Journal of the Korean Data & Information Science Society
2014, **25**(2), 455–464

http://dx.doi.org/10.7465/jkdi.2014.25.2.455
한국데이터정보과학회지

A transductive least squares support vector machine with the difference convex algorithm[†]

Jooyong Shim[1] · Kyungha Seok[2]

[12]Department of Data Science, Inje University

Received 14 February 2014, revised 28 February 2014, accepted 7 March 2014

Abstract

Unlabeled examples are easier and less expensive to obtain than labeled examples. Semisupervised approaches are used to utilize such examples in an effort to boost the predictive performance. This paper proposes a novel semisupervised classification method named transductive least squares support vector machine (TLS-SVM), which is based on the least squares support vector machine. The proposed method utilizes the difference convex algorithm to derive nonconvex minimization solutions for the TLS-SVM. A generalized cross validation method is also developed to choose the hyperparameters that affect the performance of the TLS-SVM. The experimental results confirm the successful performance of the proposed TLS-SVM.

Keywords: Difference convex algorithm, generalized cross validation function, kernel trick, least squares support vector machine, semisupervised learning, transductive least squares support vector machine.

1. Introduction

Classifiers can be informative or discriminative. Classical linear discriminant analysis is the most popular informative method and logistic regression is a popular discriminative method. In general, logistic regression is more robust than linear discriminant analysis because it relies on fewer assumptions about the classes. An important advantage of logistic regression is that it outputs an estimate of the probability that an object belongs to each of the possible classes. In recent years, there have been many new and exciting developments in kernel-based learning, particularly with regard to discriminative classification. The new developments have been largely stimulated by the research of Vapnik (1995, 1998) on statistical learning theory and support vector machines (SVM). Of all the kernel machines, the least squares SVM (LS-SVM) is the most appealing and promising method (Suykens and Vanderwalle, 1999; Suykens, 2000). Its strong point is that it solves linear equations, enabling computations to be performed in a simple, time-saving manner. Suykens *et al.* (2002)

† 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 (2012000646) and (2011-0009705).

1 Adjunct professor, Department of Data Science, Institute of Statistical Information, Inje University, Kimhae 621-749, Korea.

2 Corresponding author: Professor, Department of Data Science, Institute of Statistical Information, Inje University, Kimhae 621-749, Korea. E-mail: statskh@inje.ac.kr

achieved a great performance with the LS-SVM on several benchmark data set problems. Their results highlight the potential of the LS-SVM as a promising research tool. Model selection, which is the process of determining the optimal hyperparameters such as the kernel and regularization parameters, is a central issue in the task of fitting kernel machines. The goal of model selection is to identify the model that will yield the best generalization performance. Another advantage of the LS-SVM over a standard SVM is that it provides an easy model selection method called the generalized cross validation (GCV) function.

In classification problems, the label indicates the category or subpopulation to which the corresponding data belong. Most classification methods rely on the availability of large labeled data because the predictive performance improves as the volume of training data increases. In practice, there are many reasons why it is not always possible to obtain an example that consists solely of labeled data. To overcome this problem, Blum and Mitchell (1988) proposed a so called co-training algorithm to combine labeled and unlabeled data. Since then, researchers have studied semisupervised learning principle such as margin based learning (Vapnik, 1998; Wang *et al.*, 2007) and graph based method (Blum and Chawla, 2001; Zhu *et al.*, 2003). For recent readings, we can refer Chapelle *et al.* (2008), Xu *et al.* (2011) and Zhu and Goldberg (2009). Semisupervised methods use large amounts of unlabeled data with small amounts of labeled data, and the empirical results confirm that unlabeled data can be used to significantly improve the predictive performance. Recently, Seok (2012, 2013) applied semisupervised method to regression problem.

Transductive SVM (TSVM) introduced by Vapnik (1998) was originally designed to assign the labels of unlabeled data. TSVM seeks the largest margin hyperplane using both labeled and unlabeled data and yields good performance (Joachims, 1999; Chen *et al.*, 2002). The cost function of TSVM is appropriate but the implementation of TSVM is inadequate (Chapelle and Zien, 2005; Astorino and Fuduli, 2007). To treat this issue, Wang *et al.* (2007) developed the nonconvex minimization routine of TSVM based on the difference convex (DC) programming (An and Tao, 1997). Their algorithm gives better performance.

