BIOINFORMATICS ORIGINAL PAPER

Vol. 22 no. 1 2006, pages 88–95
doi:10.1093/bioinformatics/bti736

Gene expression

Gene selection using support vector machines with non-convex penalty

Hao Helen Zhang[1,*], Jeongyoun Ahn[2], Xiaodong Lin[3] and Cheolwoo Park[4]

[1]Department of Statistics, North Carolina State University, Raleigh, NC 27695, USA, [2]Department of Statistics and Operations Research, University of North Carolina, Chapel Hill, NC 27599, USA, [3]Department of Mathematical Sciences, University of Cincinnati, OH 45221, USA and [4]Department of Statistics, University of Georgia, Athens, GA 30602, USA

Received on June 12, 2005; revised and accepted on October 20, 2005
Advance Access publication October 25, 2005
Associate Editor: John Quackenbush

ABSTRACT

Motivation: With the development of DNA microarray technology, scientists can now measure the expression levels of thousands of genes simultaneously in one single experiment. One current difficulty in interpreting microarray data comes from their innate nature of 'high-dimensional low sample size'. Therefore, robust and accurate gene selection methods are required to identify differentially expressed group of genes across different samples, e.g. between cancerous and normal cells. Successful gene selection will help to classify different cancer types, lead to a better understanding of genetic signatures in cancers and improve treatment strategies. Although gene selection and cancer classification are two closely related problems, most existing approaches handle them separately by selecting genes prior to classification. We provide a unified procedure for simultaneous gene selection and cancer classification, achieving high accuracy in both aspects.

Results: In this paper we develop a novel type of regularization in support vector machines (SVMs) to identify important genes for cancer classification. A special nonconvex penalty, called the smoothly clipped absolute deviation penalty, is imposed on the hinge loss function in the SVM. By systematically thresholding small estimates to zeros, the new procedure eliminates redundant genes automatically and yields a compact and accurate classifier. A successive quadratic algorithm is proposed to convert the non-differentiable and non-convex optimization problem into easily solved linear equation systems. The method is applied to two real datasets and has produced very promising results.

Availability: MATLAB codes are available upon request from the authors.

Contact: hzhang@stat.ncsu.edu

Supplementary information: http://www4.stat.ncsu.edu/~hzhang/research.html

1 INTRODUCTION

We consider the problem of gene selection for cancer classification using microarray gene expression data. The objective of gene selection is 2-fold: to provide a better understanding of the underlying biological system that generates data and to improve the prediction performance of classifiers. Effective gene selection often leads to a compact classifier with better accuracy and interpretability (Kitter, 1986).

Gene selection is treated as a variable selection problem in statistics and a dimension reduction problem in machine learning. Many greedy algorithms have been developed in the literature. Gene-ranking methods are particularly popular, which select genes according to some predetermined ranking criteria. There are two main types of ranking criteria: correlation coefficients (Golub *et al*., 1999; Furey *et al*., 2000; Pavlidis *et al*., 2001) and hypothesis testing statistics. Two-sample *t*-test methods include parametric tests (Devore and Peck, 1997; Thomas *et al*., 2001; Pan, 2002) and non-parametric tests (Troyanskaya *et al*., 2002; He, 2004). Although being useful in practice, all these methods that select important genes based on individual gene information thus fail to take into account mutual information among genes. Dimension reduction techniques project the full data onto the first few principal directions then conduct classification in the low-dimensional subspace. West (2003) proposed the idea of 'meta-genes', which are linear combinations of the original genes. One disadvantage of projection methods is that none of the original genes can be discarded since each principal component generally involves all the genes.

Support vector machines (SVMs) (Boser *et al*., 1992; Vapnik, 1995; Cristianini and Shawe-Taylor, 1999) have demonstrated superior performances in classifying high-dimensional and low sample size data. However, the standard SVM can suffer from the presence of redundant variables (Hastie *et al*., 2001; Guyon *et al*., 2002), since its decision rule utilizes all the variables without discrimination. Several methods have been proposed for variable selection in the SVM (Furey *et al*., 2000; Rakotomamonjy, 2003; Grandvalet and Canu, 2002; Mukherjee *et al*., 2000; Chapelle *et al*., 2002; Weston *et al*., 2000). Guyon *et al*. (2002) developed the recursive feature elimination algorithm, which successively eliminates features by training a sequence of SVM classifiers. Bradley and Mangasarian (1998) suggested the L_1 SVM, which imposes the absolute value penalty on the directional vector of the separating plane.

Different from all the methods above, we formulate the SVM as a regularization problem with a novel form of the penalty. The optimization problem consists of two parts: the data fit is represented by the hinge loss function, and the regularization is defined as the smoothly clipped absolute deviation penalty. In the regression context, this penalty was proposed and studied by Fan and Li (2001) and shown to have better theoretical properties than the L_1 penalty. Following their terminology, we will refer our method as the SCAD

[*] To whom correspondence should be addressed.

SVM. The SCAD SVM conducts variable selection and classification simultaneously, resulting in a compact classifier with high accuracy. We give an iterative algorithm to solve the SCAD SVM, and show that only linear equation system solvers are needed for its implementation. The SCAD SVM is applicable to any biological data of high-dimensional low sample size. Its performances on one microarray dataset and on one metabolism dataset are illustrated in the paper.

2 METHODS

Given a training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i = (x_{i1}, \ldots x_{id} \in \mathbb{R}^d$ is the input vector and $y_i \in \{+1, -1\}$ indicates its class label, the classification problem is to learn a discrimination rule $f: \mathbb{R}^d \to \{+1, -1\}$ so that we can assign a class label to any new subject observed in the future. For microarray gene expression data, $\mathbf{x}_i$ represents the expression levels of d genes of the i-th sample tissue and y_i is 'normal' or 'cancerous'; often we have $d \gg n$. In the statistical framework, we assume $(\mathbf{x}_i, y_i)$s are independent realizations of the random pair $(\mathbf{X}, Y)$ which follows a joint distribution $P(\mathbf{X}, Y)$. Define $g(\mathbf{x}) = \mathrm{Prob}(Y = +1|\mathbf{X} = \mathbf{x})$. With the 0-1 loss

$$L[f(\mathbf{x}), y] = \begin{cases} 1 & \text{if } yf(\mathbf{x}) < 0 \\ 0 & \text{if } yf(\mathbf{x}) > 0, \end{cases}$$

the optimal rule minimizing the expected loss EL $[f(\mathbf{X}, Y)]$ is the Bayes rule sign $[g(\mathbf{x}) - 1/2]$. If $f(\mathbf{x}) = 0$, the point $\mathbf{x}$ is randomly classified as +1 or −1. The function sign(t) takes value 1 if $t > 0$ and takes value −1 if $t < 0$.

