Communications of the Korean Statistical Society Vol. 15, No. 3, 2008, pp. 441–450

Multiclass Classification via Least Squares Support Vector Machine Regression[†]

Jooyong Shim[1], Jongsig Bae[2], Changha Hwang[3]

Abstract

In this paper we propose a new method for solving multiclass problem with least squares support vector machine(LS-SVM) regression. This method implements one-against-all scheme which is as accurate as any other approach. We also propose cross validation(CV) method to select effectively the optimal values of hyper-parameters which affect the performance of the proposed multiclass method. Experimental results are then presented which indicate the performance of the proposed multiclass method.

Keywords: Classification; cross validation; least squares support vector machine; multiclass; one-against-all; support vector machine.

1. Introduction

Many real applications consist of multiclass classification problems. Support Vector Machine(SVM) was originally designed by Vapnik (1995) for binary classification. SVM is gaining popularity due to many attractive features and promising empirical performance. Extending it to multiclass problems is an ongoing research issue. There are commonly two types of multiclass extensions for SVM. One is the composition type methods built on a series of binary classification methods such as the one-against-one, one-against-all and error correcting output codes (Allwein *et al.*, 2000; Dietterich and Bakiri, 1995) and the other is the single machine type methods, which attempt to construct a multiclass classifier by solving a single optimization problem (Vapnik, 1998; Weston and Watkins, 1998; Lee *et al.*, 2001). There is no substantial agreement on which method is the best one for the multiclass problem (Rifkin and Klautau, 2004).

Despite of many successful application of SVM in classification and regression problem, training an SVM requires to solve a quadratic program(QP) problem. The QP is to optimize a quadratic function over a polyhedron, defined by linear equations and/or

[†] This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2006-311-C00246).

[1] Adjunct Professor, Department of Applied Statistics, Catholic University of Daegu, Kyungbuk 712-702, Korea.

[2] Professor, Department of Mathematics, Sungkyunkwan University, Suwon 440-746, Korea.

[3] Professor, Division of Information and Computer Science, Dankook University, Gyeonggido 448-160, Korea. Correspondence: chwang@dankook.ac.kr.

inequalities, which is time memory expensive. Suykens and Vandewalle (1999a) proposed the least squares SVM for binary classification, which is called LS-SVM. Its solution is given by a linear equation system instead of a QP problem. LS-SVM keeps explicit primal-dual formulations which has lots of advantages. Suykens and Vandewalle (1999b) proposed an extension of LS-SVM to the multiclass case.

In this paper we propose a multiclass method using LS-SVM regression approach and compare it with very popular multinomial logistic regression. This method implements one-against-all scheme which is as accurate as any other approach. We also derive the cross validation(CV) technique to select the hyper-parameters which affects the performance of the proposed multiclass method. The rest of paper is organized as follows. In Section 2 we present an overview of LS-SVM classification and regression and describe their relationship. In Section 3 we propose a multiclass method using LS-SVM regression with a brief review on LS-SVM regression for binary classification. In Section 4 we illustrate the generalized cross validation(GCV) function for selecting hyper-parameters. In Section 5 we perform the numerical studies with real data sets. In Section 6 we give the conclusions.

2. LS-SVM Classification and Regression

In this section we review some basic idea of LS-SVM classification and regression. See for further details Suykens and Vandewalle (1999a, 1999b) and Suykens (2001). We also show LS-SVM classification is actually equivalent to LS-SVM regression in binary classification case.

2.1. LS-SVM classification

We first review some basic idea of LS-SVM classification. Suykens and Vandewalle (1999a) modified Vapnik (1995)'s SVM classification formulation.

Given a training data set $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ with each input $\boldsymbol{x}_i \in R^d$ and corresponding binary class labels $y_i \in \{-1, +1\}$, we consider the following optimization problem in primal weight space:

$$L(\boldsymbol{w}_c, b_c, \boldsymbol{e}) = \frac{1}{2}\boldsymbol{w}_c^t \boldsymbol{w}_c + \frac{\gamma}{2}\sum_{i=1}^n e_i^2, \tag{2.1}$$

subject to equality constraints

$$y_i \left[\boldsymbol{w}_c^t \boldsymbol{\Phi}(\boldsymbol{x}_i) + b_c\right] = 1 - e_i, \quad i = 1, \ldots, n, \tag{2.2}$$

