized, while if the jth gene is irrelevant and expressed similarly

across different classes, we would like the corresponding w_j to be large, hence the jth gene is heavily penalized. How to pre-specify w_j from the data will be discussed in Section 2.4.

2.3 Method II: the adaptive hierarchically penalized NSC (AHP-NSC)

The L_∞-norm penalty makes use of the information that μ_{kj} and $\mu_{k'j}$ are associated with the same gene by shrinking the maximum absolute value of μ_{kj} within the jth gene. If we denote $M_j = \max_k(|\mu_{1j}|, \ldots, |\mu_{Kj}|)$, the corresponding L_∞-norm penalty on the jth gene is λM_j, and we can write $\mu_{kj} = M_j \alpha_{kj}$, where $-1 \le \alpha_{kj} \le 1$. This motivates us to reparameterize μ_{kj} in a more general way:

$$\mu_{kj} = \gamma_j \theta_{kj}, \quad j = 1, \ldots, p; \; k = 1, \ldots, K, \tag{14}$$

where $\gamma_j \ge 0$ (for identifiability reasons). Notice that here γ_j plays a similar role as M_j, but it does not have to be the maximum of $|\mu_{kj}|$; similarly θ_{kj} does not have to be bounded between -1 and 1. This decomposition reflects the information that $\mu_{kj}, k = 1, \ldots, K$, all belong to one single gene x_j, by treating each μ_{kj} hierarchically. γ_j is at the first level of the hierarchy, controlling $\mu_{kj}, k = 1, \ldots, K$, as a group; θ_{kj}s are at the second level of the hierarchy, reflecting differences within the jth gene.

To estimate γ_j and θ_{kj}, we consider

$$\min_{\gamma_j, \theta_{kj}} \left(\ell(\Omega) + \lambda_\gamma \sum_{j=1}^{p} \gamma_j + \lambda_\theta \sum_{j=1}^{p} \sum_{k=1}^{K} |\theta_{kj}| \right), \tag{15}$$

subject to $\gamma_j \ge 0$. Notice that there are two tuning parameters λ_γ and λ_θ. λ_γ controls the estimates at the gene-specific level, and it can effectively remove irrelevant genes: if γ_j is shrunken to zero, all μ_{kj} for the jth gene will be equal to zero. λ_θ controls the estimates at the class-specific level: if γ_j is not equal to zero, some of the θ_{kj} hence some of the $\mu_{kj}, k = 1, \ldots, K$, still have the possibility of being zero; in this sense, the hierarchical penalty shares some of the properties of the L_1-norm penalty.

The adaptive idea in (13) also applies here. If the jth gene is relevant, we would like to penalize its γ_j and θ_{kj} lightly, and if the jth gene is irrelevant, we would like to penalize its γ_j and θ_{kj} heavily. Hence, we consider the AHP-NSC:

$$\min_{\gamma_j \ge 0, \theta_{kj}} \left(\ell(\Omega) + \lambda_\gamma \sum_{j=1}^{p} w_j^\gamma \gamma_j + \lambda_\theta \sum_{j=1}^{p} \sum_{k=1}^{K} w_{kj}^\theta |\theta_{kj}| \right), \tag{16}$$

where w_j^γ and w_{kj}^θ are pre-specified weights.

2.4 Computing the adaptive weights

Regarding the adaptive weights w_j in (13), w_j^γ and w_{kj}^θ in (16), following Breiman (1995) and Zou (2006), we can choose them using the un-penalized estimates μ_{kj}^0. Specifically:

$$\begin{aligned} M_j^0 &= \max_k(|\mu_{1j}^0|, \ldots, |\mu_{Kj}^0|), \\ w_j &= 1/M_j^0, \\ w_j^\gamma &= 1/M_j^0, \\ w_{kj}^\theta &= 1/|\mu_{kj}^0|. \end{aligned}$$

3 ALGORITHMS

In this section, we describe details of our algorithms for estimating μ_{kj} in ALP-NSC and AHP-NSC. Notice that both the L_∞-norm penalty and the hierarchical penalty have non-differential points, so they pose optimization challenges.

