Robust Algorithm for Multiclass Weighted Support Vector Machine

Kang-Mo Jung*

*Professor, Department of Statistics and Computer Science, Kunsan National University, Kunsan, Chonbuk, SOUTH KOREA.
E-Mail: kmjung{at}kunsan{dot}ac{dot}kr

***Abstract*—**Support Vector Machine (SVM) has shown better performance than other methods in real world classification applications, because it gives mathematical tractability and geometrical interpretation. However, the standard SVM can suffer from outliers in either the response or the predictor space. SVM can be viewed as a penalized method with the hinge loss function and penalty functions. Instead of L_2 penalty function we considered the Smoothly Clipped Absolute Deviation (SCAD) function, because it has two advantages, sparse learning and unbiasedness. However, it has drawbacks of non-robustness when there are outliers in the data. We develop a robust algorithm for SVM using a weight function of the SCAD with a Local Linear Approximation (LLA) method and a Local Quadratic Approximation (LQA). We compare the performance of the proposed algorithm with the standard SVM using L_1 and L_2 penalty functions.

***Keywords*—**Local Linear Approximation; Local Quadratic Approximation; Penalized Function; Robust; Smoothly Clipped Absolute Deviation; Support Vector Machine; Weight.

***Abbreviations*—**Least Absolute Shrinkage and Selection Operator (LASSO); Local Linear Approximation (LLA); Local Quadratic Approximation (LQA); Smoothly Clipped Absolute Deviation (SCAD); Support Vector Machine (SVM).

I. INTRODUCTION

CLASSIFICATION is an important method in pattern recognition or discrimination and recently it sheds new light on big data technologies. There are many algorithms for classification such as linear discrimination function, logistic regression function, k-nearest neighbour, boosting and neural networks [Hastie et al., 2001]. Vapnik (1995) introduced Support Vector Machine (SVM) which is an optimal margin classifier among linear classifiers. SVM has shown better performance than other methods in real applications of engineering and bioinformatics, because it gives mathematical tractability and geometrical interpretation.

Let x denote a feature vector. The class labels, y, are coded as $\{-1, 1\}$. For a given training data set $\{ x_i , y_i\}, i = 1,2, \cdots , n$, the SVM can be written by a penalized hinge loss function

$$min_{\beta,\beta_0} \sum_{i=1}^{n} [1 - y_i (x_i^T \beta + \beta_0)]_+ + \lambda ||\beta \quad ^2 \qquad (1)$$

where the subscript + means the positive part, for example $x_+ = \max(x, 0)$. The SVM classifier becomes the sign of function $\hat{\beta}_0 + x^T \hat{\beta}$ for a given feature vector x. Equation (1) can be interpreted as a penalized regression with the hinge loss function and L_2 penalty function. It is well known that L_2 penalty function may consider a model with all components of feature vector. For the sparseness of the solution Tibshirani (1996) proposed the Least Absolute Shrinkage and Selection Operator (LASSO) in linear regression with L_1 penalty function instead of L_2 function. However, the LASSO estimates can be biased for large coefficients since larger penalties are imposed on larger coefficients. Fan & Li (2001) proposed a non-convex penalty function, the Smoothly Clipped Absolute Deviation (SCAD) penalty function which gives unbiased estimates for even large coefficients.

In real classification problems the binary SVM can be useless. Lee et al., (2004) proposed a simultaneous algorithm for multiclass classification problems and Jung (2012) suggested a simultaneous multiclass SVM algorithm with the SCAD penalty function. The SCAD multiclass SVM conducts variable selection and classification simultaneously, and it gives a compact classifier with high accuracy. Because gene selection treats the expression levels of thousands of genes simultaneously in one single experiment, the SCAD SVM can be very useful in bioinformatics [Zhang et al., 2006].