with $\boldsymbol{\Phi} : R^d \to R^{d_f}$ a function which maps the input space into a higher dimensional feature space of dimension d_f, weight vector $\boldsymbol{w}_c \in R^{d_f}$ in primal weight space, error variables $e_i \in R$ and bias term b_c. To find minimizer of the objective function, we can construct the Lagrangian function as follows,

$$L(\boldsymbol{w}_c, b_c, \boldsymbol{e}; \boldsymbol{\alpha}_c) = \frac{1}{2}\boldsymbol{w}_c^t\boldsymbol{w}_c + \frac{\gamma}{2}\sum_{i=1}^{n} e_i^2 - \sum_{i=1}^{n} \alpha_i(y_i\left[\boldsymbol{w}_c^t\boldsymbol{\Phi}(\boldsymbol{x}_i) + b_c\right] - 1 + e_i), \tag{2.3}$$

where α_i's are the Lagrange multipliers. Then, the conditions for optimality are given by

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{w}_c} &= \mathbf{0} \rightarrow \boldsymbol{w}_c = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{\Phi}(\boldsymbol{x}_i),\\
\frac{\partial L}{\partial b_c} &= 0 \rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0,\\
\frac{\partial L}{\partial e_i} &= 0 \rightarrow e_i = \frac{1}{\gamma}\alpha_i, \quad i = 1, \ldots, n,\\
\frac{\partial L}{\partial \alpha_i} &= 0 \rightarrow y_i\left[\boldsymbol{w}_c^t\boldsymbol{\Phi}(\boldsymbol{x}_i) + b_c\right] - 1 + e_i = 0, \quad i = 1, \ldots, n.
\end{aligned} \tag{2.4}$$

After eliminating e_i and $\boldsymbol{w}_c$, we could have the solution by the following linear equations

$$\begin{bmatrix} 0 & \boldsymbol{y}^t \\ \boldsymbol{y} & \boldsymbol{\Omega} + \frac{1}{\gamma}\boldsymbol{I} \end{bmatrix} \begin{bmatrix} b_c \\ \vec{\alpha}_c \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix}, \tag{2.5}$$

where $\boldsymbol{y} = (y_1, \ldots, y_n)^t$, $\mathbf{1} = (1, \ldots, 1)^t$, $\boldsymbol{\alpha}_c = (\alpha_1, \ldots, \alpha_n)^t$ and $\boldsymbol{\Omega} = \{\Omega_{kl}\}$ with $\Omega_{kl} = y_k y_l \boldsymbol{\Phi}(\boldsymbol{x}_k)^t\boldsymbol{\Phi}(\boldsymbol{x}_l)$, $k, l = 1, \ldots, n$. From application of the Mercer's condition (Mercer, 1909) we can choose a kernel $K(\cdot, \cdot)$ such that

$$K(\boldsymbol{x}_k, \boldsymbol{x}_l) = \boldsymbol{\Phi}(\boldsymbol{x}_k)^t\boldsymbol{\Phi}(\boldsymbol{x}_l), \quad k, l = 1, \ldots, n. \tag{2.6}$$

It is noted that $\boldsymbol{\Omega} = \boldsymbol{YKY}$ for $\boldsymbol{Y} = \text{diag}\{\boldsymbol{y}\}$ and $\boldsymbol{K} = \{K_{kl}\}$ with $K_{kl} = K(\boldsymbol{x}_k, \boldsymbol{x}_l)$.

By solving the linear equations (2.5), we obtain the solution

$$\boldsymbol{\alpha}_c = \left(\boldsymbol{YKY} + \frac{1}{\gamma}\boldsymbol{I}\right)^{-1}(\mathbf{1} - b_c\boldsymbol{y}) \quad \text{and} \quad b_c = \frac{\boldsymbol{y}^t\left(\boldsymbol{YKY} + \frac{1}{\gamma}\boldsymbol{I}\right)^{-1}\mathbf{1}}{\boldsymbol{y}^t\left(\boldsymbol{YKY} + \frac{1}{\gamma}\boldsymbol{I}\right)^{-1}\boldsymbol{y}}. \tag{2.7}$$

Finally, for a given $\boldsymbol{x}$ in dual space the nonlinear LS-SVM classifier becomes

$$\hat{y}_c(\boldsymbol{x}) = \text{sign}\left[\sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b_c\right]. \tag{2.8}$$

In particular, for the given training data set, we obtain

$$\hat{\boldsymbol{y}}_c = \text{sign}[\boldsymbol{KY}\boldsymbol{\alpha}_c + b_c\mathbf{1}]. \tag{2.9}$$