2.1 Support vector machines

SVM is a large margin classifier which separates two classes by maximizing the margin between them. For non-separable data, the soft-margin SVM uses the slack variable to control an upper bound of the misclassification error. For classifying the data with complicated structures where a linear separation is not plausible, the non-linear SVM maps the data from the original input space into a high-dimensional feature space and then implements the linear classification in the feature space. Lin (2002) showed that under some general conditions, the SVM solution approaches the Bayes rule when the sample size increases.

The SVM finds $f(\mathbf{x}) = b + \mathbf{w} \cdot h(\mathbf{x})$ by minimizing

$$\frac{1}{n}\sum_{i=1}^{n}[1 - y_i(b + \mathbf{w} \cdot h(\mathbf{x}_i))]_+ + \lambda\|w\|^2, \tag{1}$$

where b is constant, $\mathbf{w}$ is the directional vector and $\mathcal{D} = \{h_1(\mathbf{x}), \ldots, h_q(\mathbf{x})\}$ is a dictionary of basis functions. The parameter λ controls the trade-off between minimizing the loss function and maximizing the margin. The hinge loss $[1 - yf(\mathbf{x})]_+$ is a convex upper bound for the 0-1 loss. One nice property of the SVM is that its solution only depends on a small subset of the training set called 'support vectors'. The standard SVM cannot select important variables, since all the input variables are used for constructing the classifier. For variable selection purposes, the following thresholding functions can be used to replace the L_2 penalty $\|\mathbf{w}\|^2$:

$$L_0(\mathbf{w}) = \sum_{j=1}^{q} I(w_j \neq 0),$$

$$L_1(\mathbf{w}) = \sum_{j=1}^{q} |w_j|,$$

$$L_\gamma(\mathbf{w}) = \sum_{j=1}^{q} |w_j|^\gamma, \quad 0 < \gamma < 1.$$

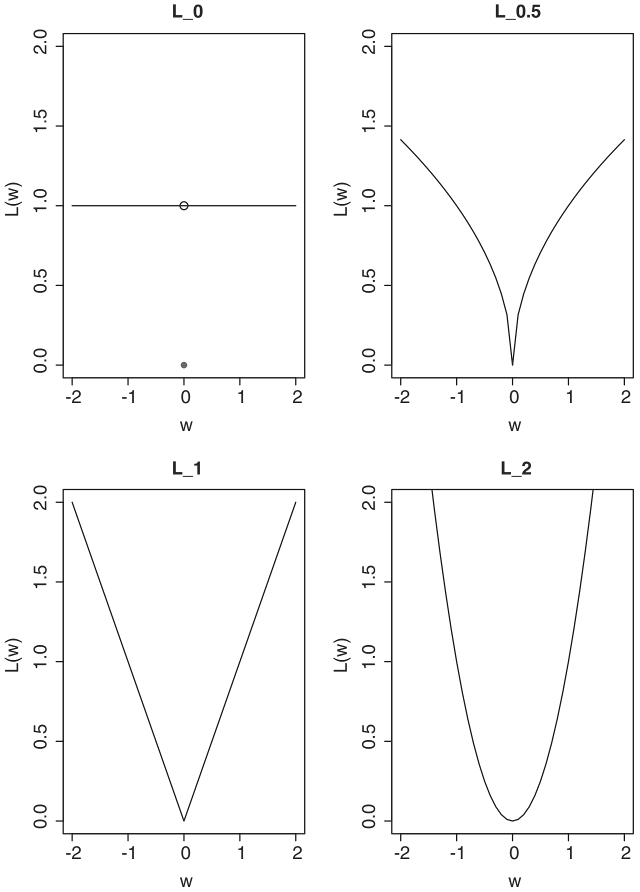


Fig. 1. Hard-thresholding penalty L_0; soft-thresholding penalties $L_{0.5}$, L_1 and L_2 penalty.

Figure 1 plots the L_0, $L_{0.5}$, L_1, and L_2 penalty functions. The L_0 penalty is the hard-thresholding penalty, which shrinks small coefficients to zero while keeping large coefficients intact. The discontinuity of the L_0 penalty makes the optimization problem hard and tends to produce unstable solutions, therefore soft-thresholding penalties are generally more preferred. In the context of wavelet shrinkage, Donoho and Johnstone (1994) proposed hard- and soft-thresholding methods for signal denoising, where the former leaves the magnitudes of coefficients unchanged if they are larger than a given threshold, while the latter shrinks them to zero by the threshold value. It is known that the L_γ function is a soft-thresholding penalty only if $\gamma \leq 1$ (Bradley and Mangasarian, 1998). This explains why the standard SVM corresponding to $\gamma = 2$ does not select variables. In Figure 1, the first three penalty functions are all non-differentiable at the origin, which is a necessary condition for a penalty function to produce sparse solutions (Fan and Li, 2001).

In the statistics literature, the L_1 penalty is also known as the LASSO (Tibshirani, 1996) and widely used for variable selection in linear regression models. The L_1 SVM was proposed by Bradley and Mangasarian (1998). Recently Zhu *et al*. (2003) studied its solution properties and suggested an algorithm to find the whole solution path over a range of tuning parameters. Fung and Mangasarian (2004) developed a fast Newton algorithm to solve its dual problem.