3.1 Estimating μ_{kj} in ALP-NSC

In ALP-NSC, (13) can be decomposed into p separate minimization problems

$$\min_{\mu_{kj}} \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{K} \frac{z_{ik}}{n_k} (y_{ij} - \mu_{kj})^2 + \lambda \cdot w_j \cdot \max(|\mu_{1j}|, \ldots, |\mu_{Kj}|), \tag{17}$$

where $j = 1, \ldots, p$. For each j, (17) can be transformed into a quadratic programming problem:

$$\min_{\mu_{kj}, M_j} \quad \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{K} \frac{z_{ik}}{n_k} (y_{ij} - \mu_{kj})^2 + \lambda \cdot w_j \cdot M_j \tag{18}$$

$$\text{subject to} \quad -M_j \le \mu_{kj} \le M_j, \quad k = 1, \ldots, K \tag{19}$$

$$M_j \ge 0 \tag{20}$$

Hence, most commercially available packages can be used to solve it.

We have also explored explicit forms for the solutions to (17), which help us gain more insights into the nature of the L_∞-norm penalty. We can show that $\hat{\mu}_{kj}$, the solution to the minimization problem (17), can be achieved by shrinking an average of μ_{kj}^0 (1), i.e. the solution to (17) when there is no penalty (or $\lambda = 0$).

THEOREM 1. *For the jth minimization problem (17), if there exists an indices set $C = \{k_1, \ldots, k_r\}$, such that*

$$|\hat{\mu}_{k_1 j}| = \cdots = |\hat{\mu}_{k_r j}| > |\hat{\mu}_{kj}|, \text{ for } k \notin C \tag{21}$$

then

$$\hat{\mu}_{kj} = \begin{cases} \mu_{kj}^0 & k \notin C \\ \operatorname{sgn}(\mu_{kj}^0)\left(\frac{1}{r}\sum_{s=1}^{r} \left|-\mu_{k_s j}^0\right| - \frac{\lambda w_j}{r}\right)_+ & k \in C \end{cases} \tag{22}$$

where $(:)_+$ is the positive part of the argument.

Details of the proof are in the Supplementary Material. From Theorem 1, we can see when there are r maximums among $|\hat{\mu}_{kj}|$, only the corresponding μ_{kj}^0 will be shrunken by the L_∞-norm penalty, and they are shrunken to the same absolute value. This value is based on an average of μ_{kj}^0 of the corresponding r classes. We can also see that if the jth gene is irrelevant and all $|\mu_{kj}^0|$ are close to zero, then the L_∞-norm penalty tends to shrink all of them to zero (with an appropriately chosen λw_j).

To implement Theorem 1, we need to decide r, the number of maximums among $\hat{\mu}_{kj}$, and the set $\{k_1, \ldots, k_r\}$, which indicates which r μ_{kj}^0 should be shrunken. When K is not very large, say $K \le 20$, we can use an exhaustive search to find r and $\{k_1, \ldots, k_r\}$, i.e. for each $1 \le r \le K$, we search over all possible sets $\{k_1, \ldots, k_r\}$. For each possible set, we estimate $\hat{\mu}_{kj}$ using (22), then check whether the estimates satisfy the assumption (21). If the assumption is satisfied, we compute the corresponding value for the objective function (17). Finally, we choose

$\hat{\mu}_{kj}$ that give the smallest value for the objective function. When K is large, we will resort to the quadratic programming (18)–(20).

3.2 Estimating μ_{kj} in AHP-NSC

In AHP-NSC (16), we can use an iterative approach to estimate γ_j and θ_{kj}, i.e. we first fix θ_{kj} and estimate γ_j, then we fix γ_j and estimate θ_{kj}, and we iterate between these two steps until the solution converges. Since at each step, the value of the objective function (16) decreases, the solution is guaranteed to converge. We have the following theorem that helps us solve for γ_j and θ_{kj} at each step.

THEOREM 2.

