Weighted Support Vector Machines with the SCAD Penalty

Kang-Mo Jung^{1,a}

^aDepartment of Statistics and Computer Science, Kunsan National University

Abstract

Classification is an important research area as data can be easily obtained even if the number of predictors becomes huge. The support vector machine(SVM) is widely used to classify a subject into a predetermined group because it gives sound theoretical background and better performance than other methods in many applications. The SVM can be viewed as a penalized method with the hinge loss function and penalty functions. Instead of L_2 penalty function Fan and Li (2001) proposed the smoothly clipped absolute deviation(SCAD) satisfying good statistical properties. Despite the ability of SVMs, they have drawbacks of non-robustness when there are outliers in the data. We develop a robust SVM method using a weight function with the SCAD penalty function based on the local quadratic approximation. We compare the performance of the proposed SVM with the SVM using the L_1 and L_2 penalty functions.

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

1. Introduction

Classification is a crucial method in statistical pattern recognition and its importance has increased with the development of big data. There are many algorithms of classification such as linear discriminant analysis, logistic regression, k-nearest neighbor, boosting and neural networks (Hastie *et al.*, 2001). The support vector machine(SVM) was recently introduced by Vapnik (1995), which is an optimal margin classifier among linear classifiers. The SVM has been successfully adapted to real applications of engineering and bioinformatics (Zhang *et al.*, 2006).

The algorithm for the standard binary SVM can be written in terms of a penalized regression. It is usually consisted of a loss function and a penalized function. The standard SVM uses the convex hinge loss function and the L_2 penalized function. Tibshirani (1996) proposed the least absolute shrinkage and selection operator(LASSO) in regression model with the L_2 loss function and the L_1 penalized function that provides a sparse estimator. The LASSO yields the estimation of regression parameters and the selection of valuable predictors simultaneously. However, it is known that the solution of LASSO can be biased for large coefficients. Instead of the L_2 penalty function Fan and Li (2001) proposed a non-convex penalty function, the smoothly clipped absolute deviation(SCAD) penalty. Like the regression model, the standard SVM with the L_2 penalty function may consider a model with all predictors. However, a model with all predictors has a disadvantage when there are many noise predictors (Efron *et al.*, 2004). Therefore L_1 penalty (Bradley and Mangasarian, 1998) is

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 and NRF-2012R1A1A4 A01004594).

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

inclined to select a small number of predictors. The SVM with the SCAD penalty supplements the deficiencies of the L_1 penalty.

The L_1 and L_2 penalty functions are convex; consequently, the solution of the L_2 SVM and the L_1 SVM can be obtained by quadratic programming and linear programming methods, respectively. However, we cannot obtain the solution of the SVM with the SCAD penalty function using conventionally quadratic programming approaches since the SCAD function is not convex. Zhang *et al.* (2006) proposed a quadratic approximation method to the SCAD penalty in binary classifications. Successive quadratic algorithm(SQA) is a general method of Newton's method to solve unconstrained optimization problems.

In real classification problems the number of classes are usually more than two. The algorithms of multiclass SVM have been proposed to extend the algorithms of binary SVM. Among them Weston and Watkins (1999) proposed the "one-versus-rest" algorithm by constructing classifiers corresponding to the number of class; however, this algorithm provides poor performance when the data is dominated by only one class. To solve the multiclass classification problem Lee *et al.* (2004) proposed a simultaneous algorithm. Jung (2012) suggested a simultaneous multiclass SVM algorithm with the SCAD penalty function.

SVM is known to be sensitive to noisy training data despite its significant success. Outliers in the SVM can be defined to data lying far away from their own classes, because the unbounded hinge loss function affects strongly the SVM algorithm. Wu and Liu (2007) proposed a SVM methodology that involves truncating the unbounded hinge loss function. Liu and Shen (2006) developed a nonconvex loss function in ψ -learning for treating robust problems in the SVM. Wu and Liu (2013) used a SVM with a weight loss function and the L_2 penalty function.