In order to utilize the strong points of the LS-SVM in the fields on the semisupervised learning literature, Adankon *et al.* (2009) proposed a semisupervised LS-SVM and Zhang *et al.* (2009) proposed least square transduction SVM. The simulation results in several benchmarks of their proposed algorithm yield that the methods can exploit unlabeled data to give good performance. However though their objective functions that are nonconvex are appropriate to use both labeled and unlabeled data, the algorithms to solve the objective functions are inadequate to solve nonconvex problem. Furthermore, the hyperparameters including kernel and regularization parameters are chosen by heuristic method that is not objective. To overcome nonconvex problem we derive a novel algorithm for learning a transductive LS-SVM for semisupervised learning. We utilize the DC algorithm to solve a nonconvex minimization of a negative objective function of the TLS-SVM. A GCV method is also developed in this paper to choose the hyperparameters that affect the performance of the TLS-SVM. The experimental results with real and synthetic data confirm the successful performance of the proposed TLS-SVM.

The rest of this paper is organized as follows. Section 2 reviews the LS-SVM and the GCV function. Section 3 solves the TLS-SVM with the DC algorithm. Section 4 presents some numerical studies that indicate the performance of the TLS-SVM. Section 5 contains conclusion and discussion.

2. LS-SVM

Since LS-SVM classification is actually equivalent to LS-SVM regression for binary response {-1,1}, we review LS-SVM regression in this section. 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:

$$L(\boldsymbol{w}, b, \boldsymbol{e}) = \frac{1}{2}\boldsymbol{w}'\boldsymbol{w} + \frac{C}{2}\sum_{i=1}^{n} e_i^2$$

subject to equality constraints

$$y_i = \boldsymbol{w}'\phi(\boldsymbol{x}_i) + b + e_i,\ \ i = 1, \cdots, n,$$

where $C > 0$ is a regularization parameter and $\phi(\cdot)$ is a feature mapping function which maps the input space into a higher dimensional space. It is well known that $\phi(\boldsymbol{x}_i)'\phi(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, which are obtained from the applicaion of Mercer (1909). 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:

$$L(\boldsymbol{w}, b, \boldsymbol{e}; \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{w}'\boldsymbol{w} + \frac{C}{2}\sum_{i=1}^{n} e_i^2 - \sum_{i=1}^{n} \alpha_i(\boldsymbol{w}'\phi(\boldsymbol{x}_i) + b + e_i - y_i)$$

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

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{0} \quad &\rightarrow \quad \boldsymbol{w} = \sum_{i=1}^{n} \alpha_i \phi(\boldsymbol{x}_i),\\
\frac{\partial L}{\partial b} = 0 \quad &\rightarrow \quad \sum_{i=1}^{n} \alpha_i = 0,\\
\frac{\partial L}{\partial e_i} = 0 \quad &\rightarrow \quad e_i = \frac{1}{C}\alpha_i, \quad i = 1, \cdots, n,\\
\frac{\partial L}{\partial \alpha_i} = 0 \quad &\rightarrow \quad y_i - b - \boldsymbol{w}'\phi(\boldsymbol{x}_i) - e_i = 0, \quad i = 1, \cdots, n.
\end{aligned}$$

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

$$\begin{bmatrix} \boldsymbol{K} + \frac{1}{C}\boldsymbol{I} & \boldsymbol{1} \\ \boldsymbol{1}' & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} \\ 0 \end{bmatrix}, \tag{2.1}$$

where $\boldsymbol{1}$ is an $n \times 1$ vector of 1's and $\boldsymbol{I}$ is an $n \times n$ identity matrix.