- *When θ_{kj}, $j = 1,\ldots,p$ and $k = 1,\ldots,K$, are fixed,*

$$\hat{\gamma}_j = \mathrm{I}_{(\exists k,\, \theta_{kj}\neq 0)} \cdot \left(\sum_{k=1}^{K} \xi_k \frac{\mu^0_{kj}}{\theta_{kj}} - \frac{\lambda_\gamma w_j^\gamma}{\sum_{k=1}^{K} \theta^2_{kj}} \right)_+, \tag{23}$$

where $\xi_k = \theta^2_{kj} / \sum_{k=1}^{K} \theta^2_{kj}$.

- *When γ_j, $j = 1,\ldots,p$, are fixed,*

$$\hat{\theta}_{kj} = \mathrm{I}_{(\gamma_j>0)} \cdot \mathrm{sgn}(\mu^0_{kj}) \left(\frac{|\mu^0_{kj}|}{\gamma_j} - \frac{\lambda_\theta w^\theta_{kj}}{\gamma_j^2} \right)_+ \tag{24}$$

Equations (23) and (24) show that both $\hat{\gamma}_j$ and $\hat{\theta}_{kj}$ are soft-thresholding estimates. Details of the proof are in the Supplementary Material. Here we give some intuitive explanation.

We first look at $\hat{\gamma}_j$ (23). If all θ_{kj} are zero, it is natural to estimate γ_j also to be zero because of the penalty on γ_j. If not all θ_{kj} are 0, say, $\theta_{k_1 j},\ldots,\theta_{k_r j}$ are not zero, then from our reparameterization, we have $\gamma_j = \mu_{k_s j}/\theta_{k_s j}$, $1 \le s \le r$. Plugging in $\mu^0_{k_s j}$ for $\mu_{k_s j}$, we obtain r estimates for γ_j: $\tilde{\gamma}_j = \mu^0_{k_s j}/\theta_{k_s j}$, $1 \le s \le r$. A natural estimate for γ_j is then a weighted average of the $\tilde{\gamma}_j$, and Equation (23) provides such a (shrunken) average, with weights proportional to ξ_k.

Now considering $\hat{\theta}_{kj}$ (24). If $\gamma = 0$, it is natural to estimate all θ_{kj} also to be zero because of the penalty on θ_{kj}. When $\gamma_j > 0$, we have $\theta_{kj} = \mu_{kj}/\gamma_j$. Again, plugging in μ^0_{kj} for μ_{kj}, we obtain $\tilde{\theta}_{kj} = \mu^0_{kj}/\gamma_j$. Equation (24) shrinks $\tilde{\theta}_{kj}$ and the amount of shrinkage is inversely proportional to γ_j^2. When γ_j is large, which indicates the jth gene is relevant, the amount of shrinkage is small, while when γ_j is small, which indicates the jth gene is less relevant, the amount of shrinkage is large.

4 SIMULATION STUDY

In this section, we use simulated data to demonstrate our methods ALP-NSC and AHP-NSC, and compare the results with that of the NSC. We also compare our methods with an improved version of NSC, i.e. NSC with adaptive choice of thresholds (Tibshirani *et al.*, 2003). The 'adaptive' in Tibshirani *et al.* (2003) has a different meaning from that in our two methods: it still treats different genes 'flatly', but uses large thresholds for classes easy to classify and small thresholds for classes difficult to classify.

We first considered a two-class classification scenario. There were a total of $p = 10\,000$ variables with only the first 20 relevant while the other 9980 irrelevant in forming two classes. Specifically, the first 20 variables were i.i.d. $N(0, 1)$ for the first class and i.i.d. $N(1, 1)$ for the second class, whereas the remaining 9980 variables were all i.i.d. $N(0, 1)$ for both classes.

We generated $n = 100$ training observations, with 70 in the first class and 30 in the second one; similarly, we also generated 1000 test observations, with the same class prior and the same within-class distribution. We denote this as the '70-30' example. Tuning parameters were chosen using 5-fold cross-validation (CV) on the training data. We then computed the misclassification error rate of the chosen model on the test data. We also recorded both the number of relevant variables and the number of irrelevant variables that were selected. We repeated this 100 times. The results are summarized in Table 1.