In this paper we extend the binary SVM with the SCAD penalty function to the multiclass SVM which is robust to outliers. We propose a simultaneous multiclass SVM algorithm with a weight function that reduces the impact of outliers. The paper is organized as follows. In Section 2 we describe a binary SVM problem and preliminaries to suggest an algorithm. We briefly reviewed the L_1 , L_2 and SCAD penalty functions; in addition, we propose a weighted version of the binary SVM with the SCAD penalty function. Section 3 provides an algorithm of a weighted multiclass SVM with the SCAD penalty. As described in the previous paragraph the SCAD penalty function is nonconvex. Thus we use an approximation algorithm to solve the non-differentiable and non-convex objective function in the multiclass SVM with the SCAD penalty function. Section 4 provides the results of simulation and a real data set. It shows that the developed algorithm has a superior performance than L_2 , L_1 and SCAD SVM with respect to robustness. Section 5 provides some discussion and future work direction.

2. Binary SVM and Its Weighted SVM

For a binary classification problem, it is given a training sample $\{(\mathbf{x}_i, y_i), i = 1, \dots, n$, where $\mathbf{x}_i \in \mathcal{R}^d$ and $y_i \in \{-1, 1\}$ denote the input vector and the output label, respectively, and n is the sample size, d is the dimensionality of the input space.

2.1. Binary SVM

Consider the linear binary SVM which determines the decision function $f(\mathbf{x}) = \beta_0 + \beta^T \mathbf{x}$ so that $\hat{y} = \text{sign}(f(\mathbf{x}))$ can be used for classification. For a separable case there exists the hyperplane $\{\mathbf{x} | f(\mathbf{x}) = 0\}$

completely separating the two classes. SVM solves the optimization problem

$$\begin{aligned} \min_{\beta_0, \boldsymbol{\beta}} \quad & \frac{1}{2} \|\boldsymbol{\beta}\|^2 \\ \text{subject to} \quad & y_i f(\mathbf{x}_i) \geq 1, \quad \text{for } i = 1, \dots, n. \end{aligned}$$

For non-separable case the slack variables are considered and SVM solves a similar optimization problem.

Wahba (1998) showed that SVM paradigm could fit the regularization framework, where the objective function is composed of the loss function and the penalty function. Then SVM can be formulated as

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{n} \sum_{i=1}^n [1 - y_i f(\mathbf{x}_i)]_+ + \lambda \|\boldsymbol{\beta}\|^2, \quad (2.1)$$

where $(\cdot)_+$ is referred as the hinge loss function and λ controls the trade-off between training error and model complexity. Here the subscript $+$ indicates a positive part. Note that the hinge loss function is closely related to the misclassification loss function (Lee *et al.*, 2004). This convex optimization problem (2.1) using the Lagrange multipliers can be converted into the dual problem as

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle - \sum_{i=1}^n \alpha_i \\ \text{subject to} \quad & \sum_{i=1}^n y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq \frac{1}{\lambda}, \quad \forall i, \end{aligned} \quad (2.2)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product.

For the purpose of variable selection the latter term $\|\boldsymbol{\beta}\|^2$ of (2.1) can be replaced by $\sum_{j=1}^d I(\beta_j \neq 0)$, $\sum_{j=1}^d |\beta_j|$ and so on. The former L_0 penalty function has some drawbacks such that the discontinuity function makes unstable solutions and the hard-thresholding function shrinks small coefficients to zero. The L_1 penalty function can be widely used for variable selection in linear regression, which is known as LASSO (Tibshirani, 1996). The SVM with the L_1 penalty can be written by

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{n} \sum_{i=1}^n [1 - y_i f(\mathbf{x}_i)]_+ + \lambda \sum_{j=1}^n |\beta_j|. \quad (2.3)$$

As in linear regression the solutions of L_1 regression are sparse and it provides useful information on variable selection, the L_1 SVM (Zhu *et al.*, 2003) may have some advantage especially when there are redundant noise input variables. LASSO estimates can be biased for large coefficients since larger penalties are imposed on larger coefficients.