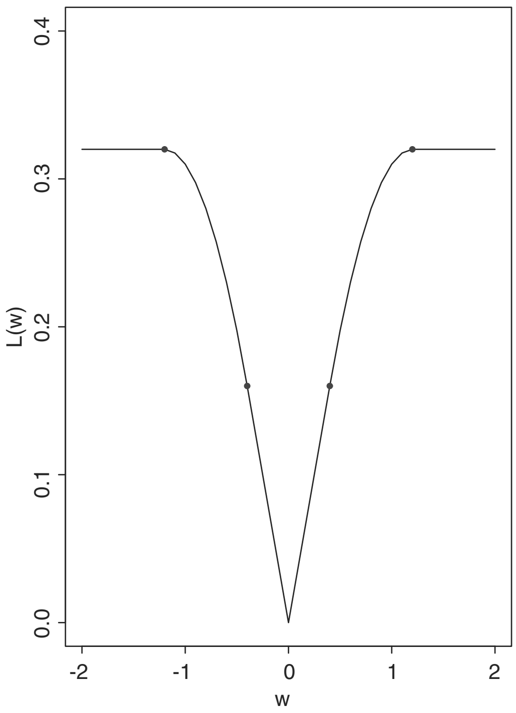


Fig. 2. The SCAD penalty function with $\lambda = 0.4$ and $a = 3$.

Other methods incorporating the model parsimony include some Bayesian methods (Lee *et al*., 2003; Bae and Mallick, 2004).

2.2 The SCAD SVM

Though the L_1 penalty gives sparse solutions, the estimates can be biased for large coefficients since larger penalties are imposed on larger coefficients. In linear regression models, Fan and Li (2001) proposed the smoothly clipped absolute deviation (SCAD) penalty which overcomes the biasness problem of the L_1 penalty. They showed that the SCAD penalty produces sparse solutions by thresholding small estimates to zero, provides nearly unbiased estimates for large coefficients and gives a model continuous in data. In this paper, we propose the SCAD SVM to conduct variable selection in the context of classification and study its performances on high-dimensional low sample size data.

The SCAD function, as plotted in Figure 2, is symmetric, non-convex and singular at the origin. Though having the same form as the L_1 penalty at the neighborhood of zero, the SCAD applies a constant penalty for large coefficients while the L_1 penalty increases linearly as the coefficient increases. It is this distinct feature that guards the SCAD penalty against producing biases for estimating large coefficients. Mathematically, the SCAD penalty has the expression

$$p_\lambda(|w|) = \begin{cases} \lambda|w| & \text{if } |w| \le \lambda, \\ -\frac{(|w|^2 - 2a\lambda|w| + \lambda^2)}{2(a-1)} & \text{if } \lambda < |w| \le a\lambda, \\ \frac{(a+1)\lambda^2}{2} & \text{if } |w| > a\lambda, \end{cases} \qquad (2)$$

where $a > 2$ and $\lambda > 0$ are tuning parameters. In Figure 2, we have $a = 3$ and $\lambda = 0.4$. The function in (2) is a quadratic spline function with two knots at λ and $a\lambda$. Except being singular at the origin, the function $p_\lambda(w)$ has a continuous first-order derivative. We propose the SCAD SVM as

$$\min_{b,\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n}[1 - y_i(b + \mathbf{w}\cdot h(\mathbf{x}_i))]_+ + \sum_{j=1}^{q} p_\lambda(|w_j|). \qquad (3)$$

The objective function in (3) consists of the hinge loss part and the SCAD penalty on $\mathbf{w}$. The parameter λ balances the trade-off between data fitting and model parsimony. If λ is too small, the procedure tends to overfit the training data and gives a classifier with little sparsity; if λ is too large, the produced classifier can be very sparse but have a poor discriminating power. To tune λ properly, we generate a tuning set for the simulated data and use cross validation for the real data. Fan and Li (2001) showed that the Bayes risks are not sensitive to the choice of a, and $a = 3.7$ is a good choice for various problems. We also use $a = 3.7$ in our examples.

Interestingly, when $d \gg n$, linear classifiers often give better performances than non-linear ones in many applications (Hastie *et al*., 2001), even though non-linear methods are known to be more flexible. This fact is related to the asymptotic results in Hall *et al*. (2005): when $d \gg n$, under mild assumptions for data distribution, the pairwise distances between any two points are approximately identical to each other so the data points form an n-simplex. Linear classifiers then become natural choices to discriminate two simplices. Since we focus on classifying high-dimensional low sample size data, only linear SVMs are considered in this paper. In other words, we use the input vector $\mathbf{x}$ as basis functions, i.e. $h(\mathbf{x}) = \mathbf{x}$ and $q = d$.

3 ALGORITHM

The standard SVM and L_1 SVM are often solved using quadratic programming and linear programming methods. However, many standard optimization packages fail to solve (3), because the hinge loss function is not differentiable at zero and the SCAD penalty is not convex in $\mathbf{w}$. In this section, we propose an iterative algorithm to solve the SCAD SVM efficiently and show that only a series of linear equation systems need to be solved.

Successive quadratic algorithm (SQA) is a generalization of Newton's method for unconstrained optimization in that it finds a step away from the current point by minimizing a quadratic approximation of the problem. Numerous optimization packages, including NPSOL, NLPQL, OPSYC, OPTIMA and MATLAB, are found on this approach (More and Wright, 1993). We propose using the SQA to solve the SCAD SVM.

Denote the objective function in (3) by $A(b, \mathbf{w})$. For each i, we have $y_i^2 = 1$ and

$$[1 - y_i(b + \mathbf{w}\cdot\mathbf{x}_i)]_+ = \frac{1 - y_i(b + \mathbf{w}\cdot\mathbf{x}_i)}{2} + \frac{|y_i - (b + \mathbf{w}\cdot\mathbf{x}_i)|}{2}. \qquad (4)$$

Assume an initial value $(b_0, \mathbf{w}_0)$ is given, we consider the local quadratic approximation for the second term in (4):

$$|y_i - (b + \mathbf{w}\cdot\mathbf{x}_i)| \approx \frac{1}{2}\frac{[y_i - (b + \mathbf{w}\cdot\mathbf{x}_i)]^2}{|y_i - (b_0 + \mathbf{w}_0\cdot\mathbf{x}_i)|} + \frac{1}{2}|y_i - (b_0 + \mathbf{w}_0\cdot\mathbf{x}_i)|.$$

For the SCAD penalty $p_\lambda(|w_j|)$, we use the following quadratic approximation

$$p_\lambda(|w_j|) \approx p_\lambda(|w_{j0}|) + \frac{p'_\lambda(|w_{j0}|)}{2|w_{j0}|}(w_j^2 - w_{j0}^2).$$