As we can see, our ALP-NSC and AHP-NSC methods performed similarly to the NSC method in terms of selecting relevant variables, but tended to keep much fewer irrelevant variables. The error rates of ALP-NSC and AHP-NSC also seemed to be smaller than that of the NSC.

We then considered two three-class classification scenarios, with highly unbalanced classes. In both scenarios, there were a total of $p = 4000$ variables with the first 2 relevant while the other 3998 irrelevant in forming three classes. The first 2 variables were i.i.d. from $N(0, 1)$ for the first class, i.i.d. $N(2.5, 1)$ for the second class and i.i.d. $N(5, 1)$ for the third class, whereas the remaining 3998 variables were all i.i.d. from $N(0, 1)$ for all three classes. In the first scenario, we generated 20 observations for each of the first class and the third class, and 100 for the second class We denote it as the '20-100-20' example. In the second scenario, we generated 100 observations for each of the first class and the third class, and 20 for the second class. We denote it as the '100-20-100' example. For each scenario, we also generated test observations with the same class prior and the same within-class distribution, but with the sample size 10 times larger than that of the training datasets. We repeated this 100 times. The results are summarized in Table 2.

Table 1. Simulation results for the '70-30' example

Method	# Info	# Non-info	CV error	Test error
NSC-No-Noise	–	–	–	0.03 (0.01)
NSC	19 (1)	90 (89)	0.10 (0.03)	0.09 (0.02)
NSC-Ada	19 (1)	90 (89)	0.10 (0.03)	0.09 (0.02)
ALP-NSC	18 (2)	39 (38)	0.04 (0.02)	0.04 (0.01)
AHP-NSC	16 (3)	10 (8)	0.00 (0.01)	0.04 (0.02)

'# Info' is the average number of selected relevant variables (out of 20) over 100 repetitions. '# Non-info' is the average number of irrelevant variables (out of 9980) that were kept. 'CV error' is the average misclassification error rate in 5-fold cross-validation. 'Test error' is the average misclassification error rate on test data. The numbers in the parentheses are the corresponding SDs. 'NSC-No-Noise' is to apply the NSC method to the dataset with only the first 20 relevant variables, and its 'Test Error' can be considered as an 'oracle' benchmark. 'NSC-Ada' refers to the NSC with adaptive thresholds.

Table 2. Simulation results for '20-100-20' and '100-20-100' examples: the upper part is for the '20-100-20' example, and the lower part is for the '100-20-100' example

Method	# Info	# Non-info	CV error	Test error
'20-100-20'				
NSC-No-Noise	–	–	–	0.04 (0.01)
NSC	2 (0)	41 (67)	0.17 (0.03)	0.17 (0.02)
NSC-Ada	2 (0)	41 (67)	0.17 (0.03)	0.17 (0.02)
ALP-NSC	2 (0)	0.1 (0.3)	0.07 (0.02)	0.07 (0.01)
AHP-NSC	2 (0)	0.0 (0.0)	0.04 (0.02)	0.05 (0.01)
'100-20-100'				
NSC-No-Noise	–	–	–	0.02 (0.00)
NSC	2 (0)	97 (148)	0.09 (0.00)	0.09 (0.00)
NSC-Ada	2 (0)	62 (120)	0.09 (0.00)	0.09 (0.00)
ALP-NSC	2 (0)	0.1 (0.2)	0.03 (0.01)	0.03 (0.01)
AHP-NSC	2 (0)	0.0 (0.0)	0.03 (0.01)	0.03 (0.00)

Descriptions for the columns are the same as those in the caption of Table 1.

As we can see, our ALP-NSC and AHP-NSC methods removed all 3998 irrelevant variables for every repetition (out of 100), and the classification error rates were just slightly higher than that of the NSC using only the first 2 relevant variables, i.e., the 'oracle'. In contrast, the NSC method and the NSC with adaptive thresholds tended to select more noise variables and have higher error rates.

5 REAL DATA ANALYSIS

In this section, we apply the ALP-NSC and the AHP-NSC methods to three gene microarray datasets.