Fan and Li (2001) proposed the SCAD penalty to overcome biasedness of the L_1 penalty. Using SCAD function Zhang *et al.* (2006) proposed the SCAD SVM to perform variable selection in classification problems. The SCAD function is

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

where $a > 2$ and $\lambda > 0$ are tuning parameters. The standard SVM (2.1) becomes the SCAD SVM solving the optimization problem

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{n} \sum_{i=1}^n [1 - y_i f(\mathbf{x}_i)]_+ + \sum_{j=1}^d p_\lambda(|\beta_j|). \quad (2.5)$$

The parameter λ in the objective function (2.5) regulates the trade-off between data fitting and model parsimony. See Jung (2008) in linear regression model. The parameter a is set by $a = 3.7$, because Fan and Li (2001) showed that the Bayes risks are not sensitive to the choice of a and $a = 3.7$ showed good results for many problems. The objective function (2.5) is not convex and cannot be easily solved, even though the standard SVM in (2.1) and L_1 SVM in (2.3) can be solved by quadratic programming and linear programming methods, respectively. Zhang *et al.* (2006) proposed a successive quadratic algorithm (SQA) which is an approximation method of Newton's method that converts the objective function (2.5) into a series of linear equations.

2.2. Weighted SVM

The solution in (2.2) has a sparse form in which most of α_i become zero and only data points with $\alpha_i > 0$ can have an impact on the SVM classifier, namely the support vectors. Among support vectors the misclassified points lying far from the hyperplane significantly impact the classifier (in linear regression they are called outliers). To reduce the influence of outliers Wu and Liu (2007) used a truncated hinge loss function.

In linear regression one of the robust estimates is a weighted estimate obtained by a reduction of the impact of large residuals (Jung, 2011). The idea was adapted to standard SVM (Wu and Liu, 2013) as

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{n} \sum_{i=1}^n w_i [1 - y_i f(\mathbf{x}_i)]_+ + \lambda \|\boldsymbol{\beta}\|^2. \quad (2.6)$$

In addition, the dual form of (2.6) can be written as the same objective function of (2.2) except the constraint $0 \leq \alpha_i \leq w_i/\lambda$ for all i . This dual problem can be solved by a quadratic programming solver. They set the weight $w_i(\mathbf{x}_i) = 1/(1 + |\hat{f}_{SSVM}(\mathbf{x}_i)|)$ for $i = 1, \dots, n$ and trained the weighted SVM using this weight, where $\hat{f}_{SSVM}(\cdot)$ is the solution of (2.1). The function $w_i(u)[1 - y_i f(u)]_+ = \max\{0, 1 - u\}/(1 + |u|)$ becomes same as the 0–1 loss except $[0, 1]$. However, the weighted hinge loss function is continuous. In case $w_i = 1$ for all training data the weighted SVM reduces to standard SVM (2.1).

In this paper, we adapt the weight function for a robust SVM to SCAD penalty as

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{n} \sum_{i=1}^n w_i [1 - y_i f(\mathbf{x}_i)]_+ + \sum_{j=1}^d p_\lambda(|\beta_j|). \quad (2.7)$$

Denote the solution of (2.5) by $\hat{f}_{SSVM}(\cdot)$. Similar to (2.6), the weight w_i of (2.7) is defined to be a reciprocal proportion to the size of $|\hat{f}_{SSVM}(u)|$ since data points corresponding to be larger $|\hat{f}_{SSVM}(\mathbf{x}_i)|$ can be outliers. The larger $|\hat{f}_{SSVM}(\mathbf{x}_i)|$ for correctly classified observations have not influence on the classifier; however, the larger $|\hat{f}_{SSVM}(\mathbf{x}_i)|$ for misclassified observations whose $y_i \hat{f}_{SSVM}(\mathbf{x}_i)$ are negative have influence on the classifier. This is because the former points cannot be support vectors and the latter points are support vectors.