It is easy to check that both approximating functions have the same gradient as the original functions at the current point (b_0, $\mathbf{w}_0$). Thus minimizing the local quadratic approximation assures the convergence of the algorithm towards the correct descending direction of the original function. The quadratic form of the entire objective A (b, $\mathbf{w}$) is given as

$$\begin{aligned} A(b,\mathbf{w}) \approx & \frac{1}{2} - \frac{1}{2n}\sum_{i=1}^{n} y_i(b+\mathbf{w}\cdot\mathbf{x}_i) \\ & + \frac{1}{4n}\sum_{i=1}^{n}|y_i - (b_0 + \mathbf{w}_0\cdot\mathbf{x}_i)| \\ & + \frac{1}{4n}\sum_{i=1}^{n}\frac{\{y_i - (b+\mathbf{w}\cdot\mathbf{x}_i)\}^2}{|y_i - (b_0+\mathbf{w}_0\cdot\mathbf{x}_i)|} \\ & + \sum_{j=1}^{d}[p_\lambda(|w_{j0}|) + \frac{p'_\lambda(|w_{j0}|)}{2|w_{j0}|}(w_j^2 - w_{j0}^2)]. \end{aligned}$$

Removing the terms which do not involve (b, $\mathbf{w}$), we get

$$\begin{aligned} \tilde{A}(b,\mathbf{w}) = & -\sum_{i=1}^{n}\frac{y_i(b+\mathbf{w}\cdot\mathbf{x}_i)}{2n} + \sum_{j=1}^{d}\frac{p'_\lambda(|w_{j0}|)}{2|w_{j0}|}w_j^2 \\ & - \frac{1}{2n}\sum_{i=1}^{n}\frac{y_i(b+\mathbf{w}\cdot\mathbf{x}_i)}{|y_i-(b_0+\mathbf{w}_0\cdot\mathbf{x}_i)|} \\ & + \frac{1}{4n}\sum_{i=1}^{n}\frac{(b+\mathbf{w}\cdot\mathbf{x}_i)^2}{|y_i-(b_0+\mathbf{w}_0\cdot\mathbf{x}_i)|}. \end{aligned}$$

Define the matrix $X = [\mathbf{1}, \mathbf{x}_1, \ldots, \mathbf{x}_d]$, where $\mathbf{1}$ is the vector of 1s with length n and $\mathbf{x}_j$ is the j-th input vector. Define $\mathbf{y} = [y_1, \ldots, y_n]^T$, $\mathbf{w} = [w_1, \ldots, w_d]^T$ and $\boldsymbol{\epsilon} = [\epsilon_1, \ldots, \epsilon_n]^T$ with $\epsilon_i = y_i - (b_0 + \mathbf{w}_0 \cdot \mathbf{x}_i)$. Define $\mathbf{r} = [y_1/|\epsilon_1|, \ldots, y_n/|\epsilon_n|]^T$, $D_1 = \frac{1}{2n}\mathrm{diag}[1/|\epsilon_1|, \ldots, 1/|\epsilon_n|]$, $P = \frac{1}{2n}(\mathbf{y}+\mathbf{r})^T X$ and $D_2 = \mathrm{diag}[0, p'_\lambda(|w_{10}|)/|w_{10}|, \ldots, p'_\lambda(|w_{d0}|)/|w_{d0}|]$. Then the approximate optimization problem becomes

$$\min_{b,\mathbf{w}} \tilde{A}(b,\mathbf{w}) = \frac{1}{2}\begin{pmatrix} b \\ \mathbf{w}\end{pmatrix}^T Q \begin{pmatrix} b \\ \mathbf{w}\end{pmatrix} - P\begin{pmatrix} b \\ \mathbf{w}\end{pmatrix}. \tag{5}$$

Since (5) is quadratic in (b, $\mathbf{w}$), solving (5) is equivalent to solving the following linear equation system

$$Q\begin{pmatrix} \hat{b} \\ \hat{\mathbf{w}}\end{pmatrix} = P. \tag{6}$$

We propose the following algorithm to solve the SCAD SVM by solving a series of linear equation systems iteratively:

Step 1: Set $k = 1$ and specify the initial value ($b^{(1)}$, $\mathbf{w}^{(1)}$).
Step 2: Let (b_0, $\mathbf{w}_0$) = ($b^{(k)}$, $\mathbf{w}^{(k)}$). Minimize $\tilde{A}(b, \mathbf{w})$ by solving (6). The solution is denoted as ($b^{(k+1)}$, $\mathbf{w}^{(k+1)}$).
Step 3: Let $k = k + 1$. Go to step 2 until convergence.

If some $w_j^{(k)}$ is very close to zero, say, smaller than a certain threshold, then the j-th variable is regarded as redundant. Following Fan and Li (2001), we remove the j-th column from the matrix X and adjust the coefficient matrix in (6) correspondingly. The iteration then continues as a reduced optimization problem. The algorithm stops when there is no change in ($b^{(k)}$, $\mathbf{w}^{(k)}$). The j-th variable is regarded as unimportant if $|w_j| < \epsilon$, where ϵ is a preselected small positive thresholding value. Based on our experiences, the solutions from the standard SVM provide good starting values. In all of our examples, this algorithm converges quickly.

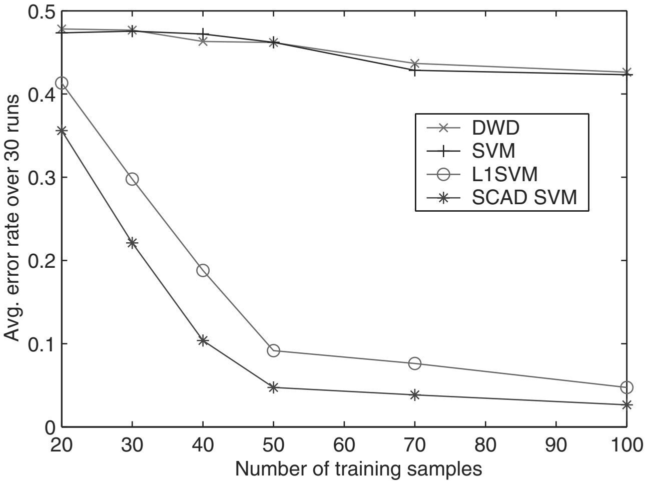


Fig. 3. Average test error rates plotted against the sample size.