The first dataset we considered is the Leukemia dataset in Golub *et al.* (1999). This dataset consists of 38 training data and 34 test data for two types of acute leukemia, acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). Each sample is a vector of $p = 7129$ genes. We applied the ALP-NSC and the AHP-NSC methods to the training data. Tuning parameters were chosen using 10-fold CV, and the chosen models were evaluated on the test data. The results are summarized in the upper part of Table 3. Both the ALP-NSC and the AHP-NSC had two misclassification on the test data, and they selected 16 exactly same genes. Figure 1 shows the corresponding heatmap. Clear separation of the two classes is evident.

The second dataset we considered consists of microarray experiments of small round blue cell tumors (SRBCT) of childhood cancer (Khan *et al.*, 2001). The tumors are classified as Burkitt lymphoma (BL), Ewing sarcoma (EWS), neuroblastoma (NB) or rhabdomyosarcoma (RMS). A total of 63 training samples and 20 test samples were provided. Each sample consists of expression measurements on $p = 2308$ genes. We analyzed this dataset in the similar way as with the Leukemia dataset. Tuning parameters were chosen using 8-fold CV. The results are summarized in Table 3. The CV errors and the test errors are all zero. The ALP-NSC selected 38 genes, and the AHP-NSC selected 40 genes. The 38 genes selected by ALP-NSC and the 40 genes selected by AHP-NSC have

Table 3. Results on the real datasets: the upper part is for the Leukemia dataset, and the lower part is for the SRBCT dataset

Method	Number of genes	Number of CV errors	Number of test errors
The Leukemia dataset			
Golub *et al.* (1999)	50	3/38	4/34
NSC	21	1/38	2/34
ALP-NSC	16	1/38	2/34
AHP-NSC	16	1/38	2/34
The SRBCT dataset			
Kahn *et al.* (2001)	96	0/63	0/20
NSC	43	0/63	0/20
ALP-NSC	38	0/63	0/20
AHP-NSC	40	0/63	0/20

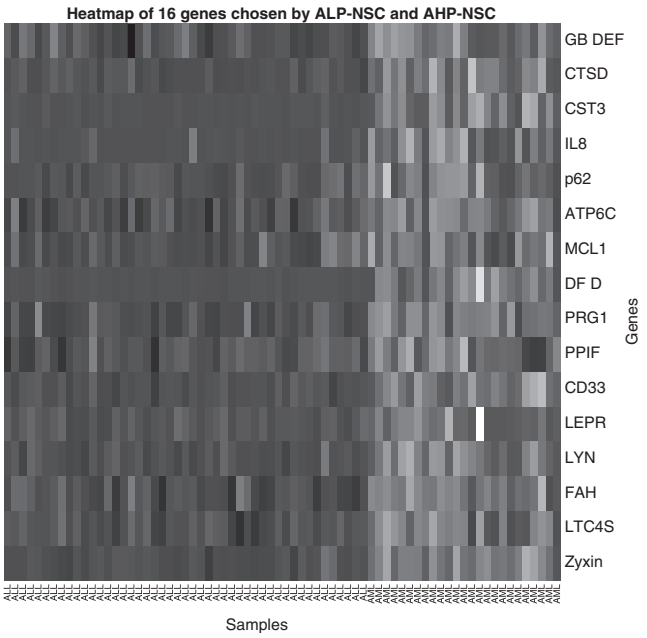

Fig. 1. Heatmap of the 16 genes selected by both the ALP-NSC and the AHP-NSC from the Leukemia dataset.

34 overlapping genes. Figures 2 and 3 show the heatmaps of the selected genes. Similar as in Figure 1, genes that distinguish each class from other classes are also evident.

To further assess the genes that were selected from the Leukemia and the SRBCT datasets, we randomly split each dataset into the training and the test sets for 100 times. The sizes of the two sets and the class priors were kept the same as the original training/test split. Each training/test split was also analyzed in the same way as for the original training/test split. The results are summarized in Tables 4–6. For the Leukemia dataset, out of the 16 genes selected from the original training/test split, 10 genes were selected for more than 70 times out of the 100 random training/test splits. For the SRBCT dataset, out of the 44 genes that were selected by ALP-NSC and AHP-NSC from the original training/test split, 29 genes were

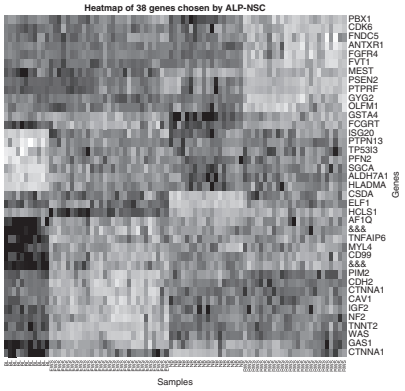

Fig. 2. Heatmap of the 38 genes selected by ALP-NSC from the SRBCT dataset.