It is difficult to solve (2.7) because neither the first nor the second term is convex. The non-convexity of the first term can be settled by transforming the primal problem into the dual form as

in (2.6). Many researchers have been devoted to the non-convexity optimization of the second term. Kim *et al.* (2008) proposed an optimization algorithm based on the concave-convex procedure in linear regression. Park *et al.* (2012) showed the oracle property of the penalized SVM with the SCAD penalty and the simulation results that used a linear quadratic approximation algorithm (Fan and Li, 2001). Recently Fan *et al.* (2013) proposed an algorithm to obtain the locally oracle solution of the penalized regression with the SCAD penalty function under mild conditions. They used the one-step local linear approximation which Zou and Li (2008) proposed to solve the non-concave penalized likelihood models. We used the LQA algorithm in this article; however, the optimization problem can be converted into a series of quadratic programming problems that use local linear approximation.

The weighted SCAD SVM procedure consisted of two steps. The first step is to solve (2.5) for all training data points, and the second step is to solve (2.7) after obtaining the weights for all data points. Wu and Liu (2013) recommended one-step weighted iteration because the iterative solution cannot guarantee convergence if original hinge loss based weights are assigned instead of the one-step solution.

Now we first obtain the solution of SCAD SVM (2.5). We use the quadratic approximation of the hinge function u_+ and the SCAD function $p_\lambda(u)$. We have $u_+ = (u + |u|)/2$ and the approximation $|u| \approx (1/2)u^2/|u_0| + (1/2)|u_0|$ for nonzero u_0 near u ; therefore, we have $u_+ \approx (1/4)u^2/|u_0| + (1/2)u + (1/4)|u_0|$. We have the approximation of the SCAD function

$$p_\lambda(|u|) \approx p_\lambda(|u_0|) + \frac{p'_\lambda(|u_0|)}{2|u_0|} (u^2 - u_0^2),$$

where

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

The convergence of the approximate function can be proven by a function that has the same derivative function at u_0 . Assume that an initial value $(\beta_{00}, \beta_0^T)^T$ is given. Define the augmented $n \times (d+1)$ matrix $\tilde{X}$ whose i^{th} row vector is $\tilde{\mathbf{x}}_i = (1, \mathbf{x}_i^T)$. Let $\mathbf{y} = (y_1, \dots, y_n)^T$, $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_n)^T$ with $\varepsilon_i = y_i - (\beta_{00} + \beta_0^T \mathbf{x}_i)$. Define $\mathbf{r} = (y_1/|\varepsilon_1|, \dots, y_n/|\varepsilon_n|)^T$, $D_1 = (1/2n)\text{diag}(1/|\varepsilon_1|, \dots, 1/|\varepsilon_n|)$, $L_1 = (1/2n)(\mathbf{y} + \mathbf{r})^T \tilde{X}$ and $D_2 = \text{diag}(0, p'_\lambda(|\beta_{10}|)/|\beta_{10}|, \dots, p'_\lambda(|\beta_{d0}|)/|\beta_{d0}|)$. Thus the objective function (2.5) becomes the approximate optimization problem (Zhang *et al.*, 2006)

$$\frac{1}{2} \boldsymbol{\eta}_1^T Q_1 \boldsymbol{\eta}_1 - L_1 \boldsymbol{\eta}_1, \quad (2.8)$$

where $\boldsymbol{\eta}_1 = (\beta_0, \beta^T)^T$ and $Q_1 = D_1 \tilde{X} + D_2$ and $L_1 = (1/2n)(\mathbf{y} + \mathbf{r})^T \tilde{X}$.