SVM is known to be sensitive to noisy training data, because the loss function of (1) is not bounded. Outliers in the SVM can be defined to data lying far away from their own classes, because the unbounded hinge loss function affects strongly SVM algorithm. In this paper we consider a robust algorithm for SVM using the weight function that

makes the loss function be bounded. It can yield higher correct classification rate than ordinal SVM in many problems. Furthermore the number of the support vectors for the proposed SVM is less than that of the SVM, because our algorithm is not affected by outliers and the proposed method does not retain the support vectors of outliers. That is, the set of support vectors for our algorithm is a subset of the set of support vectors for the SVM [Wu & Liu, 2007]. Hence, it does not require much computation time and it can give easy interpretation of the input variables.

The paper is organized as follows. Section 2 describes the previously related works. Section 3 provides our proposed algorithm of a weighted SVM with the SCAD penalty. Since the SCAD function is not convex, we use an approximation algorithm to solve the non-differentiable and non-convex objective function in SVM with the SCAD penalty. The SVM can solve linear programming problems or quadratic programming problems. It requires general software to obtain the solution. We provide two results, the solver of linear programming and linear equation system. Section 4 illustrates the results of simulation and a real data set. It shows that the proposed algorithm has superior to other methods from the view of robustness.

II. Related Works

For variable selection the standard SVM (1) can be converted to

$$\min_{\beta,\beta_0} \sum_{i=1}^{n} [1 - y_i (x_i^T \beta + \beta_0)]_+ + \lambda \sum_{j=1}^{d} |\beta_j| \qquad (2)$$

where d is the dimension of the input space [Bradley & Mangasarian, 1998]. The L_1 penalty function is well known as the LASSO and widely used for variable selection and regression estimation simultaneously in linear regression models. The solution of the L_1 SVM in (2) can be obtained by solving a linear programming problem. Zhu et al., (2003) studied a solution path for the L_1 SVM. Fung & Mangasarian (2004) proposed a fast Newton algorithm to solve the dual problem for the L_1 SVM. Wang & Shen (2007) developed a L_1-norm multiclass SVM and investigated its feasibility in classification and variable selection. The L_1 SVM has some advantage especially when there are redundant noise input variables. The L_1 SVM can be extended to the SCAD SVM which solves the optimization problem

$$\min_{\beta,\beta_0} \sum_{i=1}^{n} [1 - y_i (x_i^T \beta + \beta_0)]_+ + \sum_{j=1}^{d} p_\lambda(|\beta_j|) \qquad (3)$$

where

$$p_\lambda'(|\beta|) = \begin{cases} \lambda, & if\ 0 \le |\beta| < \lambda \\ \dfrac{a\lambda - |\beta|}{a-1}, & if\ \lambda \le |\beta| < a\lambda \\ 0, & if\ |\beta| > a\lambda \end{cases}$$

where $a > 2$ and $\lambda > 0$ are tuning parameters. The parameter λ in the objective function (3) regulates the trade-off between data fitting and model parsimony. The parameter a is set as $a = 3.7$, because Fan & 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.

To reduce the influence of outliers Wu & Liu (2007) used a truncated hinge loss function as a method of a bounded loss function. They proposed to apply the difference convex algorithm to solve the non-convex problem through a sequence of convex sub-problems in (1), because the truncated hinge loss function is furthermore not convex. Liu & Shen (2006) developed a non-convex loss function in ψ-learning to treat robust problems in SVM. They generalized binary ψ-learning to the multiclass case. Wu & Liu (2013) used a SVM with a weight loss function of L_2 penalty, which gives larger weights for points closer to the boundaries and smaller weights for points farther away.

III. Methods

3.1. Binary Weighted SVM

The solution in (1) is sparse in which most of coefficients become zeroes and only support vectors can have an impact on the SVM classifier. Among support vectors the misclassified points lying far from the hyper-plane significantly impact the classifier, because the points are misclassified and the distance from the boundary is larger than others.

