Communications of the Korean Statistical Society
2012, Vol. 19, No. 5, 655–662

DOI: http://dx.doi.org/10.5351/CKSS.2012.19.5.655

Multiclass Support Vector Machines with SCAD

Kang-Mo Jung[1,a]

[a]Department of Statistics and Computer Science, Kunsan National University

Abstract

Classification is an important research field in pattern recognition with high-dimensional predictors. The support vector machine(SVM) is a penalized feature selector and classifier. It is based on the hinge loss function, the non-convex penalty function, and the smoothly clipped absolute deviation(SCAD) suggested by Fan and Li (2001). We developed the algorithm for the multiclass SVM with the SCAD penalty function using the local quadratic approximation. For multiclass problems we compared the performance of the SVM with the L_1, L_2 penalty functions and the developed method.

Keywords: Local quadratic approximation, multiclass support vector machine, penalized, smoothly clipped absolute deviation.

1. Introduction

Classification is one of important methods in pattern recognition, especially in data mining procedures. There are many approaches of classification such as linear discriminant analysis, logistic regression, k-nearest neighbor, boosting and neural networks (Hastie *et al.*, 2001). In recent days the support vector machine(SVM) was introduced by Vapnik (1995), which is an optimal margin classifier of a Perceptron. The SVM has been successfully applied to engineering and bioinformatics (Zhang *et al.*, 2006).

The standard binary SVM can be formulated as a penalized logistic regression with a convex hinge loss function and a L_2 penalized function. In regression problems Tibshirani (1996) proposed the least absolute shrinkage and selection operator(LASSO) which used the L_1 penalized function, which can automatically select useful input variables and estimate the regression parameters simultaneously. The solution of LASSO can be biased for large coefficients. Fan and Li (2001) proposed the non-convex penalty function to address the deficiencies of LASSO, the smoothly clipped absolute deviation(SCAD) penalty which has the advantage of the so-called "oracle property", while the LASSO does not have. In SVM the L_2 penalty includes all input variables, which means that the system can suffer from the multicollinearity. This is a drawback when there are lots of noise input variables (Efron *et al.*, 2004). The L_1 penalty (Bradley and Mangasarian, 1998), on the other hand, selects a small number of input variables. The SCAD penalty is an approach of making up for the deficiencies of the L_1 penalty.

We can solve the L_2 SVM and the L_1 SVM by using quadratic programming and linear programming methods. However, the SCAD SVM is not convex. It means that we can not use linear programming or quadratic programming methods. It requires another computing algorithm. Zhang *et al.*

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2011-0013456).

[1] Professor, Department of Statistics and Computer Science, Kunsan National University, Kunsan 573-701, Korea.
E-mail: kmjung@kunsan.ac.kr

(2006) proposed a quadratic approximation to the SCAD penalty in binary classifications. Successive quadratic algorithm(SQA) is a generalization of Newton's method for unconstrained optimization.

On the practical point of view the number of target classes is more than two. Many approaches have been proposed to solve multi-classification problems. One of them is the so-called "one-versus-rest" algorithm by constructing classifiers corresponding to the number of class (Weston and Watkins, 1999). Lee *et al.* (2004) proposed a simultaneous algorithm, because the "one-versus-rest" algorithm has poor performance if there exists a dominating class. In this case most of observations can be allocated to the dominating class by using the "one-versus-rest" algorithm.

In this article we extend the binary SVM with the SCAD penalty function to the multiclass SVM. We use the simultaneous multiclass SVM algorithm proposed by Lee *et al.* (2004) and Liu *et al.* (2007). This article is organized as follows. In Section 2 we describe the formulation of the classification problem and preliminaries for suggesting the algorithm. Penalty functions of L_1, L_2 and SCAD are briefly reviewed. Section 3 gives an algorithm to implement the multiclass SVM with the SCAD penalty. A SQA algorithm is developed to minimize the non-differentiable and nonconvex objective function in the SCAD multiclass SVM by solving a series of linear equation systems. Section 4 provides the results of small simulation and some real data sets. It shows that the developed algorithm is able to select the correct model more frequently, comparing to the L_1 and L_2 SVM. Some discussion and concluding remarks are given in Section 5.