Let the solution of (2.8) be $\hat{\boldsymbol{\eta}}_{1,SSVM}$. Then define the weight $w_i = 1/(1 + |\hat{f}_{SSVM}(\mathbf{x}_i)|)$, where $\hat{f}_{SSVM}(\mathbf{x}_i) = \hat{\boldsymbol{\eta}}_{1,SSVM}^T \mathbf{x}_i$. Similar to (2.8), we obtain the approximation of the objective function (2.7)

$$\frac{1}{2} \boldsymbol{\eta}_1^T Q_1^w \boldsymbol{\eta}_1 - L_1^w \boldsymbol{\eta}_1, \quad (2.9)$$

where $Q_1^w = D_1 W \tilde{X} + D_2$ and $L_1^w = (1/2n)(\mathbf{y} + \mathbf{r})^T W \tilde{X}$. Here $W = \text{diag}(w_1, \dots, w_n)$ is the diagonal matrix whose element is the weight for each training datum. Let the solution of (2.9) be $\hat{\boldsymbol{\eta}}_{1,SSVM}^w$.

The proposed algorithm can be summarized as:

Step 1 Set the initial values β_0, β_0 as the linear discriminant function.

Step 2 Solve the linear system (2.8) until convergence.

Step 3 Set $w_i, i = 1, \dots, n$. Solve the linear system (2.9) until convergence.

In Step 2 the algorithm will stop when the difference between $\hat{\eta}_1^{(l)}$ for the l -th iteration and $\hat{\eta}_1^{(l+1)}$ is less than the given tolerance. In Step 3 the tolerance is applied in the same manner. In this paper we use the tolerance 10^{-3} for the squared sum of difference.

3. Weighted Multiclass SVM with SCAD Penalty

Consider a K -class problem with a training set $\{\mathbf{x}_i, y_i\}_{i=1}^n$, where $\mathbf{x}_i \in \mathcal{R}^d$ is the input vector and $y_i \in \{1, 2, \dots, K\}$ represents its class label. It needs to find a K -dimensional decision function $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_K(\mathbf{x}))$ 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 objective function (Lee *et al.*, 2004)

$$\frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K I(y_i \neq k) (f_k(\mathbf{x}_i) + 1)_+ + \sum_{k=1}^K p_\lambda(f_k), \quad (3.1)$$

where $I(y_i \neq k)$ is the indicator function having zero if $y_i = k$ and one otherwise. In this paper we consider the linear decision function $f_k(\mathbf{x}_i) = \beta_{0k} + \beta_k^T \mathbf{x}_i$ and the SCAD penalty function. Then (3.1) becomes

$$\begin{aligned} & \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K I(y_i \neq k) (\beta_k^T \mathbf{x}_i + \beta_{0k} + 1)_+ + \sum_{j=1}^d \sum_{k=1}^K p_\lambda(|\beta_{jk}|), \\ \text{s.t. } & \sum_{k=1}^K \beta_{0k} = 0, \quad \sum_{k=1}^K \beta_{jk} = 0, \quad \text{for } j = 1, \dots, d. \end{aligned} \quad (3.2)$$

The classification rule induced by $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_K(\mathbf{x}))^T$ naturally becomes

$$\phi(\mathbf{x}) = \arg \max_j f_j(\mathbf{x}). \quad (3.3)$$

Similar to (2.8), we obtain the approximation (3.4) of the objective function. See Jung (2012) for details. 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 $\boldsymbol{\eta}^T = (\boldsymbol{\eta}_1^T, \dots, \boldsymbol{\eta}_{K-1}^T)$, where $\boldsymbol{\eta}_k = (\beta_{0k}, \beta_{1k}, \dots, \beta_{dk})^T$. Then we have

$$\frac{1}{2} \sum_{k=1}^{K-1} \sum_{l=1}^{K-1} \boldsymbol{\eta}_k^T \{ (Q_{B_2} + Q_{C_2}) + I(k=l) (Q_{A_2,k} + Q_{C_1,k}) \} \boldsymbol{\eta}_l + \sum_{k=1}^{K-1} \boldsymbol{\eta}_k^T (L_{A_2,k} + L_{A_1,k} + L_{B_2} + L_{B_1}), \quad (3.4)$$