In linear regression one of the robust estimates is a weighted version obtained by a reduction of the impact of large residuals. Then the points having large residuals do not impact the regression coefficients. The idea was adapted to (1) [Wu & Liu, 2013]

$$min_{\beta,\beta_0} \frac{1}{n} \sum_{i=1}^{n} w_i [1 - y_i (x_i^T \beta + \beta_0)]_+ + \lambda \left|\left|\beta\right|\right|^2 \qquad (4)$$

In addition the dual form of (4) can be obtained by a quadratic programming solver. They set the weight $w_i(x_i) = 1/(1 + |\hat{f}_{SVM}(x_i)|)$ for $i = 1, \cdots, n$ where $\hat{f}_{SVM}(\cdot)$ is the solution of (1). The function $w_i(u)[1 - y_i f(u)]_+$ 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 the data the weighted SVM reduces to the standard SVM (1).

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

$$min_{\beta,\beta_0} \frac{1}{n} \sum_{i=1}^{n} w_i [1 - y_i (x_i^T \beta + \beta_0)]_+ + \sum_{j=1}^{d} p_\lambda'(|\beta_j^0|)|\beta_j| \qquad (5)$$

where $p_\lambda'(\cdot)$ is defined in (3), $\sum_{j=1}^{d} p_\lambda'(|\beta_j^0|)|\beta_j|$ is the linearized SCAD penalty function [Zou & Li, 2008] and β^0 is an initial estimator. Unlike the objective function (3), the objective function defined in (5) is convex in β. Zou & Li (2008) proposed a new unified algorithm based on Local Linear Approximation (LLA) to the penalty function

$$p_\lambda(|\beta_j|) \approx p_\lambda(|\beta_j^0|) + p_\lambda'(|\beta_j^0|)(|\beta_j| - |\beta_j^0|), \quad \text{for } \beta_j \approx \beta_j^0 \tag{6}$$

Replacing the penalty function in (3) by Equation (6) gives the objective function (5). Updating the solution of (5) until the solution converges. The weight w_i can be small for the mis-classified data and it can be near to one for the well-classified data.

The weighted SCAD SVM procedure is consisted of two steps. The first step is to solve (5) with $w_i = 1, i = 1, \cdots, n$ for all training data points, and the second step is to solve (5) with the weights $w_i = \frac{1}{1+|\hat{f}_{SSVM}(x_i)|}$, where $\hat{f}_{SSVM}(x_i) = \hat{\beta}_0 + x_i^T\hat{\beta}$ and $\hat{\beta}_0, \hat{\beta}$ are the solution of (5) for non-weighted case $w_i = 1$. Wu & Liu (2013) recommended one-step weighted iteration because the iterative solution cannot guarantee convergence if the weights based on the original hinge loss function is assigned instead of the one-step solution.

Now we obtain the solution of (5). The equation (5) can be sufficiently solved by standard Linear Programming (LP) software. To derive the LP formulation of (5), we introduce a set of slack variables

$$\xi_i = w_i \left[1 - y_i \left(\sum_{j=1}^{d} x_{ij}(\beta_j^+ - \beta_j^-) + \beta_0^+ - \beta_0^- \right) \right]_+, \quad i = 1,2,\cdots,n,$$

and we write $\beta_j = \beta_j^+ - \beta_j^-$ where β_j^+ and β_j^- denote the positive and negative parts of β_j, respectively. Then it is straightforward to show that (5) is equivalent to

$$min_{\beta_j^+, \beta_j^-, \beta_0^+, \beta_0^-} \frac{1}{n}\sum_{i=1}^{n}\xi_i + \sum_{j=1}^{d} p_\lambda'(|\beta_j^0|)(\beta_j^+ + \beta_j^-) \tag{7}$$

subject to for all $1 \le i \le n$ and $j = 0,1,\cdots,d$

$$y_i\left(\sum_{j=1}^{d} x_{ij}(\beta_j^+ - \beta_j^-) + \beta_0^+ - \beta_0^-\right) \ge 1 - \xi_i/w_i,$$
$$\xi_i \ge 0, \beta_j^+ \ge 0, \beta_j^- \ge 0.$$