2. Problem Formulation and Preliminaries

Consider a general K-class classification problem with a training set $\{\mathbf{x}_i, y_i\}_{i=1}^n$, where $\mathbf{x}_i = (x_1, \ldots, x_d)^T \in \mathcal{R}^d$ is the input vector and $y_i \in \{1, 2, \ldots, K\}$ represents its class label. We need to find a K-dimensional function vector $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_K(\mathbf{x}))$ with a sum-to-zero constraint $\sum_{k=1}^K f_k(\mathbf{x}) = 0$ for any $\mathbf{x} \in \mathcal{R}^d$, minimizing the following quantity (Lee *et al.*, 2004)

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i \neq k)(f_k(\mathbf{x}_i) + 1)_+ + \sum_{k=1}^{K} p_\lambda(f_k), \tag{2.1}$$

where $p_\lambda(\cdot)$ is a penalized function, $(\cdot)_+$ means the truncation function and $I(y_i \neq k)$ is the indicator function having zero if $y_i = k$ and one otherwise. If we consider unequal costs for misclassification, the indicator function can be an unequal cost function value. In this article we consider the linear decision function $f_k(\mathbf{x}) = b_k + \mathbf{w}_k^T\mathbf{x}$ and the SCAD penalty function. Then (2.1) becomes

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i \neq k)\left(\mathbf{w}_k^T\mathbf{x}_i + b_k + 1\right)_+ + \sum_{j=1}^{d}\sum_{k=1}^{K} p_\lambda(|w_{jk}|), \tag{2.2}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} b_k = 0, \quad \sum_{k=1}^{K} w_{jk} = 0, \quad \text{for} \;\; j = 1, \ldots, d, \tag{2.3}$$

where

$$p_\lambda(|w|) = \begin{cases} \lambda|w|, & \text{if } |w| \leq \lambda, \\ -\dfrac{|w|^2 - 2a\lambda|w| + \lambda^2}{2(a-1)}, & \text{if } \lambda < |w| \leq a\lambda, \\ \dfrac{(a+1)\lambda^2}{2}, & \text{if } |w| > a\lambda, \end{cases} \tag{2.4}$$

and $a > 2$ and $\lambda > 0$ is a tuning parameter. The function $p_\lambda(|w|)$ has a continuous first-order derivative except the origin. Fan and Li (2001) recommended $a = 3.7$ with the consideration of the Bayes risks and performance analysis for many problems. In this article we set $a = 3.7$. The classification rule induced by $\mathbf{f}(\mathbf{x})$ naturally becomes

$$\phi(\mathbf{x}) = \arg\max_j f_j(\mathbf{x}). \tag{2.5}$$

For binary classification the objective function (2.2) can be reduced to Equation (3) in Zhang *et al.* (2006).

The parameter λ in (2.4) controls the trade-off between data-fitting and model sparsity. For too large λ the procedure yields a very sparse classifier to underfit the training data, while for too small λ the procedure has the same results with the linear classifier without penalized function to overfit the training data. Thus penalized methods require selection of the tuning parameter λ. It is usual to adopt some data-driven method such as cross-validation. See Jung (2008) for linear regression model.

3. Algorithm

When we use L_1 and L_2 instead of the SCAD function in (2.2) the objective function (2.2) reduces to the L_1 SVM and the standard SVM, respectively. For the L_1 and standard SVM we can solve the optimization problem using linear programming and quadratic programming. However the equation (2.2) can not be solved by general optimization algorithms, because the hinge function is not differentiable at zero and the SCAD penalized function is not convex in $\mathbf{w}$. That is, (2.2) becomes a nonlinear programming problem. Zhang *et al.* (2006) proposed a successive quadratic algorithm(SQA) which is an approximation method of Newton's method.

The constrained optimization problem (2.2) can be the unconstrained optimization problem by adopting the constraint (2.3)

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K-1} I(y_i \neq k)\left(1 + \mathbf{w}_k^T\mathbf{x}_i + b_k\right)_+ + \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq K)\left(1 - \sum_{k=1}^{K-1}\mathbf{w}_k^T\mathbf{x}_i - \sum_{k=1}^{K-1} b_k\right)_+ .$$