By solving the linear equations (2.1), we obtain the solution

$$\boldsymbol{\alpha} = (\boldsymbol{K} + \frac{1}{C}\boldsymbol{I})^{-1}(\boldsymbol{y} - b\boldsymbol{1}) \ \text{ and } \ b = \frac{\boldsymbol{1}'(\boldsymbol{K} + \frac{1}{C}\boldsymbol{I})^{-1}\boldsymbol{y}}{\boldsymbol{1}'(\boldsymbol{K} + \frac{1}{C}\boldsymbol{I})^{-1}\boldsymbol{1}}.$$

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

$$sign(\boldsymbol{f}) = sign(\boldsymbol{K}\boldsymbol{\alpha} + b\boldsymbol{1}),$$

where $\boldsymbol{f} = (f(\boldsymbol{x}_1), \cdots, f(\boldsymbol{x}_n))'$ can be expressed as the linear combination of y_i's as follows:

$$\boldsymbol{f} = H\boldsymbol{y}.$$

Here $H = \begin{bmatrix} \boldsymbol{K} & \boldsymbol{1} \end{bmatrix} \begin{bmatrix} S_{11} \\ S_{21} \end{bmatrix}$ where $\begin{bmatrix} S_{11} \\ S_{21} \end{bmatrix}$ is the $(n+1) \times n$ submatrix of the inverse of the leftmost matrix S of (2.1) such that $S^{-1} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}$.

The functional structures of LS-SVM is characterized by the hyperparameters - regularization parameter and kernel parameter. To choose optimal values of hyper- parameters of the model we define a leave-one-out (LOO) cross validation (CV) function as follows:

$$CV(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_i^{(-i)})^2,$$

where θ is a set of hyperparameters, that is, kernel and regularization parameters, and $f_i^{(-i)}$ is the estimate of y_i without the i^{th} observation. Ordinary cross validation (OCV) function is obtained by applying the LOO lemma (Craven and Wahba, 1979) as follows:

$$OCV(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - f_i}{1 - h_{ii}}\right)^2,$$

where $h_{ii} = \partial f_i / \partial y_i$ is the i^{th} diagonal element of H. The GCV function is obtained as follows:

$$GCV(\theta) = \frac{n\sum_{i=1}^{n}(y_i - f_i)^2}{(n - trace(H))^2}.$$

Among the candidates of sets of hyperparameters, we choose the optimal values of hyperparameters which minimize the OCV function or the GCV function.

3. Transductive LS-SVM

In the semisupervised learning, a labeled data set $(\boldsymbol{X}_1, \boldsymbol{y}_1) = \{\boldsymbol{x}_i, y_i\}_{i=1}^{n_1}$ is observed together with an independent unlabeled data set $\boldsymbol{X}_2 = \{\boldsymbol{x}_j\}_{j=n_1+1}^{n_1+n_2}$. The TLS-SVM uses an idea of minimizing the objective function given the labeled data set and the unlabeled data set as follows:

$$l(\boldsymbol{f}) = \frac{1}{2}\|\boldsymbol{f}\|_H^2 + \frac{C_1}{2}\sum_{i=1}^{n_1}(y_{1i} - f_i)^2 + \frac{C_2}{2}\sum_{j=n_1+1}^{n_1+n_2}(sign(f_j) - f_j)^2,$$

which is equivalent to

$$l(\boldsymbol{f}) = \frac{1}{2}\|\boldsymbol{f}\|_H^2 + \frac{C_1}{2}\sum_{i=1}^{n_1}(y_{1i} - f_i)^2 + \frac{C_2}{2}\sum_{j=n_1+1}^{n_1+n_2}(1 - |f_j|)^2, \tag{3.1}$$

where $f_i = f(\boldsymbol{x}_i) = \boldsymbol{w}'\phi(\boldsymbol{x}_i) + b$ and H is a reproducing kernel Hilbert space of a kernel K. Here $l(\boldsymbol{f})$ is not a convex function of $\boldsymbol{f}$ since $u(f) = (1 - |f|)^2$ is not a convex function of f. We decompose $u(f)$ into $u(f) = u_1(f) - u_2(f)$ where $u_1(f) = f^2$ and $u_2(f) = -1 + 2|f|$ as shown in Figure 3.1, where $u_1(f)$ and $u_2(f)$ are convex functions of f.