4 SIMULATION

In the high-dimensional low sample size setting, we simulate a dataset which contains many redundant variables. Four methods are compared: standard SVM L_1 SVM, SCAD SVM and another classifier called distance weighted discrimination (DWD). The DWD is a large-margin classifier developed using the second-order cone programming by Marron *et al.* (2004), and it does not suffer from the data piling problem as the standard SVM. We use the OSU SVM package (http://svm.sourceforge.net/docs/3.00/api) to implement the SVM and the algorithm of Fung and Mangasarian (2004) to implement the L_1 SVM. A tuning set with the same size as the training set is used to choose the optimal λ. Each classifier is evaluated on a test set of size 500. The thresholding value we used for removing variables is 0.001.

This example is a modification of the example used in Weston *et al.* (2000). There are $d = 200$ inputs and only the first two are relevant. The probability of $Y = +1$ or -1 is equal. The inputs X_1 and X_2 are drawn from a mixture of Normal distributions: with probability 0.7, we have $X_1 = YN(3, 1)$ and $X_2 = N(0, 1)$; with probability 0.3, we have $X_1 = N(0, 1)$ and $X_2 = YN(3, 1)$. The inputs X_j, $j = 3, \ldots, 200$ are independently generated from $N(0, 20)$. The Bayes rule is slightly non-linear around the origin but can be approximated well by linear functions. We consider various settings for the sample size: $n = 20, 30, 40, 50, 70, 100$. In each setting, we run 30 replicates and plot the average test errors in Figure 3. As n increases, the test errors of the SCAD SVM and L_1 SVM decrease prominently compared with those of the DWD and the standard SVM. This suggests that variable selection is important when too many redundant variables are present. Furthermore, the SCAD SVM consistently outperforms the L_1 SVM in all the settings.

Table 1. Number of variables selected by various methods

	$n = 20$	$n = 30$	$n = 40$	$n = 50$	$n = 70$	$n = 100$
DWD	99.53	99.83	102.13	100.23	98.53	98.97
	(0.83)	(0.73)	(0.76)	(0.67)	(0.70)	(0.61)
SVM	64.07	72.67	81.67	78.93	82.80	87.10
	(1.47)	(1.48)	(1.45)	(0.96)	(1.10)	(0.99)
L_1 SVM	15.37	9.10	12.37	11.17	12.47	14.03
	(2.39)	(0.82)	(1.37)	(0.57)	(0.62)	(0.85)
SCAD SVM	8.00	7.90	7.53	6.47	4.73	5.27
	(0.62)	(0.58)	(0.66)	(0.70)	(0.28)	(0.52)

Table 2. Frequency of selecting correct variables in 30 runs

	$n = 20$	$n = 30$	$n = 40$	$n = 50$	$n = 70$	$n = 100$
L_1 SVM	1	4	13	22	30	30
SCAD SVM	0	7	19	25	30	30

Table 1 shows the average number of variables selected over 30 runs for each method. The values in the parentheses are the standard errors of the corresponding mean values. Since the DWD and the SVM are not designed to select variables, they do not perform well in variable selection. It is observed that the SCAD SVM selects a smaller and a more stable (with smaller standard errors) number of variables than the L_1 SVM in almost all cases. Table 2 compares the frequency of selecting correct variables (X_1 and X_2) in 30 runs between the L_1 and SCAD SVM. When the sample size is too small, both methods experience a certain level of difficulty in selecting two important variables from 200 variables. As n increases, both methods tend to select the two variables more correctly. The SCAD SVM performs slightly better than the L_1 SVM.

In Figure 4, for one particular simulated data of size $n = 100$ we plot the classification boundaries given by the Bayes rule and four learning methods. Note the Bayes rule is available only for simulated data. We use‘o’ for the points from the +1 class and ‘x’ for those from the -1 class. The line symbols are the DWD (dashed), the SVM (dotted), the L_1 SVM (dash-dotted), the SCAD SVM (thick solid), and the Bayes rule (thin solid). Since only X_1 and X_2 are truly relevant to the classification boundary, all the classifiers have been projected from the 200-dimensional input space to the first two-dimensional subspace. Figure 4 shows that the SCAD SVM classifier is the closest to the Bayes rule among all the classifiers. This explains why the SCAD SVM has the smallest test error rate as shown in Figure 3.

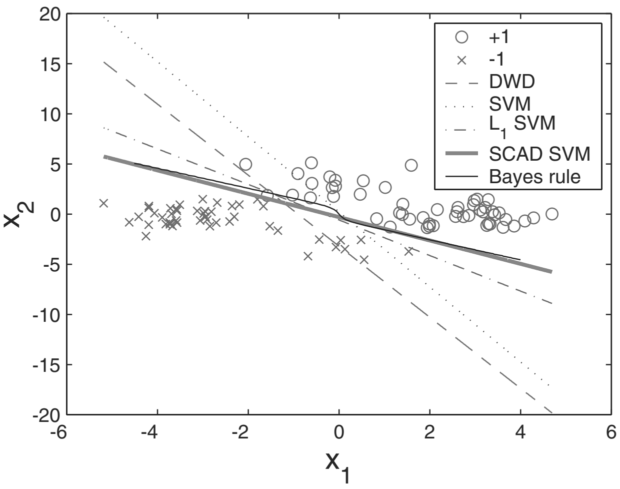


Fig. 4. The Bayes rule and classification boundaries given by four methods (projected onto the first two dimensions) in the simulated example.