We focus on the choice of an Gaussian kernel $K(\boldsymbol{x}_k, \boldsymbol{x}_l) = \exp(-\|\boldsymbol{x}_k - \boldsymbol{x}_l\|^2/\sigma^2)$ for the sequel. Here the linear classifier can be regarded as the special case of the nonlinear classifier by using identity feature mapping function, that is, $\boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{x}$ which implies the linear kernel such that $K(\boldsymbol{x}_k, \boldsymbol{x}_l) = \boldsymbol{x}_k^t\boldsymbol{x}_l$.

2.2. LS-SVM regression

The LS-SVM model for regression estimation has the following representation in feature space

$$y(\boldsymbol{x}) = \boldsymbol{w}_r^t \boldsymbol{\Phi}(\boldsymbol{x}) + b_r, \tag{2.10}$$

where $\boldsymbol{x} \in R^d$, $y \in R$. The use of the nonlinear mapping $\boldsymbol{\Phi}(\cdot)$ is similar to the classifier case.

Given a training data set $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ with each input $\boldsymbol{x}_i \in R^d$ and corresponding output $y_i \in R$, we consider the following optimization problem in primal weight space:

$$L(\boldsymbol{w}_r, b_r, \boldsymbol{e}) = \frac{1}{2}\boldsymbol{w}_r^t \boldsymbol{w}_r + \frac{\gamma}{2}\sum_{i=1}^n e_i^2, \tag{2.11}$$

subject to equality constraints

$$y_i = \boldsymbol{w}_r^t \boldsymbol{\Phi}(\boldsymbol{x}_i) + b_r + e_i, \quad i = 1, \ldots, n. \tag{2.12}$$

The cost function with squared error and regularization corresponds to a form of ridge regression. To find minimizers of the objective function, we can construct the Lagrangian function as follows:

$$L(\boldsymbol{w}_r, b_r, \boldsymbol{e}; \boldsymbol{\alpha}_r) = \frac{1}{2}\boldsymbol{w}_r^t \boldsymbol{w}_r + \frac{\gamma}{2}\sum_{i=1}^n e_i^2 - \sum_{i=1}^n \alpha_i(\boldsymbol{w}_r^t \boldsymbol{\Phi}(\boldsymbol{x}_i) + b_r + e_i - y_i), \tag{2.13}$$

where α_i's are the Lagrange multipliers. Then, the conditions for optimality are given by

$$\begin{aligned}
&\frac{\partial L}{\partial \boldsymbol{w}_r} = \mathbf{0} \rightarrow \boldsymbol{w}_r = \sum_{i=1}^n \alpha_i \boldsymbol{\Phi}(\boldsymbol{x}_i),\\
&\frac{\partial L}{\partial b_r} = 0 \rightarrow \sum_{i=1}^n \alpha_i = 0,\\
&\frac{\partial L}{\partial e_i} = 0 \rightarrow e_i = \frac{1}{\gamma}\alpha_i, \quad i = 1, \ldots, n,\\
&\frac{\partial L}{\partial \alpha_i} = 0 \rightarrow y_i - b_r - \boldsymbol{w}^t \boldsymbol{\Phi}(\boldsymbol{x}_i) - e_i, \quad i = 1, \ldots, n.
\end{aligned} \tag{2.14}$$

After eliminating e_i and $\boldsymbol{w}_r$, we could have the solution by the following linear equations

$$\begin{bmatrix} 0 & \mathbf{1}^t \\ \mathbf{1} & \boldsymbol{K} + \frac{1}{\gamma}\boldsymbol{I} \end{bmatrix} \begin{bmatrix} b_r \\ \boldsymbol{\alpha}_r \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{y} \end{bmatrix}. \tag{2.15}$$

By solving the linear equations (2.15), we obtain the solution

$$\boldsymbol{\alpha}_r = \left(\boldsymbol{K} + \frac{1}{\gamma}\boldsymbol{I}\right)^{-1}(\boldsymbol{y} - b_r\mathbf{1}) \quad \text{and} \quad b_r = \frac{\mathbf{1}^t\left(\boldsymbol{K} + \frac{1}{\gamma}\boldsymbol{I}\right)^{-1}\boldsymbol{y}}{\mathbf{1}^t\left(\boldsymbol{K} + \frac{1}{\gamma}\boldsymbol{I}\right)^{-1}\mathbf{1}}. \tag{2.16}$$