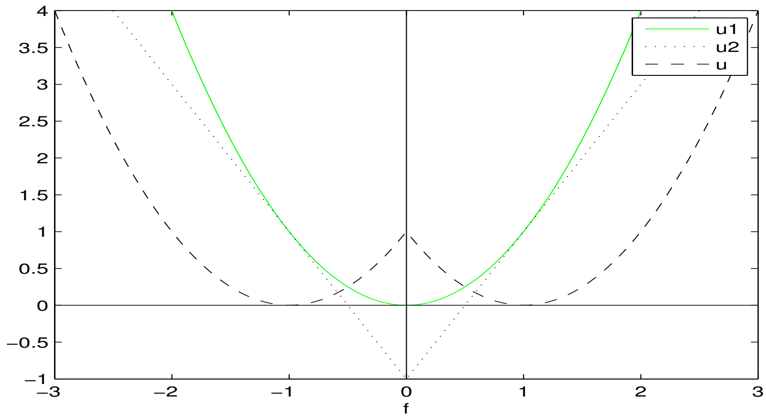

Figure 3.1 Plots of u_1, u_2 and $u = u_1 - u_2$ for the DC decomposition

Using this we construct a DC decomposition to utilize the DC algorithm as follows:

$$l(\boldsymbol{f}) = l_1(\boldsymbol{f}) - l_2(\boldsymbol{f}),$$

where $l_1(\boldsymbol{f}) = \frac{1}{2}\|\boldsymbol{f}\|_H^2 + \frac{C_1}{2}\sum_{i=1}^{n_1}(y_{1i} - f_i)^2 + \frac{C_2}{2}\sum_{j=n_1+1}^{n_1+n_2} f_j^2$ and $l_2(\boldsymbol{f}) = \frac{C_2}{2}\sum_{j=n_1+1}^{n_1+n_2}(1 - 2|f_j|)$. Given the decomposition, the DC algorithm solves a sequence of minimization problems,

$$l(\boldsymbol{f}) = l_1(\boldsymbol{f}) - \boldsymbol{f}'\triangledown l_2(\boldsymbol{f}^{(k)}),$$

where $\triangledown l_2(\boldsymbol{f}^{(k)})$ is a gradient vector of $l_2(\boldsymbol{f})$ at $\boldsymbol{f}^{(k)}$. $l(\boldsymbol{f})$ in (3.1) can be expressed in terms of $\boldsymbol{w}$, b and $\boldsymbol{e}$ as follows:

$$L(\boldsymbol{w}, b, \boldsymbol{e}) = \frac{1}{2}\boldsymbol{w}'\boldsymbol{w} + \frac{C_1}{2}\sum_{i=1}^{n_1} e_i^2 + \frac{C_2}{2}\sum_{j=n_1+1}^{n_1+n_2} e_j^2 - \boldsymbol{w}'\phi(\boldsymbol{X}_2)\triangledown_2^{(k-1)} - b\mathbf{1}_2'\phi(\boldsymbol{X}_2)\triangledown_2^{(k-1)} \tag{3.2}$$

subject to the equality constraints

$$y_i - \boldsymbol{w}'\phi(\boldsymbol{x}_i) - b = e_i,\ i = 1, \cdots, n_1 \quad \text{and} \quad \boldsymbol{w}'\phi(\boldsymbol{x}_j) + b = e_j,\ j = n_1, \cdots, n_1 + n_2,$$

where $\phi(\boldsymbol{X}_2) = (\phi(\boldsymbol{x}_{n_1+1}), \cdots, \phi(\boldsymbol{x}_{n_1+n_2}))$ and $\triangledown_2^{(k-1)} = sign(f_{n_1+1}^{(k-1)}, \cdots, f_{n_1+n_2}^{(k-1)})'$. To find minimizers of the objective function (3.2), we can construct the Lagrangian function as follows:

$$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{e}) &= \frac{1}{2}\boldsymbol{w}'\boldsymbol{w} + \frac{C_1}{2}\sum_{i=1}^{n_1} e_i^2 + \frac{C_2}{2}\sum_{j=n_1+1}^{n_1+n_2} e_j^2 - \boldsymbol{w}'\phi(\boldsymbol{X}_2)\triangledown_2^{(k-1)} - b\mathbf{1}_2'\phi(\boldsymbol{X}_2)\triangledown_2^{(k-1)} \\ &- \sum_{i=1}^{n_1}\alpha_i(e_i - y_i + \boldsymbol{w}'\phi(\boldsymbol{x}_i) + b) - \sum_{j=n_1+1}^{n_1+n_2}\alpha_j(e_i - \boldsymbol{w}'\phi(\boldsymbol{x}_j) - b), \end{aligned}$$