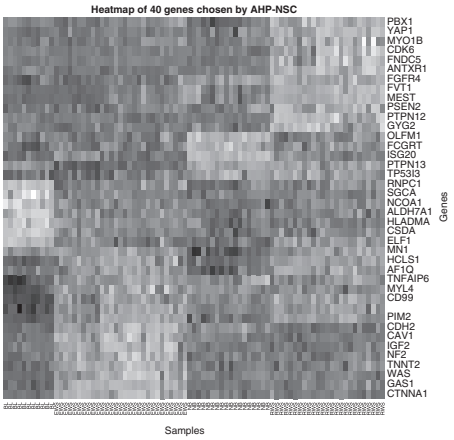

Fig. 3. Heatmap of the 40 genes selected by AHP-NSC from the SRBCT dataset.

selected for more than 70 times out of the 100 random splits. These results imply that the highly frequently selected genes may have certain power in differentiating the corresponding cancer classes from each other.

The Leukemia and the SRBCT datasets are relatively 'easy' for classification. We also considered the NCI-60 dataset (Dudoit *et al*., 2002). In this study, cDNA microarrays were used to examine the variation in gene expressions among 60 cell lines from the National Cancer Institute's anti-cancer drug screen. The cell lines are derived from tumors with eight different sites of origin: breast, central nervous system (CNS), colon, leukemia, melanoma, non-small-cell-lung-carcinoma (NSCLC), ovarian and prostate. A total of 61 samples were provided. Each sample consists of expression measurements on $p = 5244$ genes. The class sizes are all small, and some of the classes (e.g. breast and NSCLS) are known to be heterogeneous. So this is a more difficult dataset for classification than the Leukemia and the SRBCT datasets. The NCI-60 dataset came without pre-specified training/test split, so we randomly split the data into the training and the test sets comma; with sample sizes 40 and 21, respectively. We repeat it 100 times. The results are summarized in the lower part of Table 4 and Figure 4. We can see that ALP-NSC and AHP-NSC had similar misclassification errors as NSC and adaptive-NSC, but selected much fewer genes. Then, 777 genes were selected for more than 70 times out of the 100 trials. Our methods selected on average about 1000 genes, while NSC selected more than 3000 genes.

Table 4. Results on 100 random splits of the real datasets: the first part is for the Leukemia dataset, the second part is for the SRBCT dataset and the third part is for the NCI-60 dataset

Method	Number of genes	Number of CV errors	Number of test errors
The Leukemia dataset			
NSC	146 (320)	1.4 (1.1)	2.3 (1.4)
NSC-Ada	183 (322)	1.4 (1.1)	2.2 (1.5)
ALP-NSC	94 (102)	1.6 (1.2)	2.0 (1.1)
AHP-NSC	57 (43)	1.9 (1.2)	2.5 (1.2)
The SRBCT dataset			
NSC	47 (14)	0.0 (0.0)	0.3 (0.6)
NSC-Ada	49 (15)	0.0 (0.0)	0.3 (0.6)
ALP-NSC	36 (14)	0.2 (0.5)	0.1 (0.3)
AHP-NSC	34 (7)	0.2 (0.4)	0.1 (0.2)
The NCI-60 dataset			
NSC	3723 (1741)	11.9 (2.3)	6.3 (1.6)
NSC-Ada	3597 (1838)	11.9 (2.3)	6.2 (1.5)
ALP-NSC	1314 (128)	13.5 (2.7)	5.9 (1.3)
AHP-NSC	995 (131)	12.8 (2.5)	5.5 (1.3)

The errors are the averages of the number of misclassified samples over 100 random splits of the corresponding dataset. The numbers in the arentheses are the corresponding SDs.