The solution of (5) can be obtained by the Linear Quadratic Approximation (LQA) method. The quadratic approximation of the hinge function $u_+ = \frac{1}{2}(u + |u|)$ and the absolute function $|u| \approx \frac{u^2}{2|u_0|} + \frac{1}{2|u_0|}$ for nonzero u_0 near u gives $u_+ \approx \frac{u^2}{4|u_0|} + \frac{u}{2} + \frac{|u_0|}{4}$. The second term of (5) can be rewritten by $\sum_{j=1}^{d} p_\lambda'(|\beta_j^0|)/2|\beta_j^0|\beta_j^2$. Assume that an initial value $(\beta_0^0, \beta^{0T})^T$ is given. Define the augmented $n \times (d+1)$ matrix $\tilde{X}$ with the ith row vector $\tilde{x}_i = (1, x_i^T)$ and $y = (y_1, \cdots, y_n)^T$, $\varepsilon = (\varepsilon_1, \cdots, \varepsilon_n)^T$ where $\varepsilon_i = y_i - (\beta_0^0 + \beta^{0T}x_i)$ for $i = 1, \cdots, n$. Define $r = (\frac{y_1}{|\varepsilon_1|}, \cdots, \frac{y_n}{|\varepsilon_n|})^T, D_1 = \frac{1}{2n} diag\,(\frac{1}{|\varepsilon_1|}, \cdots, \frac{1}{|\varepsilon_n|})$, $L_1 = \frac{1}{2n}(y + r)^T\tilde{X})$ and $D_2 = diag(0, \frac{p_\lambda' \beta_1^0}{|\beta_1^0|}, \cdots, \frac{p_\lambda' \beta_d^0}{|\beta_d^0|})$. Therefore the objective function (5) becomes the approximate optimization problem [Jung, 2013].

$$\frac{1}{2}\eta_1^T Q_1 \eta_1 - L_1\eta_1 \tag{8}$$

where $\eta_1 = (\beta_0, \beta^T)^T$, $Q_1 = D_1\tilde{X} + D_2$ and $L_1 = \frac{1}{2n}(y + r)^T\tilde{X}$. Similar to (8), we obtain the approximation of the objective function with weighted case

$$\frac{1}{2}\eta_1^T Q_{1w}\eta_1 - L_{1w}\eta_1 \tag{9}$$

where $Q_{1w} = D_1 W\tilde{X} + D_2$ and $L_{1w} = \frac{1}{2n}(y + r)^T W\tilde{X}$. Here $W = \mathrm{diag}(w_1, \cdots, w_n)$ is the diagonal matrix whose element is the weight for each training datum. Let the solution of (9) be $\hat{\eta}_{1,SSVM}^w$.

The proposed algorithm can be summarized as

Step 1: Set the initial solution β_0^0, β^0 by the linear discriminant function.

Step 2: Solve the linear programming problem (7) or the linear system (9) with $w_i = 1$ until convergence.

Step 3: Set $w_i, i = 1, \cdots, n$. Solve (7) or (9) until convergence.

3.2. Multiclass Weighted SVM

Consider a *K*-class problem with a training set $\{x_i, y_i; i = 1, \cdots, n\}$, where x_i is the input vector and $y_i \in \{1,2,\cdots,K\}$ represents its class label. The classifier needs a *K*-dimensional decision function with a vector function $f(x) = (f_1(x), \cdots f_K(x))$ with a sum-to-zero constant $\sum_{k=1}^{K} f_k(x) = 0$ for any input vector $x \in \mathbb{R}^d$, minimizing the objective function [Lee et al., 2004].

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i \ne k)(f_k(x_i) + 1)_+ + \sum_{k=1}^{K} p_\lambda(f_k)$$

where the function $I(\cdot)$ is the indicator function which becomes one if the condition in the parenthesis is true, and zero otherwise. We consider the linear classifier $f_k(x_i) = \beta_{0k} + \beta_k^T x_i$ and the SCAD penalty function $p_\lambda(\cdot)$. The above objective function becomes

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i \ne k)(\beta_{0k} + \beta_k^T x_i + 1)_+ + \sum_{j=1}^{d}\sum_{k=1}^{K} p_\lambda(|\beta_{jk}|) \tag{10}$$
$$\sum_{k=1}^{K}\beta_{0k} = 0, \sum_{k=1}^{K}\beta_{jk} = 0, \text{ for } j = 1, \cdots, d.$$

The classification rule for the multiclass SVM naturally becomes $argmax_j\, f_j(x)$.