where $\mathbf{1}_k$ is an $n_k \times 1$ vector of 1's and α_i's are the Lagrange multipliers. Then, the conditions for optimality are given by

$$
\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{w}} = \mathbf{0} \quad &\rightarrow \quad \boldsymbol{w} = \sum_{i=1}^{n_1} \alpha_i \phi(\boldsymbol{x}_i) - \sum_{j=n_1+1}^{n_1+n_2} \alpha_j \phi(\boldsymbol{x}_j) + \phi(\boldsymbol{X}_2)\nabla_2^{(k-1)} \\
\frac{\partial L}{\partial b} = 0 \quad &\rightarrow \quad \sum_{i=1}^{n_1} \alpha_i - \sum_{j=n_1+1}^{n_1+n_2} \alpha_j + \mathbf{1}_2'\nabla_2^{(k-1)} = 0 \\
\frac{\partial L}{\partial e_i} = 0 \quad &\rightarrow \quad e_i = \frac{1}{C_1}\alpha_i, \quad i = 1, \cdots, n_1 \\
\frac{\partial L}{\partial e_j} = 0 \quad &\rightarrow \quad e_j = \frac{1}{C_2}\alpha_j, \quad j = n_1+1, \cdots, n_2 \\
\frac{\partial L}{\partial \alpha_i} = 0 \quad &\rightarrow \quad e_i - y_i + \boldsymbol{w}'\phi(\boldsymbol{x}_i) + b = 0, \quad i = 1, \cdots, n_1 \\
\frac{\partial L}{\partial \alpha_j} = 0 \quad &\rightarrow \quad e_j - \boldsymbol{w}'\phi(\boldsymbol{x}_i) - b = 0, \quad j = n_1+1, \cdots, n_1+n_2
\end{aligned}
$$

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

$$
\begin{bmatrix} K_{11} + \frac{1}{C_1}\boldsymbol{I}_1 & -K_{22} & \mathbf{1}_1 \\ -K_{21} & K_{22} + \frac{1}{C_2}\boldsymbol{I}_2 & -\mathbf{1}_2 \\ \mathbf{1}_1' & -\mathbf{1}_2' & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y}_1 - K_{12}\nabla_2^{(k-1)} \\ K_{22}\nabla_2^{(k-1)} \\ \mathbf{1}_2'\nabla_2^{(k-1)} \end{bmatrix}, \tag{3.3}
$$

where $K_{ij} = K(\boldsymbol{X}_i, \boldsymbol{X}_j)$ and $\boldsymbol{I}_k$ is an $n_k \times n_k$ identity matrix for $i, j, k = 1, 2$. Thus, estimation procedure of TLS-SVM with the DC algorithm is given as follows:

(i) Set initial value of $(f_{n_1+1}^{(0)}, \cdots, f_{n_1+n_2}^{(0)})'$ from pre-estimation with labeled data as the training data and unlabeled data as the test data.

(ii) At $(k+1)^{st}$ iteration, find $(\boldsymbol{\alpha}^{(k+1)}, b^{(k+1)})$ from (3.3) with $\nabla_2^{(k)} = C_2 sign(f_{n_1+1}^{(k)}, \cdots, f_{n_1+n_2}^{(k)})'$, where $(f_{n_1+1}^{(k)}, \cdots, f_{n_1+n_2}^{(k)})' = \begin{bmatrix} K_{21} \\ -K_{22} \end{bmatrix} \boldsymbol{\alpha}^{(k)} + K_{22}\nabla_2^{(k)} + b^{(k)}$.

(iii) Iterate (ii) until $|L(\boldsymbol{\alpha}^{(k+1)}, b^{(k+1)}) - L(\boldsymbol{\alpha}^{(k)}, b^{(k)})| < tol$.