Finally, for a given $\boldsymbol{x}$ in dual space the nonlinear LS-SVM regression becomes

$$\hat{y}_r(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b_r. \tag{2.17}$$

In particular, for the given training data set, we obtain

$$\hat{\boldsymbol{y}}_r = \boldsymbol{K}\boldsymbol{\alpha}_r + b_r\mathbf{1}. \tag{2.18}$$

2.3. Equivalence of LS-SVM classification and regression

We now show LS-SVM classification is actually equivalent to LS-SVM regression for binary classification case. To this end, we need to show $b_c = b_r$, $\boldsymbol{\alpha}_c = \boldsymbol{Y}\boldsymbol{\alpha}_r$.

First, we are going to show $b_c = b_r$. The numerator of b_c can be reexpressed as follows:

$$\begin{aligned}\boldsymbol{y}^t\left(\boldsymbol{YKY} + \frac{1}{\gamma}\boldsymbol{I}\right)^{-1}\mathbf{1} &= \boldsymbol{y}^t\boldsymbol{Y}^{-1}\left(\boldsymbol{K} + \frac{1}{\gamma}\boldsymbol{Y}^{-1}\boldsymbol{Y}^{-1}\right)^{-1}\boldsymbol{Y}^{-1}\mathbf{1}\\ &= \boldsymbol{y}^t\boldsymbol{Y}\left(\boldsymbol{K} + \frac{1}{\gamma}\boldsymbol{I}\right)^{-1}\boldsymbol{Y}\mathbf{1}\\ &= \mathbf{1}^t\left(\boldsymbol{K} + \frac{1}{\gamma}\boldsymbol{I}\right)^{-1}\boldsymbol{y},\end{aligned}$$

since $\boldsymbol{Y} = \boldsymbol{Y}^{-1}$, $\boldsymbol{Y}^{-1}\boldsymbol{Y}^{-1} = \boldsymbol{I}$ and $\boldsymbol{Y}\mathbf{1} = \boldsymbol{y}$ for binary classification case. Similarly, we can reexpress the denominator of b_c as $\mathbf{1}^t(\boldsymbol{K} + 1/\gamma\boldsymbol{I})^{-1}\mathbf{1}$. Thus, we finally show $b_c = b_r$.

We now show $\boldsymbol{\alpha}_c = \boldsymbol{Y}\boldsymbol{\alpha}_r$ as follows:

$$\begin{aligned}\boldsymbol{\alpha}_c &= \left(\boldsymbol{YKY} + \frac{1}{\gamma}\boldsymbol{I}\right)^{-1}(\mathbf{1} - b_c\boldsymbol{y})\\ &= \boldsymbol{Y}^{-1}\left(\boldsymbol{K} + \frac{1}{\gamma}\boldsymbol{Y}^{-1}\boldsymbol{Y}^{-1}\right)^{-1}\boldsymbol{Y}^{-1}(\mathbf{1} - b_r\boldsymbol{y})\\ &= \boldsymbol{Y}\left(\boldsymbol{K} + \frac{1}{\gamma}\boldsymbol{I}\right)^{-1}\boldsymbol{Y}(\mathbf{1} - b_r\boldsymbol{y})\\ &= \boldsymbol{Y}\left(\boldsymbol{K} + \frac{1}{\gamma}\boldsymbol{I}\right)^{-1}(\boldsymbol{y} - b_r\mathbf{1})\\ &= \boldsymbol{Y}\boldsymbol{\alpha}_r.\end{aligned}$$

Therefore, we obtain $\boldsymbol{KY}\boldsymbol{\alpha}_c + b_c\mathbf{1} = \boldsymbol{K}\boldsymbol{\alpha}_r + b_r\mathbf{1}$. That is, LS-SVM classification is equivalent to LS-SVM regression for binary classification case.

3. One-Against-All Multiclass LS-SVM Regression

LS-SVM is originally designed for binary classification and the extension of LS-SVM to the multiclass scenario is an ongoing research topic. The conventional way is to decompose the m-class problem into a series of two-class problems and construct several binary classifiers. The most widely used implementation is the one-against-all scheme, which constructs m LS-SVM classifiers with the j^{th} one separating class j from all the remaining classes. In this section we propose one-against-all multiclass LS-SVM regression by using the fact that LS-SVM classification is equivalent to LS-SVM regression for binary classification case.