where

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

$$\begin{aligned}
Q_{C_2} &= \text{diag} \left(0, \frac{p'_\lambda \left(\left| \sum_{m=1}^{K-1} \beta_{1m}^0 \right| \right)}{\left| \sum_{m=1}^{K-1} \beta_{1m}^0 \right|}, \dots, \frac{p'_\lambda \left(\left| \sum_{m=1}^{K-1} \beta_{dm}^0 \right| \right)}{\left| \sum_{m=1}^{K-1} \beta_{dm}^0 \right|} \right), & Q_{C_1,k} &= \text{diag} \left(0, \frac{p'_\lambda \left(\left| \beta_{1k}^0 \right| \right)}{\left| \beta_{1k}^0 \right|}, \dots, \frac{p'_\lambda \left(\left| \beta_{dk}^0 \right| \right)}{\left| \beta_{dk}^0 \right|} \right), \\
L_{A_2,k} &= \frac{1}{2n} \sum_{i=1}^n \frac{a_{ik}}{\left| 1 + \boldsymbol{\eta}_k^{0T} \tilde{\mathbf{x}}_i \right|} \tilde{\mathbf{x}}_i, & L_{A_1,k} &= \frac{1}{2n} \sum_{i=1}^n a_{ik} \tilde{\mathbf{x}}_i, \\
L_{B_2} &= -\frac{1}{2n} \sum_{i=1}^n \frac{a_{iK}}{\left| 1 - \sum_{k=1}^{K-1} \boldsymbol{\eta}_k^{0T} \tilde{\mathbf{x}}_i \right|} \tilde{\mathbf{x}}_i, & L_{B_1} &= -\frac{1}{2n} \sum_{i=1}^n a_{iK} \tilde{\mathbf{x}}_i,
\end{aligned}$$

where $\boldsymbol{\eta}_k^0$ denotes an initial value of $\boldsymbol{\eta}_k$. Then (3.4) can be simplified by

$$\frac{1}{2} \boldsymbol{\eta}^T \mathbf{Q} \boldsymbol{\eta} + \boldsymbol{\eta}^T \mathbf{L}, \quad (3.5)$$

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 k -th 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 k -th vector $L_{A_2,k} + L_{A_1,k} + L_{B_2} + L_{B_1}$ of length $(d+1)$. Since (3.5) is quadratic with respect to $\boldsymbol{\eta}$, the optimal solution of the objective function (3.5) is the solution of the linear equations

$$\mathbf{Q} \boldsymbol{\eta} + \mathbf{L} = \mathbf{0}. \quad (3.6)$$

Now we consider a weighted version of the objective function (3.2) given by

$$\frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K w_i I(y_i \neq k) \left(\boldsymbol{\beta}_k^T \mathbf{x}_i + \beta_{0k} + 1 \right)_+ + \sum_{j=1}^d \sum_{k=1}^K p_\lambda \left(\left| \beta_{jk} \right| \right), \quad (3.7)$$

where the weight w_i is defined by $\min\{f_y(\mathbf{x}) - f_j(\mathbf{x}), j \neq y\}$ (Wu and Liu, 2013). The weight weakens influence of misclassified observations far from the decision boundaries which can be support vectors. Then the optimal solution of (3.7) can be obtained by solving

$$\mathbf{Q}^w \boldsymbol{\eta} + \mathbf{L}^w = \mathbf{0}, \quad (3.8)$$

where $\mathbf{Q}^w$ and $\mathbf{L}^w$ can be defined similarly to $\mathbf{Q}$ and $\mathbf{L}$ of (3.6) except that the element a_{ik} should be replaced by $w_i a_{ik}$. When $w_i = 1$ for all i , then the weighted objective function (3.8) becomes the SCAD SVM (3.6).

We summarize an algorithm for the weighted multiclass SVM with the SCAD penalty function by the following iterative steps:

Step 1. Set the initial values $\beta_0, \boldsymbol{\beta}_0$ as the linear discriminant function.