Let a_{ik} be the (i,k) element of the $n \times K$ matrix having the indicator function value of $I(y_i \neq k)$. The vector of parameter is denoted by $\boldsymbol{\eta}^T = (\boldsymbol{\eta}_1^T, \ldots, \boldsymbol{\eta}_{K-1}^T)$, where $\boldsymbol{\eta}_k = (b_k, w_{1k}, \ldots, w_{dk})^T$. Define the augmented data vector $\tilde{\mathbf{x}}_i$ by $(1, \mathbf{x}_i^T)^T$. Then using the fact that $u_+ = (u + |u|)/2$ and the approximation $|u| \approx (1/2)u^2/|u_0| + (1/2)|u_0|$ for nonzero u_0 near u, we have $u_+ \approx (1/4)u^2/|u_0| + (1/2)u + (1/4)|u_0|$. The above objective function can be written by

$$A_2 + A_1 + B_2 + B_1 + \text{constant},$$

where

$$A_2 = \frac{1}{4n}\sum_{i=1}^{n}\sum_{k=1}^{K-1}\frac{a_{ik}\left(1 + \boldsymbol{\eta}_k^T\tilde{\mathbf{x}}_i\right)^2}{\left|\boldsymbol{\eta}_k^{0T}\tilde{\mathbf{x}}_i + 1\right|}, \qquad A_1 = \frac{1}{2n}\sum_{i=1}^{n}\sum_{k=1}^{K-1} a_{ik}\left(1 + \boldsymbol{\eta}_k^T\tilde{\mathbf{x}}_i\right),$$

$$B_2 = \frac{1}{4n}\sum_{i=1}^{n}\frac{a_{iK}\left(1 - \sum_{k=1}^{K-1}\boldsymbol{\eta}_k^T\tilde{\mathbf{x}}_i\right)^2}{\left|1 - \sum_{k=1}^{K-1}\boldsymbol{\eta}_k^{0T}\tilde{\mathbf{x}}_i\right|}, \qquad B_1 = \frac{1}{2n}\sum_{i=1}^{n} a_{iK}\left(1 - \sum_{k=1}^{K-1}\boldsymbol{\eta}_k^T\tilde{\mathbf{x}}_i\right).$$

After some algebra we obtain

$$\begin{aligned} A_2 + A_1 + B_2 + B_1 &= \frac{1}{2}\sum_{k=1}^{K-1}\sum_{l=1}^{K-1} \boldsymbol{\eta}_k^T \left(Q_{B_2} + I(k=l)Q_{A_2,k}\right)\boldsymbol{\eta}_l \\ &+ \sum_{k=1}^{K-1} \boldsymbol{\eta}_k^T \left(L_{A_2,k} + L_{A_1,k} + L_{B_2} + L_{B_1}\right) + \text{constant}, \end{aligned} \tag{3.1}$$

where

$$Q_{A_2,k} = \frac{1}{2n}\sum_{i=1}^{n} \frac{a_{ik}}{\left|1+\boldsymbol{\eta}_k^{0T}\tilde{\mathbf{x}}_i\right|}\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i^T, \qquad Q_{B_2} = \frac{1}{2n}\sum_{i=1}^{n} \frac{a_{iK}}{\left|1-\sum_{k=1}^{K-1}\boldsymbol{\eta}_k^{0T}\tilde{\mathbf{x}}_i\right|}\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i^T,$$

$$L_{A_2,k} = \frac{1}{2n}\sum_{i=1}^{n} \frac{a_{ik}}{\left|1+\boldsymbol{\eta}_k^{0T}\tilde{\mathbf{x}}_i\right|}\tilde{\mathbf{x}}_i, \qquad L_{A_1,k} = \frac{1}{2n}\sum_{i=1}^{n} a_{ik}\tilde{\mathbf{x}}_i,$$

$$L_{B_2} = -\frac{1}{2n}\sum_{i=1}^{n} \frac{a_{iK}}{\left|1-\sum_{k=1}^{K-1}\boldsymbol{\eta}_k^{0T}\tilde{\mathbf{x}}_i\right|}\tilde{\mathbf{x}}_i, \qquad L_{B_1} = -\frac{1}{2n}\sum_{i=1}^{n} a_{iK}\tilde{\mathbf{x}}_i.$$

In addition, we have the approximating function of the SCAD function

$$p_\lambda(|w|) \approx p_\lambda(|w_0|) + \frac{p'_\lambda(|w_0|)}{2|w_0|}\left(w^2 - w_0^2\right),$$

where

$$p'_\lambda(|w|) = \begin{cases} \lambda, & \text{if } 0 \le |w| < \lambda, \\ \dfrac{a\lambda - |w|}{a-1}, & \text{if } \lambda \le |w| < a\lambda, \\ 0, & \text{if } |w| > a\lambda. \end{cases}$$