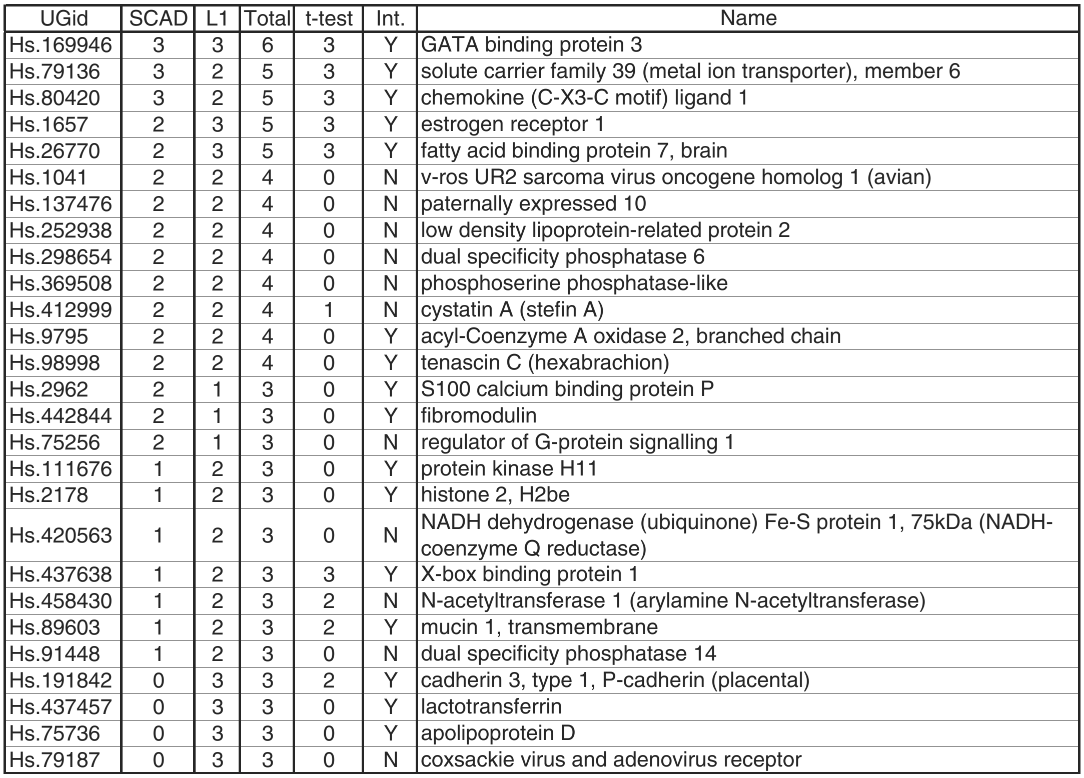
5 REAL DATA

In practice, gene-ranking methods are widely used to select genes, which are highly differentiated between two types of tissues prior to training. Ranking criteria are often based on the t-statistic (Pan, 2002) or correlation coefficients. For each gene $\mathbf{x}_j$, the mean μ_j^+ (resp. μ_j^-) and standard deviation σ_j^+ (resp. σ_j^-) using only the tissues labeled +1 (resp. -1) are calculated. Define $w_j = (\mu_j^+ - \mu_j^-)/(\sigma_j^+ + \sigma_j^-)$. Golub *et al.* (1999) selected p genes with the largest positive w_js and p genes with the largest negative w_js. Furey *et al.* (2000) used $|w_j|$ to select top p genes. Pavlidis *et al.* (2001) suggested a Fisher-discriminant type correlation coefficient. We compare our method with two ranking methods: t-test and Furey *et al.* (2000). Each ranking criterion is first applied to select the top 50 and 100 genes, then the standard SVM is fitted. There are two problems with ranking methods: (1) one has to specify the number of selected genes p in advance and often subjectively and (2) the selection is individual-based and hence ignores correlation among genes.

5.1 UNC breast cancer dataset

Three public microarray gene expression datasets are used in this section. They are from Perou *et al.* (2000), van't Veer *et al.* (2002), Sotiriou *et al.* (2003), respectively, and for convenience we use ‘Stanford’, ‘Rosetta’, and ‘Singapore’ to refer them. Originally the three sets have 5974 genes and 104 patients, 24187 genes and 97 patients, and 7650 genes and 99 patients, respectively. In a recent study (Z. Hu, C. Fan, J.S. Marron, X. He, B.F. Qaqish, G. Karaca, C. Livasy, L. Carey, E. Reynolds, L. Dressler, A. Nobel, J. Parker, W.G. Ewend, L.R. Sawyer, D. Xiang, J. Wu, Y. Liu, M. Karaca, R. Nanda, M. Tretiakova, A.R. Orrico, D. Dreher, J.P. Palazzo, L. Perreard, E. Nelson, M. Mone, H. Hansen, M. Mullins, J.F. Quackenbush, O.I. Olapade, B.S. Bernard and C.M. Perou (2005). The molecular portraits of breast tumors are conserved across microarray platforms, manuscript submitted.), the three datasets are imputed for missing values, combined and then corrected to adjust the bias since they are from three different batches. The DWD is used for the batch adjustment process; see Benito *et al.* (2004) for a detailed description of the systematic bias adjustment for microarray data using the DWD. As for the gene identifier for the combined dataset, UniGene is used since it is most convenient to map the identifiers from each dataset to UniGene identifier (Build 161). In case of multiple occurrences of a UCID, the median value is used.

The combined dataset has 2924 genes and totally 300 patients. Our primary interest is to select important genes and use them to classify the tissues into two different types of breast cancer. We use the source information to separate the whole data into three folds