Now we consider a robust version of (10) given by

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} w_i I(y_i \ne k)(\beta_{0k} + \beta_k^T x_i + 1)_+ + \sum_{j=1}^{d}\sum_{k=1}^{K} p_\lambda' \beta_{jk}^0 |\beta_{jk}|, \tag{11}$$
$$\sum_{k=1}^{K}\beta_{0k} = 0, \sum_{k=1}^{K}\beta_{jk} = 0, \quad \text{for } j = 1, \cdots, d,$$

where the weight w_i is defined by $min\{f_y(x) - f_j(x), j \neq y\}$ [Wu & Liu, 2013]. The weight does have less impact on the misclassified data points far from the decision boundaries which can be support vectors.

Let $\xi_{ik} = w_i I(y_i \neq k)(\beta_{0k} + \beta_k^T x_i + 1)_+$. The minimization of the objective function (11) can solved via linear programming by solving

$$\begin{aligned} min_{\beta_{jk}^+, \beta_{jk}^-, \beta_{0k}^+, \beta_{0k}^-, \xi_{ik}} \quad & \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} \xi_{ik} \\ & + \sum_{j=1}^{d} \sum_{k=1}^{K} p_\lambda' \quad \beta_{jk}^0 \quad (\beta_{jk}^+ + \beta_{jk}^- \end{aligned} \tag{12}$$

subject to for all $1 \leq i \leq n$ and $j = 0,1, \cdots, d$

$$y_i \left(\sum_{j=1}^{d} x_{ij} (\beta_{jk}^+ - \beta_{jk}^-) + \beta_{0k}^+ - \beta_{0k}^- \right) \geq 1 - \frac{\xi_{ik}}{w_i},$$

$$\sum_{k=1}^{K} (\beta_{jk}^+ - \beta_{jk}^-) = 0, \sum_{k=1}^{K} (\beta_{0k}^+ - \beta_{0k}^-) = 0,$$

$$\xi_{ik} \geq 0, \beta_{jk}^+ \geq 0, \beta_{jk}^- \geq 0, \beta_{0k}^+ \geq 0, \beta_{0k}^- \geq 0.$$

Similar to Section 3.1, we can use the LQA method in multiclass SVM. Let a_{ik} the (i,k) element of the $n \times K$ matrix having the value of $I(y_i \neq k)$. Define $\eta^T = (\eta_1^T, \cdots, \eta_{K-1}^T)$, where $\eta_k = (\beta_{0k}, \beta_{1k}, \cdots, \beta_{dk})^T$. In the un-weighted case of (11) the objective function can be written by

$$\begin{aligned} \frac{1}{2} \sum_{k=1}^{K-1} \sum_{l=1}^{K-1} & \eta_k^T \{(Q_{B_2} + Q_{C_2}) + I(k = l)(Q_{A_{2,k}} \\ & + Q_{C_{1,k}})\} \eta_l \\ + \sum_{k=1}^{K-1} & \eta_k^T (L_{A_{1,k}} + L_{A_{2,k}} + L_{B_1} + L_{B_2}) \end{aligned} \tag{13}$$

where

$$Q_{A_{2,k}} = \frac{1}{2n} \sum_{i=1}^{n} \frac{a_{ik} \tilde{x}_i \tilde{x}_i^T}{1 + \eta_k^{0T} \tilde{x}_i}, Q_{B_2} = \frac{1}{2n} \sum_{i=1}^{n} \frac{a_{iK} \tilde{x}_i \tilde{x}_i^T}{|1 - \sum_{k=1}^{K-1} \eta_k^{0T} \tilde{x}_i|},$$