The convergence of the approximating function can be proved by the fact that the function has the same derivative function at u_0. Thus the right term in (2.2) can be rewritten by

$$C_1 + C_2 + \text{constant},$$

where

$$C_1 = \frac{1}{2}\sum_{j=1}^{d}\sum_{k=1}^{K-1} \frac{p'_\lambda\left(\left|w_{jk}^0\right|\right)}{\left|w_{jk}^0\right|} w_{jk}^2, \qquad C_2 = \frac{1}{2}\sum_{j=1}^{d} \frac{p'_\lambda\left(\left|\sum_{m=1}^{K-1} w_{jm}^0\right|\right)}{\left|\sum_{m=1}^{K-1} w_{jm}^0\right|}\sum_{k=1}^{K-1}\sum_{l=1}^{K-1} w_{jk}w_{jl}.$$

Following the above computation yields

$$C_1 + C_2 = \frac{1}{2}\sum_{k=1}^{K-1}\sum_{l=1}^{K-1} \boldsymbol{\eta}_k^T \left(Q_{C_2} + I(k=l)Q_{C_1,k}\right)\boldsymbol{\eta}_l + \text{constant}, \tag{3.2}$$

where

$$Q_{C_2} = \text{diag}\left(0, \frac{p'_\lambda\left(\left|\sum_{m=1}^{K-1} w_{1m}^0\right|\right)}{\left|\sum_{m=1}^{K-1} w_{1m}^0\right|}, \ldots, \frac{p'_\lambda\left(\left|\sum_{m=1}^{K-1} w_{dm}^0\right|\right)}{\left|\sum_{m=1}^{K-1} w_{dm}^0\right|}\right),$$

$$Q_{C_1,k} = \operatorname{diag}\left(0, \frac{p'_\lambda\left(\left|w^0_{1k}\right|\right)}{\left|w^0_{1k}\right|}, \ldots, \frac{p'_\lambda\left(\left|w^0_{dk}\right|\right)}{\left|w^0_{dk}\right|}\right).$$

By combining (3.1) and (3.2) the objective function (2.2) and (2.3) can be written by

$$\frac{1}{2}\sum_{k=1}^{K-1}\sum_{l=1}^{K-1} \boldsymbol{\eta}_k^T \left\{(Q_{B_2} + Q_{C_2}) + I(k = l)(Q_{A_2,k} + Q_{C_1,k})\right\} \boldsymbol{\eta}_l + \sum_{k=1}^{K-1} \boldsymbol{\eta}_k^T \left(L_{A_2,k} + L_{A_1,k} + L_{B_2} + L_{B_1}\right).$$

Therefore, the constrained optimization problem (2.2) and (2.3) is converted into the unconstrained quadratic optimization problem given by

$$\frac{1}{2}\boldsymbol{\eta}^T \mathbf{Q} \boldsymbol{\eta} + \boldsymbol{\eta}^T \mathbf{L}, \tag{3.3}$$

where $\mathbf{Q}$ is the $(K-1)(d+1) \times (K-1)(d+1)$ matrix having off-diagonal $(d+1) \times (d+1)$ partitioned matrix $Q_{B_2} + Q_{C_2}$ and the kth diagonal partitioned matrix $Q_{B_2} + Q_{C_2} + Q_{A_2,k} + Q_{C_1,k}$ and $\mathbf{L}$ is the vector of length $(K-1)(d+1)$ having the kth vector $L_{A_2,k} + L_{A_1,k} + L_{B_2} + L_{B_1}$ of length $(d+1)$. Since (3.3) is quadratic in $\boldsymbol{\eta}$, the optimal solution of the objective function (2.2) is the solution of the linear equations

$$\mathbf{Q}\boldsymbol{\eta} + \mathbf{L} = \mathbf{0}. \tag{3.4}$$

Therefore we describe an algorithm for the multiclass SVM with the SCAD penalty function by the following iterative steps:

Step 1. Let $l = 1$ and set the initial values $\boldsymbol{\eta}_k^{(l)}$ for $k = 1, \ldots, K-1$.

Step 2. Let $\boldsymbol{\eta}^0 = \boldsymbol{\eta}^{(l)}$. Solution to the equation (3.4) gives new solution $\boldsymbol{\eta}^{(l+1)}$.