Finally, for a given test data $\boldsymbol{x}_t$ the nonlinear TLS-SVM function becomes

$$
f(\boldsymbol{x}_t) = \sum_{i=1}^{n_1} K(\boldsymbol{x}_t, \boldsymbol{x}_i)\alpha_i - \sum_{j=n_1+1}^{n_1+n_2} K(\boldsymbol{x}_t, \boldsymbol{x}_j)\alpha_j + \sum_{j=n_1+1}^{n_1+n_2} K(\boldsymbol{x}_t, \boldsymbol{x}_j)\nabla_{2,j-n_1} + b. \tag{3.4}
$$

At k^{th} iteration, we define the LOO-CV function as follows:

$$
CV(\theta) = \frac{1}{n_1}\sum_{i=1}^{n_1}(y_{1i} - f_i^{(-i)})^2,
$$

where θ is a set of hyperparameters, that is, kernel and regularization parameters, and $f_i^{(-i)}$ is the estimate of y_{1i} obtained from (3.3) and (3.4) with $(\boldsymbol{X}_1, \boldsymbol{y}_1)$ and $\boldsymbol{X}_2$ except the

i^{th} observation for $i = 1, \cdots, n_1$. By applying the LOO lemma and the first order Taylor expansion, we have

$$y_{1i} - f_i^{(-i)} \approx \frac{y_{1i} - f_i}{1 - h_{1,ii}}, \ i = 1, \cdots, n_1,$$

where $h_{1,ii} = \partial f_i / \partial y_{1i}, \ i = 1, \cdots, n_1$, is the i^{th} diagonal element of H_1 such that

$$\begin{aligned} \boldsymbol{f}_1 &= [K_{11}, -K_{12}, \mathbf{1}_1] \begin{bmatrix} S_{11} \\ S_{21} \end{bmatrix} \boldsymbol{y}_1 + [K_{11}, -K_{12}, \mathbf{1}_1] S^{-1} \begin{bmatrix} -K_{12} \\ K_{22} \\ \mathbf{1}_2' \end{bmatrix} \triangledown_2^{(k)} \\ &= H_1 \boldsymbol{y}_1 + H_2 \triangledown_2^{(k)}, \end{aligned} \tag{3.5}$$

where $\begin{bmatrix} S_{11} \\ S_{21} \end{bmatrix}$ is the $(n_1 + n_2 + 1) \times n_1$ submatrix of the inverse of the leftmost matrix S of (3.3) such that $S^{-1} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}$. From (3.5) the OCV function for the TLS-SVM is obtained as follows:

$$OCV(\theta) = \frac{1}{n_1} \sum_{i=1}^{n_1} \left(\frac{y_i - f_i}{1 - h_{1,ii}} \right)^2.$$

By replacing $h_{1,ii}$ with $trace(H_1)/n_1$, the GCV function is obtained as follows:

$$GCV(\theta) = \frac{n_1 \sum_{i=1}^{n_1} (y_i - f_i)^2}{(n_1 - trace(H_1))^2}.$$

4. Numerical studies

The performance of the proposed TLS-SVM is illustrated through the use of simulated data sets and real data sets.

Example 4.1 The performance of the TLS-SVM is compared with the LS-SVM and the semisupervised LS-SVR (S2LS-SVR; Xu *et al.*, 2011). The S2LS-SVR is simple to implement since it solves quadratic problem like the LS-SVM. As usual, LS-SVM does not use the unlabeled data. To compare we generate 100 data sets, in which y_i is generated from a Bernoulli distribution with $p = 0.5$, x_{1i} and x_{2i} are independently generated from $N(y_i, 1), i = 1, \cdots, 400$. In each data set, 100 instances are randomly selected for training, and the remaining 300 instances are retained for testing. Among the training data, 80 unlabeled data are obtained by removing labels from a randomly chosen subset of the training data, whereas the remaining 20 training data are treated as labeled. We used the Gaussian kernel, $K(x_1, x_2) = exp(\frac{1}{s^2}||x_1 - x_2||^2)$ for the TLS-SVM, the LS-SVM and the S2LS-SVM. To obtain the optimal value of the hyperparameter, we used the OCV functions instead of the GCV functions due to the small size of the labeled data. Figure 4.1 shows values of the LOO-CV function, the OCV function and the GCV function for various values of s^2 in Gaussian kernel and $(C_1, C_2) = (100, 1)$. It shows that the OCV yields a more similar minimizer than that of the GCV. The average of 100 mean misclassification errors and their standard errors for the the TLS-SVM as (0.1428, 0.000684), (0.1738, 0.000743) for the LS-SVM and

(0.1736, 0.000729) for the S2LS-SVM. These results imply that the TLS-SVM has better capabilities than the LS-SVM or the S2LS-SVM and are shown in Figure 4.2. The better performance of the TLS-SVM is due to the fact that it acquires more information well from the unlabeled data.