Step 2. Solve the linear system (3.6) until convergence.

Step 3. Set $w_i, i = 1, \dots, n; k = 1, \dots, K$. Solve the linear system (3.8) until convergence.

4. Simulation and Real Data

This section demonstrates simulations to show the robustness of the method proposed in Sections 2 and 3. We numerically compare the proposed method with the L_2 SVM, the L_1 SVM and the SCAD SVM.

Table 1: Simulation results for binary class

Contamination Rate	L_2 SVM	L_1 SVM	SCAD SVM	Weighted SCAD SVM
5%	0.102 (0.032)	0.090 (0.030)	0.091 (0.027)	0.078 (0.017)
10%	0.156 (0.031)	0.149 (0.031)	0.150 (0.030)	0.133 (0.021)
20%	0.263 (0.034)	0.254 (0.036)	0.253 (0.034)	0.245 (0.026)
30%	0.376 (0.045)	0.367 (0.048)	0.366 (0.042)	0.357 (0.042)

Table 2: Simulation results for three classes

Contamination Rate	L_2 SVM	L_1 SVM	SCAD SVM	Weighted SCAD SVM
5%	0.269 (0.050)	0.268 (0.050)	0.269 (0.079)	0.234 (0.012)
10%	0.303 (0.050)	0.300 (0.053)	0.292 (0.046)	0.269 (0.009)
20%	0.388 (0.054)	0.384 (0.052)	0.383 (0.058)	0.345 (0.014)
30%	0.465 (0.047)	0.465 (0.048)	0.462 (0.049)	0.420 (0.015)

Table 3: The average misclassification rate for liver data

Contamination Rate	L_2 SVM	L_1 SVM	SCAD SVM	Weighted SCAD SVM
0%	0.354 (0.055)	0.345 (0.057)	0.339 (0.052)	0.329 (0.048)
3%	0.374 (0.063)	0.377 (0.068)	0.358 (0.057)	0.351 (0.056)
6%	0.403 (0.066)	0.401 (0.069)	0.382 (0.065)	0.375 (0.061)
10%	0.409 (0.066)	0.410 (0.068)	0.393 (0.063)	0.390 (0.064)

4.1. Simulation

We set the sample sizes of the training data, the tuning data and the test data as 100, 1,000, 10,000, respectively. The tuning sets identify the tuning parameter λ . To choose an appropriate tuning parameter λ we search over a grid $\log_2(\lambda) = -10, -9, \dots, 1, 2$. The optimum λ is found by minimizing the misclassification rate. When ties occurs, the maximum value of λ s would be selected for general learning capability. The tuning parameter of the weighted SCAD SVM is set by the same parameter of the SCAD SVM.

First we consider a binary classification problem with the two dimensional data set generated from the uniform distribution on the unit disc $\{(x_1, x_2) | x_1^2 + x_2^2 < 1\}$. The corresponding label y is assigned to the sign of x_2 (Wu and Liu, 2007). To check robustness we contaminate the data by changing the sign of the label for randomly cases as many as given perc(= 5%, 10%, 20%, 30%). The contaminated observations may be potential outliers. Table summarizes the simulation results. The number in the table is the mean of the misclassification rate. The number in parenthesis is the sample standard deviation. It implies that the weighted SCAD SVM is superior to other penalties in the binary classification. The misclassification rate increases relative to the contamination rate; however, the proposed method is the best among the four methods for the mean and standard deviation of misclassification rates. In addition, our method takes only few computation steps rather than the SCAD SVM.

We consider a multiclass example with $K = 3$ and $d = 2$. The data $\mathbf{x}$ is generated from the bivariate normal distribution $N(\boldsymbol{\mu}_k, \mathbf{I}_2)$ 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 $\sigma^2 = 2$ (Jung, 2012). For contamination of the data we flipped the response to one of other response values with perc/2 probability. For example, if the data of response value 1 is contaminated up to 5% the corresponding response is replaced by one of response values 2, 3 with probability 2.25%.