Step 3. Let $l = l + 1$. Go to Step 2 until convergence.

The algorithm will stop when the difference between $\boldsymbol{\eta}^{(l+1)}$ and $\boldsymbol{\eta}^{(l)}$ is less than the given tolerance. For our simulation in Section 4 we use the tolerance 10^{-3} for the sum of the absolute difference. For the initial solution of $\boldsymbol{\eta}^{(1)}$ we use the multiple linear discriminant function.

4. Simulation and Real Data

In this section we demonstrate simulations to show the effectiveness of the multiclass SCAD SVM. We numerically compare the proposed method with the L_1 SVM and the L_2 SVM. All simulations are carried out using R codes.

4.1. Simulation

We consider the multiclass example with $K = 3$ given in Liu *et al.* (2007). The data $\mathbf{x}$ is generated from the bivariate normal distribution $N(\boldsymbol{\mu}_k, \sigma^2\mathbf{I})$ for $k = 1, 2, 3$, where $\boldsymbol{\mu}_1 = (\sqrt{3}, 1)^T, \boldsymbol{\mu}_2 = (-\sqrt{3}, 1)^T, \boldsymbol{\mu}_3 = (0, -2)^T$ and $\sigma^2 = 2$. For the selection of tuning parameter we generate the sample with the sample size 100. From the simulation results we set $\lambda = 0.11$ for the L_1 SVM, the L_2 SVM and the SCAD SVM. After learning the training data with the sample size 100 we compute the mean and variance of the misclassification rate for the test data with the sample size 1,000. We summarize the results for 100 replications in Table 1. We can see that the performance are comparable with three

Table 1: Simulation results for three class

Method	Test error	Correct
L_2 SVM	0.2267(0.0039)	4(0)
L_1 SVM	0.2514(0.0111)	4(0)
SCAD SVM	0.1941(0.0045)	4(0)

Table 2: Simulation results for three class with noise variables

No. of noise	Method	Test error	Correct	Incorrect
20	L_2 SVM	0.2918 (0.0294)	4 (0)	31.40 (2.6208)
	L_1 SVM	0.2893 (0.0295)	4 (0)	30.54 (2.6185)
	SCAD SVM	0.2709 (0.0326)	3.98 (0.1407)	27.97 (3.4449)
40	L_2 SVM	0.4076 (0.0440)	4 (0)	64.86 (4.3686)
	L_1 SVM	0.3670 (0.0470)	3.96 (0.1825)	58.40 (9.8246)
	SCAD SVM	0.3446 (0.0365)	4 (0)	57.76 (7.9163)
60	L_2 SVM	0.6005 (0.0350)	3.90 (0.3051)	98.40 (6.3658)
	L_1 SVM	0.4710 (0.0450)	3.93 (0.2537)	90.53 (6.1573)
	SCAD SVM	0.4046 (0.0671)	3.76 (0.8976)	77.46 (10.0059)
80	L_2 SVM	0.6529 (0.0212)	3.90 (0.3051)	131.36 (5.1292)
	L_1 SVM	0.5413 (0.0424)	3.63 (0.5560)	100.43 (7.2951)
	SCAD SVM	0.4017 (0.0980)	3.26 (1.3113)	60.70 (17.7863)
100	L_2 SVM	0.6597 (0.0154)	3.80 (0.401)	159.37 (4.8599)
	L_1 SVM	0.6377 (0.0302)	1.63 (1.0980)	104.10 (20.1876)
	SCAD SVM	0.5599 (0.0890)	1.93 (1.2576)	81.83 (15.9115)

methods. And the performance of the SCAD SVM is slightly superior to that of the other methods. The number in parenthesis is the sample standard deviation. The number of non-zero vector $\boldsymbol{\eta}_k$ is two. For all three methods the rate of correct model is one for 100 replications, because the number of coefficients is only four.