$$Q_{C_2} = \text{diag}\left(0, \frac{p_\lambda' \left| \sum_{m=1}^{K-1} \beta_{1m}^0 \right|}{|\sum_{m=1}^{K-1} \beta_{1m}^0|}, \cdots, \frac{p_\lambda' \left| \sum_{m=1}^{K-1} \beta_{dm}^0 \right|}{|\sum_{m=1}^{K-1} \beta_{dm}^0|}\right),$$

$$Q_{C_{1,k}} = \text{diag}\left(0, \frac{p_\lambda'(|\beta_{1k}^0|}{\beta_{1k}^0}, \cdots, \frac{p_\lambda'(|\beta_{dk}^0|}{\beta_{dk}^0}\right),$$

$$L_{A_{2,k}} = \frac{1}{2n} \sum_{i=1}^{n} \frac{a_{ik} \tilde{x}_i}{1 + \eta_k^{0T} \tilde{x}_i}, L_{A_{1,k}} = \frac{1}{2n} \sum_{i=1}^{n} a_{ik} \tilde{x}_i,$$

$$L_{B_2} = -\frac{1}{2n} \sum_{i=1}^{n} \frac{a_{iK} \tilde{x}_i}{|1 - \sum_{k=1}^{K-1} \eta_k^{0T} \tilde{x}_i|}, L_{B_1} = -\frac{1}{2n} \sum_{i=1}^{n} a_{iK} \tilde{x}_i,$$

where η_k^0 denotes an initial value of η_k. Then (13) can be simplified by

$$\frac{1}{2} \eta^T Q \eta + \eta^T L \tag{14}$$

where 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 L is the vector of the 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 $(d+1)$.

Now we consider a weighted version of the objective function (11). The minimization problem of (11) can be written by

$$\frac{1}{2} \eta^T Q_w \eta + \eta^T L_w \tag{15}$$

where Q_w and L_w are defined similarly to Q and L in (14) except that the element a_{ik} should be replaced by $w_i a_{ik}$. In the un-weighted case, the objective function (15) reduces to (14).

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

Step 1: Set the initial solution β_0^0, β^0 by the linear discriminant function.

Step 2: Solve the linear programming problem (12) or with $w_i = 1$ until convergence.

Step 3: Set $w_i, i = 1, \cdots, n$. Solve the linear programming problem (12) until convergence.

IV. SIMULATION RESULTS

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

4.1. Simulation

We consider the sample sizes 100, 1000, 10000 of the training data, the tuning data and the test data, respectively. To select an appropriate tuning parameter λ we find the tuning parameter minimizing the misclassification rate for $log_2 \lambda = -10, -9, \cdots, 1, 2$. If the minimum values are tied, the maximum value of such tuning parameters would be selected for general learning capability. And we used the approximation objective function (9) and (15).

First we consider a multiclass example with $K = 3$, $d = 2$. The data x is generated from the bivariate normal distribution $N(\mu_k, I_2)$, where $\mu_1 = (\sqrt{3}, 1)^T$, $\mu_2 = (-\sqrt{3}, 1)^T$, $\mu_3 = (0, -2)^T$ and $\sigma^2 = 2$ [Jung, 2012]. We contaminated the data by flipping the response to one of other response value with a given probability per $c/2$.

After finishing the learning for the training data we compute the mean and standard deviation of the misclassification rate for the test data. Table1 summarizes the results for 100 replications. The number in the table is the mean of the misclassification rate and the number in parenthesis is the sample standard deviation. It shows that our proposed robust algorithm is the best among the four methods for the mean of misclassification rate. Especially in the point of the standard deviation the value of our proposed algorithm is the least value among the methods.

Table 1: Simulation Results for Three Classes

Contamination Rate	L_2 SVM	L_1 SVM	SCAD SVM	Robust 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)

4.2. Real Data

We choose a real data 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 as usual, the training set, the tuning set and the test set. The number of samples for each set is 115, respectively. For contaminating the data we conducted the flipping as in Section 4.1. The flipping rates are 0%, 3%, 6% and 10%. The simulation is conducted for 100 replications. Table 3 shows the performance for robustness on the contaminated data. The robustness of our proposed method is best among the methods in view of resistance to outliers.