Let the training data set be denoted by $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ with each input vector $\boldsymbol{x}_i \in R^d$ and the class label $y_i \in \{1, 2, \ldots, m\}$, where m is the number of classes. One-against-all multiclass LS-SVM regression constructs m binary LS-SVM regressor, each of which separates one class from all the rest. The j^{th} LS-SVM regressor is trained with all the training examples of the j^{th} class with positive labels and all the others with negative labels. Thus, for one-against-all multiclass LS-SVM regression, we transform $\boldsymbol{y}$ into $n \times m$ matrix $\boldsymbol{Y}$ which consists of -1 and 1 such that $Y_{ij} = 1$ and $Y_{ik} = -1$ for $j \neq k$ implies i^{th} example belongs to the j^{th} class. We have m LS-SVM regressors for binary classification with $\{\boldsymbol{x}_i, Y_{ij}\}_{i=1}^n$ for $j = 1, \ldots, m$. Mathematically the j^{th} LS-SVM regressor

$$\hat{y}_j(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i^j K(\boldsymbol{x}_i, \boldsymbol{x}) + b^j, \tag{3.1}$$

can be solved from the following linear equation system

$$\begin{bmatrix} 0 & \boldsymbol{1}^t \\ \boldsymbol{1} & \boldsymbol{K} + \frac{1}{\gamma}\boldsymbol{I} \end{bmatrix} \begin{bmatrix} b^j \\ \boldsymbol{\alpha}^j \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{Y}_{\cdot j} \end{bmatrix}, \tag{3.2}$$

where $\boldsymbol{Y}_{\cdot j}$ is the j^{th} column of $\boldsymbol{Y}$, b^j is a bias and $\boldsymbol{\alpha}^j$ is a vector of Lagrange multipliers.

For $\boldsymbol{K}_0 = [K(\boldsymbol{x}_1, \boldsymbol{x}), \ldots, K(\boldsymbol{x}_n, \boldsymbol{x})]$, we can rewrite LS-SVM regressor (3.1) as

$$\begin{aligned} \hat{y}_j(\boldsymbol{x}) &= b^j + \boldsymbol{K}_0\boldsymbol{\alpha}^j = [1\ \boldsymbol{K}_0] \begin{bmatrix} b^j \\ \boldsymbol{\alpha}^j \end{bmatrix} = [1\ \boldsymbol{K}_0] \begin{bmatrix} 0 & \boldsymbol{1}^t \\ \boldsymbol{1} & \boldsymbol{K} + \frac{1}{\gamma}\boldsymbol{I} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \boldsymbol{Y}_{\cdot j} \end{bmatrix} \\ &= [1\ \boldsymbol{K}_0] \begin{bmatrix} a_{11} & \boldsymbol{a}_{12} \\ \boldsymbol{a}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \begin{bmatrix} 0 \\ \boldsymbol{Y}_{\cdot j} \end{bmatrix} = [\boldsymbol{a}_{12} + \boldsymbol{K}_0\boldsymbol{A}_{22}]\boldsymbol{Y}_{\cdot j} = \boldsymbol{h}_0\boldsymbol{Y}_{\cdot j}, \end{aligned} \tag{3.3}$$

where $a_{11}, \boldsymbol{a}_{12}, \boldsymbol{a}_{21}$ and $\boldsymbol{A}_{22}$ are the corresponding components of the inverse matrix of the leftmost partitioned matrix in (3.2). Since $\boldsymbol{h}_0$ does not depend on $\boldsymbol{Y}_{\cdot j}$, we can write for the given example $\boldsymbol{x}$,

$$(\hat{y}_1(\boldsymbol{x}), \hat{y}_2(\boldsymbol{x}), \ldots, \hat{y}_m(\boldsymbol{x})) = \boldsymbol{h}_0\boldsymbol{Y}. \tag{3.4}$$

Therefore, once we have $\boldsymbol{a}_{12}$ and $\boldsymbol{A}_{22}$, we do not need to solve m linear equations (3.2). That is, we can obtain multiclass LS-SVM regression in one step.