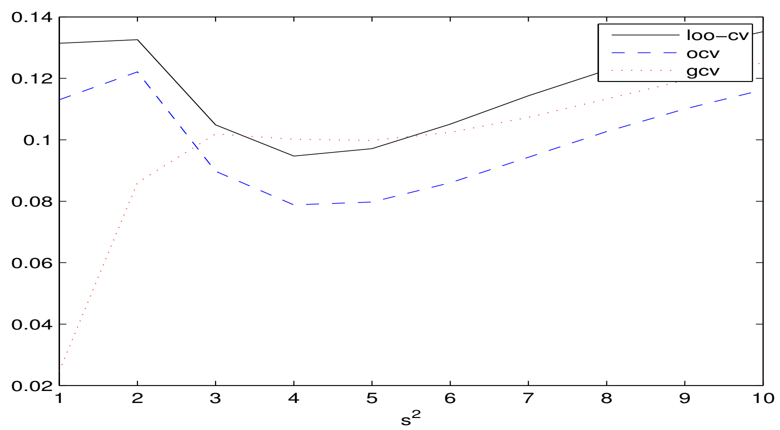

Figure 4.1 Values of LOO-CV function, OCV function and GCV function for various values of kernel parameters in Example 4.1

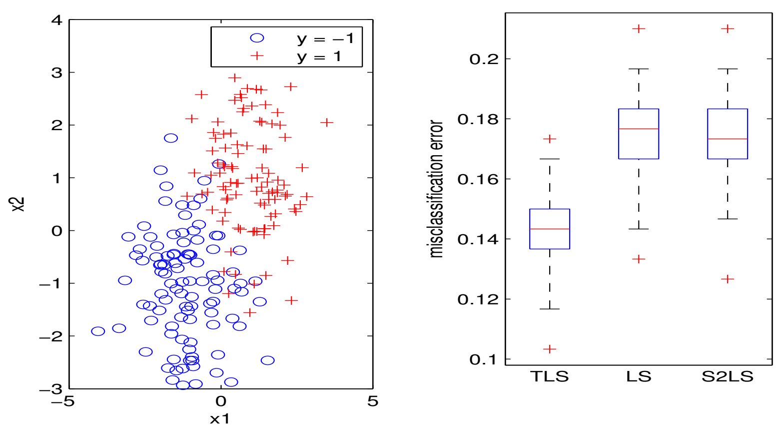

Figure 4.2 Scatter plots of a training data (left) and box plots of misclassification error rates of 100 test data sets in Example 4.1 (right)

Example 4.2 Five benchmark data sets - Wisconsin Breast Cancer, Ionosphere, Iris of Virginia and Setosa, Mushroom, Pima Indians Diabetes- are obtained from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/). Each data set is randomly divided into halves for training and testing, where 20% of training data are labeled and 80% are unlabeled.

We repeated 100 times to obtain the average of 100 misclassification errors of test data as well as the estimated standard error. We also obtain the average of 100 misclassification errors of test data of LS-SVM, where labeled data alone are used as the training data. The optimal values of (C_1, C_2) and Gaussian kernel parameter are obtained from OCV function. The results are summarized in Table 4.1, which indicate that TLS-SVM yields better performance than LS-SVM and S2LS-SVM in all data sets. Especially the error rate 0.0209 for Mushroom data is much less than 0.0903 and 0.0941 for LS-SVM's and S2LS-SVM's. The superiority of TLS-SVM may be due to the DC minimization strategy, where the DC property of the cost function has been effectively applied.