We add noise variables generated from the normal distribution with mean 0 and standard deviation $\sqrt{20}$. The number of noise variables is 20, 40, 60, 80 and 100. For example 20 noise variables, the model is perfect when the estimates for the only first two coefficients are non-zero and the estimates for the remaining coefficients are zero. Table 2 summarizes the results for three methods. The number in parenthesis represents the sample standard deviation. We consider the two terms of model complexity. The column labeled "Correct" presents the average number of correctly estimated zeros, and the column labeled "Incorrect" means the average number of coefficients erroneously set to zero in the same manner as done by Tibshirani (1996). In Table 2 the "Correct" and "Incorrect" terms should be 4 and 0, respectively under the ideal case. Table 2 shows that the correct classification rate of the SCAD SVM is the best among three methods for all cases. The average of number of the nonzero first two variables indicates the ability to choose the important variables. Due to the property of the L_2 cost function the L_2 SVM has the best performance. Increasing the number of noise variables decreases "Correct" terms of the L_1 SVM and the SCAD SVM. However the SCAD SVM is best in the ability not to choose the redundant variables. Regardless of the number of noise variables "Incorrect" term of the L_2 SVM is about 80%. It means that the L_2 SVM includes 80% redundant variables. For all cases the number of "Incorrect" term for the SCAD SVM is less than that for the L_1 SVM. It implies that the SCAD SVM is a best estimator in the sense of classification accuracy and model complexity.

4.2. Real data

We applied the developed multi-class SVM to three real data sets obtained from the UCI repository, `http://archive.ics.uci.edu/ml/datasets.html`. The characteristic of the data sets is wine

Table 3: The average testing error for real data sets

Method	wine	glass	waveform
L_2 SVM	0.0388 (0.0293)	0.5516 (0.0603)	0.0432 (0.0466)
L_1 SVM	0.0434 (0.0258)	0.5459 (0.0611)	0.1227 (0.0913)
SCAD SVM	0.0386 (0.0240)	0.5265 (0.0861)	0.0193 (0.0849)

data(13 input variables, 3 classes, 178 observations), glass(9 input variables, 6 classes, 214 observations) and waveform data(21 input variables, 3 classes, 300 observations). See the site for details.

We divide randomly the data sets into two parts, the training set and the test set. The training set is 2/3 of the total set and the remaining is the test set. Five-fold cross-validation within the training set is used to choose the tuning parameter λ. We did the experiment for 100 replications. Table 3 gives the average testing errors of the misclassification rate for L_2 SVM, L_1 SVM and the SCAD SVM. The number of parenthesis represents the sample standard deviation. For wine data we used the tuning parameter $0.042, 0.001, 0.001$ for the L_2 SVM, the L_1 SVM, the SCAD SVM, respectively. We set $\lambda = 0.001$ and 0.06 for all methods on glass data and waveform data, respectively. From Table 3 we can see that the SCAD SVM is slightly better than the L_2 SVM and L_1 SVM.

5. Concluding Remarks

In this paper we developed the multiclass SCAD SVM to improve the correct classification and the model complexity. When the data has redundant variables, numerical simulations shows that our method is superior to the L_2 SCAD and the L_1 SCAD. The method can be extended to the Huberized loss function (Li *et al.*, 2011). We can control the amount of robustness as the shape parameter of the Huber function.

References

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*, 82–90

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression, *Annals of Statistics*, **32**, 407–499.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties, *Journal of the American Statistical Association*, **96**, 1348–1360.

Hastie, T., Tibshirani, R. and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer.

Jung, K.-M. (2008). Robust statistical methods in variable selection, *Journal of the Korean Data Analysis Society*, **10**, 3057–3066.

Lee, Y., Lin, Y. and Wahba, G. (2004). Multicategory support vector machines, theory and applications to the classification of microarray data and satellite radiance data, *Journal of American Statistical Association*, **99**, 67–81.

Li, J., Jia, Y. and Li, W. (2011). Adaptive huberized support vector machine and its application to microarray classification, *Neural Computing & Applications*, **20**, 123–132.

Liu, Y., Zhang, H. H., Park, C. and Ahn, J. (2007). Support vector machines with adaptive L_q penalty, *Computational Statistics & Data Analysis*, **51**, 6380–6394.

Tibshirani, R. J. (1996). Regression shrinkage and selection via the LASSO, *Journal of the Royal Statistical Society, Series B*, **58**, 267–288.

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

Weston, J. and Watkins, C. (1999). Support vector machines for multi-class pattern recognition, In *Proceedings of the Seventh European Symposium on Artificial Neural Networks*.

Zhang, H. H., Ahn, J., Lin, X. and Park, C. (2006). Gene selection using support vector machines with non-convex penalty, *Bioinformatics*, **22**, 88–95.

Received June 26, 2012; Revised July 17, 2012; Accepted July 17, 2012