UGid	SCAD	L1	Total	t-test	Int.	Name
Hs.169946	3	3	6	3	Y	GATA binding protein 3
Hs.79136	3	2	5	3	Y	solute carrier family 39 (metal ion transporter), member 6
Hs.80420	3	2	5	3	Y	chemokine (C-X3-C motif) ligand 1
Hs.1657	2	3	5	3	Y	estrogen receptor 1
Hs.26770	2	3	5	3	Y	fatty acid binding protein 7, brain
Hs.1041	2	2	4	0	N	v-ros UR2 sarcoma virus oncogene homolog 1 (avian)
Hs.137476	2	2	4	0	N	paternally expressed 10
Hs.252938	2	2	4	0	N	low density lipoprotein-related protein 2
Hs.298654	2	2	4	0	N	dual specificity phosphatase 6
Hs.369508	2	2	4	0	N	phosphoserine phosphatase-like
Hs.412999	2	2	4	1	N	cystatin A (stefin A)
Hs.9795	2	2	4	0	Y	acyl-Coenzyme A oxidase 2, branched chain
Hs.98998	2	2	4	0	Y	tenascin C (hexabrachion)
Hs.2962	2	1	3	0	Y	S100 calcium binding protein P
Hs.442844	2	1	3	0	Y	fibromodulin
Hs.75256	2	1	3	0	N	regulator of G-protein signalling 1
Hs.111676	1	2	3	0	Y	protein kinase H11
Hs.2178	1	2	3	0	Y	histone 2, H2be
Hs.420563	1	2	3	0	N	NADH dehydrogenase (ubiquinone) Fe-S protein 1, 75kDa (NADH-coenzyme Q reductase)
Hs.437638	1	2	3	3	Y	X-box binding protein 1
Hs.458430	1	2	3	2	N	N-acetyltransferase 1 (arylamine N-acetyltransferase)
Hs.89603	1	2	3	2	Y	mucin 1, transmembrane
Hs.91448	1	2	3	0	N	dual specificity phosphatase 14
Hs.191842	0	3	3	2	Y	cadherin 3, type 1, P-cadherin (placental)
Hs.437457	0	3	3	0	Y	lactotransferrin
Hs.75736	0	3	3	0	Y	apolipoprotein D
Hs.79187	0	3	3	0	N	coxsackie virus and adenovirus receptor

Fig. 5. Gene selection frequency for breast cancer data.

Table 3. Cross validation error rate for breast cancer data

	Stanford	Rosetta	Singapore	Average
t-Test ($P = 50$)	0.202	0.217	0.192	0.203
t-Test ($P = 100$)	0.192	0.206	0.111	0.170
SVM	0.154	0.175	0.051	0.127
L_1 SVM	0.125	0.216	0.081	0.141
SCAD SVM	0.115	0.175	0.061	0.117

Table 4. Number of selected genes for breast cancer data

	Stanford	Rosetta	Singapore	Average
L_1 SVM	59	63	72	65
SCAD SVM	15	19	31	22

naturally. We train each classifier on two folds and test it on the remaining one. For example, the SCAD SVM is first trained on Rosetta and Singapore data, then tested on Stanford data. We refer this as ‘Stanford’ learning. Then we repeat this procedure for the other two learnings, referred as the ‘Rosetta’ learning and the ‘Singapore’ learning. To choose the tuning parameter λ, we use 10-fold cross validation within the training set.

Table 3 shows the cross validation error in each learning and the average error rate for five methods. In the Stanford learning, the SCAD SVM has the lowest error rate 0.115. In the Rosetta learning, the SCAD SVM and the SVM are equally best. In the Singapore learning, the SVM is best and the SCAD SVM is slightly worse and the second best. Two ranking methods give the same result (hence only *t*-test reported), and they are worse than the SVM which uses all the genes. It can be explained by that the ranking methods select individual genes separately and ignore their correlations. For a fair comparison, the DWD is not included here because it was used for preprocessing the combined data. Overall speaking, the SCAD SVM gives the lowest error rate among all.

Table 4 gives the number of genes selected in each learning by the SCAD SVM and the L_1 SVM. The L_1 SVM selects 59–72 genes in each learning, where the SCAD SVM only selects 15–31 genes for each case. Note that the misclassification rates of the SCAD SVM shown in Table 2 are only based on the selected 15–31 genes, which shows the very strong gene selection power of the method. Also note that the gene selection results of both methods are consistent in the sense that they both select the smallest number of genes for prediction in the Stanford learning and both select the largest number of genes in the Singapore learning.

Figure 5 lists the UniGene identifiers of all the genes that are selected at least three times by either the SCAD SVM or the L_1 SVM. In the second and third columns are the frequencies of each gene selected in three learnings by, respectively, the SCAD SVM and the L_1 SVM. The sum of these two columns is in the fourth column, and the fifth column lists the number of times that each gene selected by the *t*-test. The sixth column shows whether or not the selected gene is in the list of ‘intrinsic’ genes selected by Perou

Table 5. Cross validation error and the number of metabolites selected for metabolism data

	Test error	Metabolite selected
t-Test ($P = 50$)	0.370 (0.018)	50
t-Test ($P = 100$)	0.235 (0.016)	100
Furey ($P = 50$)	0.375 (0.011)	50
Furey ($P = 100$)	0.230 (0.011)	100
DWD	0.159 (0.012)	315
SVM	0.190 (0.013)	307
L_1 SVM	0.174 (0.012)	32
SCAD SVM	0.143 (0.020)	18

et al. (2000). The last column displays the corresponding descriptive names of UniGene identifiers. We see that the top gene Hs.169946 is selected by all the methods in each learning and also classified as an intrinsic gene. Hs.79136 and Hs.80420, both intrinsic genes, are selected three times by the SCAD SVM but only two times by the L_1 SVM. The top five genes are intrinsic and also selected by the t-test. However, there are 9 out of total 27 selected genes which are neither intrinsic nor selected by t-test. This suggests that one should consider the multivariate gene selection approaches, such as the SCAD SVM and the L_1 SVM, rather than individual gene-by-gene methods such as t-test procedures.

5.2 Metabolism dataset

Metabolic datasets contain the quantitative measurements of all small molecule metabolites in biological samples. Some biological studies show that most of the metabolites are not informative in predicting disease or non-disease outcomes (Stitt and Fernie, 2003). Consequently, hybrid methods that incorporate variable selection with classification techniques can be very effective in analyzing datasets of this sort. Our metabolism dataset is provided by Metabolon, Inc. and we are actually one of the first research groups to analyze it.

There are metabolic profiles of 63 samples: 32 healthy subjects and 31 subjects diagnosed with a certain disease. Within the patient group, 9 subjects are taking medication and 22 are are not. For each sample, its metabolic profile contains the intensity levels of 317 compounds (metabolites). Table 5 shows the average leave-one-out cross validation error and the number of metabolites selected by each method. The SCAD SVM gives the smallest cross validation error 0.143. Moreover, the SCAD SVM selects 18 important metabolites out of 317 and the L_1 SVM selects 32 metabolites. Hence the SCAD SVM achieves the highest classification accuracy using the fewest number of metabolites. This result has great implications on metabolic studies, since one main issue from biological aspects is to identify which metabolites are more relevant to the occurrence of the disease.