At the classification phase, an example $\boldsymbol{x}$ is classified as in class j^* whose $\hat{y}_{j^*}$ produces the largest value

$$j^* = \underset{j=1,2,\ldots,m}{\text{argmax}}\ \hat{y}_j(\boldsymbol{x}). \tag{3.5}$$

4. GCV for the Proposed Multiclass LS-SVM Regression

Although each LS-SVM is tuned very well for the binary problem, there is no guarantee that they work well together for the entire multiclass problem. Thus, we propose a CV technique to effectively select hyper-parameters γ, σ^2 in one step for multiclass LS-SVM regression. The CV function can be defined as follows:

$$\mathrm{CV}(\boldsymbol{\lambda}) = \frac{1}{n}\sum_{i=1}^{n}\left\{Y_{im_i} - \hat{Y}_{im_i}^{(-i)}(\boldsymbol{\lambda})\right\}^2, \tag{4.1}$$

where $\boldsymbol{\lambda}$ is the set of hyper-parameters and $\hat{Y}_{im_i}^{(-i)}(\boldsymbol{\lambda})$ is the predicted value of Y_{im_i} obtained from the data without i^{th} observation. Here m_i is the column number of the i^{th} row of $\boldsymbol{Y}$ such that $Y_{im_i} = 1$, which implies that the i^{th} observation belongs to the m_i^{th} class. The CV can be rewritten as

$$\mathrm{CV}(\boldsymbol{\lambda}) = \frac{1}{n}\sum_{i=1}^{n}\left\{1 - \hat{Y}_{im_i}^{(-i)}(\boldsymbol{\lambda})\right\}^2. \tag{4.2}$$

Since for each candidate of hyper-parameters, $\hat{Y}_{im_i}^{(-i)}(\boldsymbol{\lambda})$ for $i = 1, \ldots, n$ should be evaluated, selecting parameters using CV function is computationally formidable.

By leaving-out-one lemma (Kimeldorf and Wahba, 1971),

$$\begin{aligned}\left\{Y_{im_i} - \hat{Y}_{im_i}^{(-i)}(\boldsymbol{\lambda})\right\} - (Y_{im_i} - \hat{Y}_{im_i}) &= \hat{Y}_{im_i} - \hat{Y}_{im_i}^{(-i)}(\boldsymbol{\lambda}) \\ &\approx \frac{\partial \hat{Y}_{im_i}}{\partial Y_{im_i}}\left\{Y_{im_i} - \hat{Y}_{im_i}^{(-i)}(\boldsymbol{\lambda})\right\},\end{aligned}$$

we have

$$Y_{im_i} - \hat{Y}_{im_i}^{(-i)}(\boldsymbol{\lambda}) \approx \frac{Y_{im_i} - \hat{Y}_{im_i}}{1 - \dfrac{\partial \hat{Y}_{im_i}}{\partial Y_{im_i}}} \quad \text{and} \quad \hat{Y}_{im_i} = \boldsymbol{h}_i \boldsymbol{Y}_{\cdot m_i}, \tag{4.3}$$

where $\boldsymbol{h}_i = [\boldsymbol{a}_{12} + \boldsymbol{K}_i \boldsymbol{A}_{22}]$ and $\boldsymbol{K}_i = [K(\boldsymbol{x}_1, \boldsymbol{x}_i), \ldots, K(\boldsymbol{x}_n, \boldsymbol{x}_i)]$ for $i = 1, \ldots, n$. Then the CV function can be obtained as

$$\mathrm{CV}(\boldsymbol{\lambda}) \approx \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1 - \hat{Y}_{im_i}(\boldsymbol{\lambda})}{1 - \dfrac{\partial \hat{Y}_{im_i}}{\partial Y_{im_i}}}\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1 - \hat{Y}_{im_i}(\boldsymbol{\lambda})}{1 - h_{ii}(\boldsymbol{\lambda})}\right)^2, \tag{4.4}$$

where h_{ii} for $i = 1, \ldots, n$, is the i^{th} diagonal element of the hat matrix $\boldsymbol{H} = [\boldsymbol{h}_1^t, \ldots, \boldsymbol{h}_n^t]^t$.

By replacing h_{ii}'s in (4.4) with their average $\mathrm{tr}(\boldsymbol{H})/n$, the generalized cross validation(GCV) function can be then obtained as follows:

$$\mathrm{GCV}(\boldsymbol{\lambda}) = \frac{n\sum_{i=1}^{n}\left\{1 - \hat{Y}_{im_i}(\boldsymbol{\lambda})\right\}^2}{\{n - \mathrm{tr}(\boldsymbol{H})\}^2}. \tag{4.5}$$

Table 5.1: The misclassification rates for multiclass LS-SVM regression and multinomial kernel logistic regression (standard deviation in parenthesis)

	Iris	Wine	Glass
LSSVM	0.0154(0.0142)	0.0309(0.0193)	0.2255(0.0389)
KLogistic	0.0226(0.0155)	0.0388(0.0201)	0.1984(0.0301)

5. Numerical Studies

We illustrate the performance of the proposed procedure through three real data sets available from UCI Machine Learning Depository (`http://www.ics.uci.edu/mlearn/MLRepository.html`), which are iris data set, wine data set and glass data set. The Gaussian kernel is used for multiclass classifications of given data sets.