6 CONCLUSION

In this article, we first re-interpreted the popular NSC method in the framework of LASSO regression. Based on the penalized linear regression framework, we have considered two new methods for microarray classification, which improve over the NSC. Unlike the L_1-norm penalty used in LASSO, the penalty terms that we consider make use of the fact that parameters belonging to one gene should be treated as a natural group. We have presented some evidence that the two new methods tend to remove irrelevant genes more effectively and provide better classification results than the L_1-norm approach.

ACKNOWLEDGEMENT

We would like to thank the Associate Editor and two reviewers for their thoughtful and useful comments. We would also like to thank Trevor Hastie and Rob Tibshirani for helpful discussions. S.W and J.Z are partially supported by grant DMS-0505432 from the National Science Foundation and grant NHLBI-18 HL60884 from National Institutes of Health.

Conflict of Interest: none declared.

Table 5. The 16 genes that were selected by the ALP-NSC and the AHP-NSC from the Leukemia dataset

Gene accession number	Selection frequency	Gene description
X95735_at	100	Zyxin
M23197_at	98	CD33 CD33 antigen (differentiation antigen)
M63138_at	98	CTSD Cathepsin D (lysosomal aspartyl protease)
M27891_at	98	CST3 Cystatin C (amyloid angiopathy and cerebral hemorrhage)
M16038_at	97	LYN V-yes-1 Yamaguchi sarcoma viral-related oncogene homolog
X17042_at	91	PRG1 Proteoglycan 1, secretory granule
M84526_at	90	DF D component of complement (adipsin)
M55150_at	85	FAH Fumarylacetoacetate
U50136_rna1_at	80	Leukotriene C4 synthase (LTC4S) gene
M62762_at	75	ATP6C Vacuolar H+ ATPase proton channel subunit
Y00787_s_at	53	INTERLEUKIN-8 PRECURSOR
U82759_at	51	GB DEF = Homeodomain protein HoxA9 mRNA
U46751_at	35	Phosphotyrosine independent ligand p62 for the Lck SH2 domain mRNA
M80254_at	29	PEPTIDYL-PROLYL CIS-TRANS ISOMERASE, MITOCHONDRIAL PRECURSOR
L08246_at	27	INDUCED MYELOID LEUKEMIA CELL DIFFERENTIATION PROTEIN MCL1
Y12670_at	25	LEPR Leptin receptor

'Selection frequency' is the number of times that the corresponding gene was selected out of 100 random splits of the training and testing data.

Table 6. The 44 genes that were selected by the ALP-NSC and the AHP-NSC from the SRBCT dataset

Unique gene ID	Gene symbol	Selection frequency	Unique gene ID	Gene symbol	Selection frequency	Unique gene ID	Gene symbol	Selection frequency
Hs.2157	WAS	100	Hs.356717	MYL4	99	Hs.408222	PBX1	62
Hs.349109	IGF2	100	Hs.221889	CSDA	99	Hs.65029	GAS1	46
Hs.74034	CAV1	100	Hs.74376	OLFM1	99	Hs.351279	HLA-DMA	43
Hs.80205	PIM2	100	Hs.274520	ANTXR1	98	Hs.236361	RNPC1	38
Hs.251664	–	100	Hs.254321	CTNNA1	97	Hs.169907	GSTA4	37
Hs.283477	CD99	100	Hs.902	NF2	97	Hs.312102	ALDH7A1	32
Hs.75823	AF1Q	100	Hs.124030	ELF1	96	Hs.268515	MN1	22
Hs.99931	SGCA	100	Hs.170548	YAP1	94	Hs.75216	PTPRF	21
Hs.387553	PTPN13	100	Hs.407132	TNNT2	93	In multiple clusters	–	20
Hs.105434	ISG20	100	Hs.50649	TP53I3	88	Hs.121576	MYO1B	13
Hs.111903	FCGRT	100	Hs.407546	TNFAIP6	83	Hs.386092	NCOA1	5
Hs.74050	FVT1	100	Hs.14601	HCLS1	80	Hs.25363	PSEN2	2
Hs.165950	FGFR4	100	Hs.380757	GYG2	77	Hs.91747	PFN2	2
Hs.15463	FNDC5	100	Hs.440459	MEST	76	Hs.62	PTPN12	2
Hs.388761	CDK6	100	Hs.334131	CDH2	62	–	–	–

'Selection frequency' is the number of times that the corresponding gene was selected out of 100 random splits of the training and testing data.