Table 2: Average of Misclassification Rates for the Liver Data

Contamination Rate	L_2 SVM	L_1 SVM	SCAD SVM	Robust 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)

V. CONCLUSION

In this paper we proposed a robust algorithm for multiclass SVM with the SCAD penalty function. We used a weight function for robustness. We derived two approximation objective functions to treat the non-convex optimization problem. One is LLA and the other is LQA. Even though LLA is more efficient than LQA, however its implementation is not easy. The simulation shows the effectiveness of our proposed algorithm.

ACKNOWLEDGEMENTS

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 (NRF-2012R1A1A4A01004594).

REFERENCES

[1] V. Vapnik (1995), "The Nature of Statistical Learning Theory", *Springer*, New York, Pp. 131–170.

[2] R. Tibshirani (1996), "Regression Shrinkage and Selection via the LASSO", *Journal of the Royal Statistical Society, Series B*, Vol. 58, Pp. 267–288.

[3] P.S. Bradley & O.L. Mangasarian (1998), "Feature Selection via Concave Minimization and Support Vector Machines", *Proceedings of the Fifteenth International Conference on Machine Learning*, Editors: J. Shaclik, Pp. 82–90, *Morgan Kaufmann*, San Francisco.

[4] J. Fan & R. Li (2001), "Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties", *Journal of the American Statistical Association*, Vol. 96, Pp. 1348–1360.

[5] T. Hastie, R. Tibshirani & J. Friedman (2001), "The Elements of Statistical Learning: Data Mining, Inference, and Prediction", *Springer*, New York, Pp. 9–21.

[6] Y. Lee, Y. Lin & G. Wahba (2004), "Multicategory Support Vector Machines, Theory and Applications to the Classification of Microarray Data and Satellite Radiance Data", *Journal of American Statistical Association*, Vol. 99, Pp. 67–81.

[7] G. Fung & O.L. Mangasarian (2004), "A Feature Selection Newton Method for Support Vector Machine Classification", *Computational Optimization and Application*, Vol. 28, No. 2, Pp. 185–202

[8] Y. Liu & X. Shen (2006), "Multicategory ψ-Learning", *Journal of American Statistical Association*, Vol. 101, Pp. 500–509.

[9] H. H. Zhang, J. Ahn, X. Lin & C. Park (2006), "Gene Selection using Support Vector Machines with Non-Convex Penalty", *Bioinformatics*, Vol. 22, Pp. 88–95.

[10] L. Wang & X. Shen (2007), "On L_1-Norm Multiclass Support Vector Machines: Methodology and Theory", *Journal of American Statistical Association*, Vol. 102, Pp. 583–594.

[11] Y. Wu & Y. Liu (2007), "Robust Truncated-Hinge-Loss Support Vector Machines", *Journal of American Statistical Association*, Vol. 102, Pp. 974–983.

[12] H. Zou & R. Li (2008), "One-Step Sparse Estimates in Nonconcave Penalized Likelihood Models", *Annals of Statistics*, Vol. 36, Pp. 1509–1566.

[13] K.-M. Jung (2012), "Multiclass Support Vector Machines with SCAD", *Communications of the Korean Statistical Society*, Vol. 19, Pp. 655–662.

[14] K.-M. Jung (2013), "Weighted Support Vector Machines with the SCAD Penalty", *Communications for Statistical Applications and Methods*, Vol. 20, Pp.481–490.

[15] Y. Wu & Y. Liu (2013), "Adaptively Weighted Large Margin Classifiers", *Journal of Computational and Graphical Statistics*, Vol. 22, Pp. 416–432.

[16] J. Zhu, S. Rosset, T. Hastie & R. Tibshirani (2003), "1-Norm Support Vector Machines", *Advances in Neural Information Processing Systems*, Vol. 16, Editors: S. Thrun, L. Saul & B. Schölkopf, *MIT Press*, Cambridge, MA.