From each data set, we randomly chose one training data set and 100 test data sets. We found that the regularization parameter does not affect much on the performance of multiclass LS-SVM regression, we fix $\gamma = 10$ and obtain CV functions and GCV functions, respectively, corresponding to the various values of σ^2. To illustrate the classification performance of multiclass LS-SVM regression, we run the multinomial kernel logistic regression (Shim *et al.*, 2007) and compare misclassification rates each other. The averages of 100 misclassification rates from multiclass LS-SVM regression and the multinomial kernel logistic regression are obtained from each test data set.

Iris data set of 3 classes has 4 variables and 150 observations. The training data consist of 100 observations and the test data consist of 50 observations. The kernel parameter σ^2 is obtained from training data as 2.6 by CV function and 2 by GCV function.

Wine data set of 3 classes, which is from results of wines grown in the same region in Italy but derived from three different cultivars, has 12 variables and 178 observations. The training data consist of 120 observations and the test data consist of 58 observations. The kernel parameter σ^2 is obtained from training data as 1.1 by CV function and 1.4 by GCV function.

Glass data set of 6 classes, which are from the study of classification of types of glass motivated by criminological investigation, has 9 variables and 214 observations. The training data consist of 140 observations and the test data consist of 74 observations. The kernel parameter σ^2 is obtained from training data as 2 by CV function and GCV function.

We found the values of CV function and GCV function are very close with $\gamma = 10$ for each training data set. Figure 5.1 shows the values of CV function (solid curve) and GCV function (dashed curve) for three data sets. In Figure 5.1 we can see that CV function and GCV function show similar behaviors. The averages and their standard deviations of 100 misclassification rates by multiclass LS-SVM regression with kernel parameters from GCV function and those of the multinomial kernel logistic regression are shown in Table 5.1. From the table we can see that although it is not proper comparison, multiclass LS-SVM regression is as accurate as the other classification method in these examples.