6 DISCUSSION

For high-dimensional low sample size data, redundant input variables can affect the performances of classifiers. How to combine variable selection and classification in a unified framework has become an imminent problem. In this paper, we propose a new regularization technique for simultaneous classification and variable selection in the SVM. Compared with other methods, our nonconvex penalty function achieves more compactness and better accuracy, showing great potential for gene selection in cancer classification problems.

The non-convexity of the penalty function introduces great difficulties in optimization. To implement the SCAD SVM efficiently, we developed an iterative procedure based on the successive quadratic algorithm. From both simulated and real data analysis, we found that this algorithm converges quickly. Overall, the SCAD SVM gives competitive results in terms of both variable selection and classification.

ACKNOWLEDGEMENTS

The authors are grateful to Dr C. M. Perou and his colleagues for providing breast cancer microarray datasets. This research was supported by Statistical and Applied Mathematical Sciences Institute (SAMSI) data mining and machine learning program during 2003-2004. H.H.Z.'s research is partially supported by NSF grant DMS-0405913. X.L.'s research is partially supported by a fellowship from Charles Phelps Taft Research Center.

Conflict of interest: none declared.

REFERENCES

Bae,K. and Mallick,B.K. (2004) Gene selection using a two-level hierarchical Bayesian model. *Bioinformatics*, **20**, 3423–3430.

Benito,M. *et al*. (2004) Adjustment of systematic microarray data biases. *Bioinformatics*, **20**, 105–144.

Boser,E., Guyon,M. and Vapnik,V. (1992) A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth ACM Workshop on Computational Learning Theory*, Pittsburgh, PA, pp. 144–152.

Bradley,P.S. and Mangasarian,O.L. (1998) Feature selection via concave minimization and support vector machines. In *Proceedings of the 13th International Conference on Machine Learning* CA, pp. 82–90.

Chaplle,O. *et al*. (2002) Choosing kernel parameters for SVMs. *Mach. Learning*, **46**, 131–159.

Cristianini,N. and Shawe-Taylor,J. (1999) *An Introduction to SVM*. Cambridge University Press, Cambridge, MA.

Devore,J. and Peck,R. (1997) *Statistics: The Exploration and Analysis of Data*. 3rd edn. Duxbury Press, Pacific Grove, CA.

Donoho,D. and Johnstone,I. (1994) Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, **81**, 425–455.

Fan,J. and Li,R. (2001) Variable selection via penalized likelihood. *J. Am. Stat. Assoc*., **96**, 1348–1360.

Fung,G. and Mangasarian,O.L. (2004) A feature selection Newton method for support vector machine classification. *Comput. Optim. Appl. J*., **28**, 185–202.

Furey,T. *et al*. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, **16**, 906–914.

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

Grandvalet,Y. and Canu,S. (2002) Adaptive scaling for feature selection in SVMs. *Neural Inform. Processing Syst.,* NIPS 2002, 553–560.

Guyon,I. *et al*. (2002) Gene selection for cancer classification using SVM. *Mach. Learning*, **46**, 389–422.

Hall,P. *et al*. (2005) Geometric representation of high dimension low sample size data. *J. R. Statist. Soc. B*, **67**, 427–444.

Hastie,T., Tibshirani,R. and Friedman,J. (2001) *The Elements of Statistical Learning*. Springer, NY.

He,W. (2004) A spline function approach for detecting differentially expressed genes in microarray data analysis. *Bioinformatics*, **20**, 2954–2963.

Kitter,J. (1986) Feature selection and extraction. In Young,T.Y. and Fu,K.-S. (eds), *Handbook of Pattern Recognition and Image Processing*. Academic Press, NY.

Lee,E. *et al*. (2003) Gene selection: a Bayesian variable selection approach. *Bioinformatics*, **19**, 90–97.

Lin,Y. (2002) SVM and the Bayes rule in classification. *Data Mining Knowledge Discov*., **6**, 259–275.

Marron,J.S. *et al*. (2004) Distance weighted discrimination. *J. Am. Stat. Assoc.,* in press.

More,J.J. and Wright,S.J. (1993) *Optimization Software Guide*. SIAM, Philadelphia.

Mukherjee,S., Tamayo,P., Slonim,D., Verri,A., Golub,T., Messirov,P. and Poggio,T. (2000) SVM classification of microarray data. *AI memo 182, CBCL paper 182*. MIT, MA.

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

Parvlidis,P., Weston,J., Cai,J. and Grundy,W.N. (2001) Gene functional analysis from heterogeneous data. In *Proceedings of 5th International Conference on Computational Biology,* Pittsburgh, PA, pp. 249–255.

Perou,C.M. *et al*. (2000) Molecular portraits of human breast tumors. *Nature*, **406**, 747–752.

Rakotomamonjy,A. (2003) Variable selection using SVM-based Criteria. *J. Mach. Learning Res*., **3**, 1357–1370.

Sotiriou,C. *et al*. (2003) Breast cancer classification and prognosis based on gene expression profiles from a population-based study. *Proc. Natl Acad. Sci. USA*, **100**, 10393–10398.

Stitt,M. and Fernie,A.R. (2003) From measurements of metabolites to metabolomics: an ‘on the fly’ perspective illustrated by recent studies of carbon-nitrogen interactions. *Curr. Opin. Biotechnol*., **14**, 136–144.

Thomas,G. *et al*. (2001) An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles. *Genome Res*., **11**, 1227–1236.

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

Troyanskaya,G. *et al*. (2002) Nonparametric methods for identifying differentially expressed genes in microarray data. *Bioinformatics*, **18**, 1454–1461.

van’t Veer,L. *et al*. (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530–536.

Vapnik,V. (1995) *The Nature of Statistical Learning Theory*. Springer, NY.

West,M. (2003) Bayes factor regression models in the ‘large p, small n’ paradigm. *Bayesian Statistics*, **7**, 723–732.

Weston,J. *et al*. (2000) Feature selection for SVMs. *Adv. Neural Inform. Processing Syst*., **13**, 668–674.

Zhu,J. *et al*. (2003) 1-norm SVMs. *Neural Inform. Processing Systems*, **16**, 49–56.