After learning the training data with the sample size 100 we compute the mean and standard deviation of the misclassification rate for the test data with the sample size 10,000. Table 2 summarizes the results for 100 replications. We can see that the performance are comparable with three methods; in addition, the performance of the weighted SCAD SVM is slightly superior to other methods.

4.2. Real data

We applied a weighted SVM to a real data obtained from the UCI repository. The liver data set is consisted of 345 observations of 6 input variables with 2 classes. The first 5 variables are the measurements of blood tests believed to be sensitive to liver disorders that might arise from excessive alcohol consumption. The sixth input variable is the drinks number of half-pint equivalents of alcoholic beverages drunk per day.

We divide randomly the data sets into three parts, the training set, the tuning set and the test set. The number of observations for each set training set is 115. For the contamination of the data we flipped the response with 0%, 3%, 6% and 10%. We conducted the experiment for 100 replications. Table 3 provides the average misclassification rate for L_2 SVM, L_1 SVM, the SCAD SVM and the weighted SCAD SVM. Table 3 shows that the weighted SCAD SVM is better than other methods in the view of robustness to outlying observations.

5. Concluding Remarks

In this paper we developed a weighted multiclass SVM with the SCAD penalty function to improve the robustness of SVM. When the data has outliers, numerical simulations show that the proposed method is more robust than the L_2 SVM, L_1 SVM and the SCAD SVM. For further study we can control the weights for nonlinear SVM algorithms with the SCAD penalty function.

References

- Bradley, P. S. and Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines, *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.
- Fan, J., Xue, L. and Zou, H. (2013). Strong oracle optimality of folded concave penalized estimation, Unpublished manuscript.
- 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.
- Jung, K.-M. (2011). Weighted least absolute deviation lasso estimator, *Communications of the Korean Statistical Society*, **18**, 733–739.
- Jung, K.-M. (2012). Multiclass support vector machines with SCAD, *Communications of the Korean Statistical Society*, **19**, 655–662.
- Kim, Y., Choi, H. and Oh, H.-S. (2008). Smoothly clipped absolute deviation on high dimensions, *Journal of American Statistical Association*, **103**, 1665–1673.
- 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.
- Liu, Y. and Shen, X. (2006). Multicategory ψ -Learning, *Journal of American Statistical Association*, **101**, 500–509.

- Park, C., Kim, K.-R., Myung, R. and Koo, J.-Y. (2012). Oracle properties of SCAD-penalized support vector machine, *Journal of Statistical Planning and Inference*, **142**, 2257–2270.
- 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.
- Wahba, G. (1998). Support vector machines, reproducing kernel Hilbert spaces, and randomized GACV, In *Advances in Kernel Methods: Support Vector Learning*, eds. B. Schölkopf, C. J. C. Burges, and A. J. Smola, Cambridge, MA:MIT Press, 125–143.
- Weston, J. and Watkins, C. (1999). Support vector machines for multi-class pattern recognition, *Proceedings of the Seventh European Symposium on Artificial Neural Networks*.
- Wu, Y. and Liu, Y. (2007). Robust truncated-hinge-loss support vector machines. *Journal of the American Statistical Association*, **102**, 974–983.
- Wu, Y. and Liu, Y. (2013). Adaptively weighted large margin classifiers. *Journal of Computational and Graphical Statistics*, **22**, 416–432.
- 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.
- Zhu, J., Rosset, S., Hastie, T. and Tibshirani, R. (2003). 1-norm support vector machines, In *Advances in Neural Information Processing Systems 16*, eds, S. Thrun, L. Saul and B. Schölkopf, Cambridge, MA:MIT Press, 49–56.
- Zou, H. and Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models (with discussion), *The Annals of Statistics*, **36**, 1509–1566.

Received October 1, 2013; Revised October 17, 2013; Accepted October 31, 2013