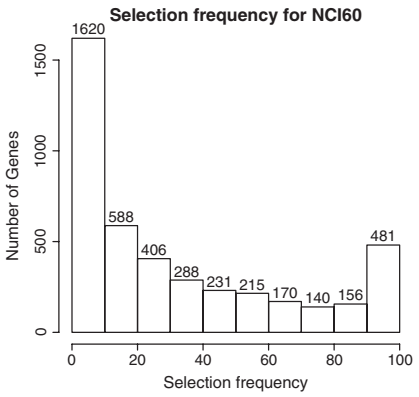

Fig. 4. Selection frequency for ALP-NSC and AHP-NSC out of 100 trials for the NCI-60 dataset.

REFERENCES

Alizadeh,A. *et al*. (2000) Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, **403**, 503–511.

Bickel,P. and Levina,L. (2004) Some theory for fisher's linear discriminant function, "naive bayes", and some alternatives when there are many more variables than observations. *Bernoulli*, **10**, 989–1010.

Breiman,L. (1995) Better subset regression using the non-negative garrote. *Technometrics*, **37**, 373–384.

Dabney,A.R. (2005) Classification of microarrays to nearest centroids. *Bioinformatics*, **21**, 4148–4154.

Dudoit,S. *et al*. (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Am. Stat. Assoc*., **97**, 77–87.

Eisen,M. *et al*. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, **95**, 14863–14868.

Golub,T. *et al*. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.

Hastie,T. *et al*. (2001) Supervised harvesting of expression trees. *Genome Biol*., **2**, 1–12.

Khan,J. *et al*. (2001) Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat. Med*., **7**, 673–679.

Liu,Y. and Shen,X. (2006) Multicategory psi-learning. *J. Am. Stat. Assoc*., **101**, 500–509.

Marron,J. and Todd,M. (2002) Distance weighted discrimination. Technical Report. School of Operations Research and Industrial Engineering, Cornell University.

Pan,W. (2002) A comparative review of statistical methods for discovering differently expressed genes in replicated microarray experiments. *Bioinformatics*, **18**, 546–554.

Shen,X. and Ye,J. (2002) Adaptive model selection. *J. Am. Stat. Assoc*., **97**, 210–221.

Shen,R. *et al*. (2006) Eigengene-based linear discriminant model for tumor classification using gene expression microarray data. *Bioinformatics*, **22**, 2635–2642.

Tibshirani,R. (1996) Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. B*., **58**, 267–288.

Tibshirani,R. *et al*. (2002) Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Natl Acad. Sci. USA*, **99**, 6567–6572.

Tibshirani,R. *et al*. (2003) Class prediction by nearest shrunken centroids, with application to DNA microarrays. *Stat. Sci*., **18**, 104–117.

Wu,B. (2006) Differential gene expression detection and sample classification using penalized linear regression models. *Bioinformatics*, **22**, 472–476.

Zhang,H. *et al*. (2006a) Gene selection using support vector machines with non-convex penalty. *Bioinformatics*, **22**, 88–95.

Zhang,H. *et al*. (2006b) Variable selection for multicategory SVM via sup-norm regularization. Institute of Statistics Mimeo Series 2596, NCSU.

Zhao,P. *et al*. (2006) Grouped and hierarchical model selection through composite absolute penalties. Technical Report. Department of Statistics, University of California at Berkeley.

Zou,H. (2006) The adaptive lasso and its oracle properties. *J. Am. Stat. Assoc*., **101**, 1418–1429.

Zou,H. and Yuan,M. (2007) The F_∞-norm support vector machine. *Stat. Sin*., to appear.