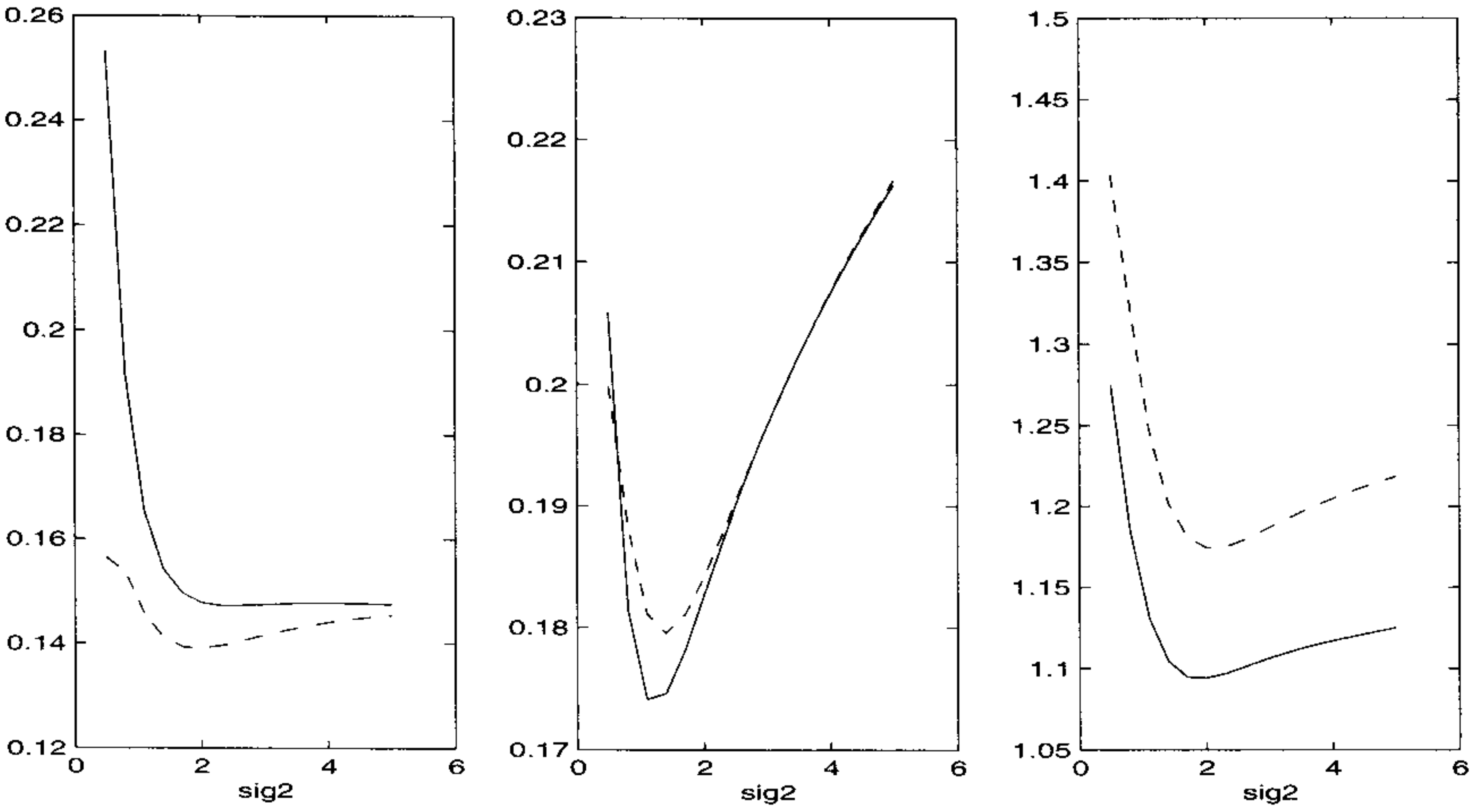

Figure 5.1: The values of CV function (solid curve) and GCV function (dashed curve) on various values of bandwidth parameter with C=10.

6. Conclusions

In this paper we proposed LS-SVM regression for the multiclass classification problem and obtained GCV function for the proposed procedure. An advantage of the proposed multiclass and model selection scheme is that it may be easily applied to multiclass problem, and selects effectively hyper-parameters of model in one step using cross-validation technique. The proposed method gives results that are comparable with the ones obtained by multinomial kernel logistic regression. The model selection using GCV function becomes easier and faster.

References

Allwein, E. L., Schapire, R. E. and Singer, Y. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers, *Journal of Machine Learning Research*, **1**, 113–141.

Dietterich, T. G. and Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes, *Journal of Artificial Intelligence Research*, **2**, 263–286.

Kimeldorf, G. S. and Wahba, G. (1971). Some results on Tchebycheffian spline functions, *Journal of Mathematical Analysis and Applications*, **33**, 82–95.

Lee, Y., Lin, Y. and Wahba, G. (2001). *Multicategory support vector machines*, Technical Report 1043, In Proceeding of the 33^{rd} Symposium on the Interface.

Mercer, J. (1909). Functions of positive and negative type and their connection with the theory of integral equations, *Philosophical Transactions of the Royal Society of London, Series A*, **209,** 415–446.

Rifkin, R. and Klautau, A. (2004). In defense of one-*vs*-all classification, *The Journal of Machine Learning Research*, **5**, 101–141.

Shim, J., Hong, D. H., Kim, D. H. and Hwang, C. (2007). Multinomial kernel logistic regression via bound optimization approach, *The Korean Communications in Statistics*, **14**, 507–516.

Suykens, J. A. K. and Vandewalle, J. (1999a). Least square support vector machine classifiers, *Neural Processing Letters*, **9**, 293–300.

Suykens, J. A. K. and Vandewalle, J. (1999b). Multiclass least squares support vector machines, In *Proceeding of the International Joint Conference on Neural Networks*, 900–903.

Suykens, J. A. K. (2001). Nonlinear modelling and support vector machines, In *Proceeding of the IEEE Instrumentation and Measurement Technology Conference*, 287–294.

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

Vapnik, V. N. (1998). *Statistical Learning Theory*, John Wieley & Sons, New York.

Weston, J. and Watkins, C. (1998). *Multi-Class SVM*, Technical Report, 98–104, Royal Holloway University of London.

[Received March 2008, Accepted April 2008]