Table 4.1 Results of five data sets in Example 4.2 (standard error in parenthesis)

data	size	kernel	TLS-SVM	LS-SVM	S2LS-SVM
Brcancer	683	linear	0.0842 (0.0018)	0.0883 (0.0019)	0.0850 (0.0018)
Ionosphere	351	Gaussian	0.3351 (0.0039)	0.3389 (0.0028)	0.3489 (0.0021)
Iris	100	Gaussian	0.2246 (0.0149)	0.2795 (0.0137)	0.2905 (0.0136)
Mushroom	8124	Gaussian	0.0209 (0.0002)	0.0903 (0.0005)	0.0941 (0.0005)
Pima	768	linear	0.2857 (0.0012)	0.2903 (0.0013)	0.2860 (0.0013)

5. Conclusions

In this paper, we implemented a semisupervised classification method with the TLS-SVM, which is based on an LS-SVM. We utilized the difference convex algorithm to derive nonconvex minimization problem for the TLS-SVM. The proposed TLS-SVM is specially important in classification problems in which it is impossible to obtain fully labeled data. The experimental results show that the proposed TLS-SVM outperforms the LS-SVM and the S2LS-SVM. Thus, the feasibility of using the proposed TLS-SVM for semisupervised classification problems is confirmed. A multiclass TLS-SVM will be considered in future work.

References

Adankon, M., Cheriet, M. and Biem, A. (2009). Semisupervised least squares support vector machine. *IEEE Tranactions on Neural Networks*, **20**, 1858-1870.

An, L. and Tao, P. (1997). Solving a class of linearly constrained indefinite quadratic problems by D.C. algorithms. *Journal of Global Optimization*, **11**, 253–285.

Astotino, A. and Fuduli, A. (2005). Nonsmooth optimization technique for semisupervised classificatin. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**, 2135–2142.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *Proceedings of the 11th Annual Conference on Computational Learning Theory*, Madison, Wisconsin, 92-100.

Blum, A. and Chawla, S. (2001). Learning from labeled and unlabeled data using graph minicuts. In *Proceedings of ICML 2001*, 19-26.

Chapelle, O. and Zien, A. (2005). Semi-supervised classification by low density separarion. In *Proceedings of tht 10th International Workshop on Artificial Intelligence and Statistics*, 57-64.

Chapelle, O., Sindhwani, V. and Keerthi, S. (2008). Optimization techniques for semisupervised support vector machines. *Journal of Machine Learning Research*, **9**, 203–233.

Chen, Y., Wang, G. and Dong, S. (2002). Learning with progressive transductive support vector machine. *Proceedings of International Conference on Data Mining*, Maebashi City, Japan, 67-74.

Craven, P. and Wahba, G. (1979). Smothing noisy data with spline functions. *Numerical Mathematics*, **31**, 377-403.

Joachims, T. (1999). Transductive inference for text classification using support vecter machines. *Proceedings of International conference on Machine Learning*, Bled, Slovenia, 200-209.

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

Seok, K. (2012). Study on semi-supervised local constant regression estimation. *Journal of the Korean Data & Information Science Society*, **23**, 579-585.

Seok, K. (2013). A study on semi-supervised kernel ridge regression estimation. *Journal of the Korean Data & Information Science Society*, **24**, 341-353.

Suykens, J. A. K. (2000). Least squares support vector machine for classication and nonlinear modeling. *Neural Network World*, Special issue on PASE 2000, **10**, 29-48.

Suykens, J. A. K. and Vanderwalle, J. (1999). Least square support vector machine classifier, *Neural Processing Letters*, **9**, 293-300.

Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B. and Vanthienen, J. (2002). *Least squares support vector machines*, World Scientific, Singapore.

Vapnik, V. N. (1995). *The nature of statistical learning theory*, Springer, New York.

Vapnik, V. N. (1998). *Statistical learning theory*, John Wiley, New York.

Wang, J., Shen, X. and Pan, W. (2007). On transductive support vector machine. *Contemporary Mathematics*, **43**, 7-19.

Xu, S., An. X., Qiao, X., Zhu, L. and Li, L. (2011) Semisupervised least squares support vector regression machines. *Journal of Information & Computational Science*, **8**, 885-892.

Xu, Z., King, I. and Lyu, M. R. (2010). *More than semi-supervised learning*, LAP LAMBERT Academic Publishing, Germany.

Zhang, R., Wang, W., Ma, Y. and Men, C. (2009). Least square transduction support vector machine. *Neural Processing Letters*, **29**, 133-142.

Zhu, X. and Goldberg, A. (2009). *Introduction to semi-supervised learning*, Morgan & Claypool Publishers, CA.