International Journal of Database Theory and Application
Vol.8, No.5 (2015), pp.65-76
http://dx.doi.org/10.14257/ijdta.2015.8.5.06

Least Squares Twin Support Vector Machine for Multi-Class Classification

Sugen Chen* and Juan Xu

School of Mathematics & Computational Science, Anqing Normal University, Anqing Anhui, 246133, PR China
chensugen@126.com

Abstract

Twin support vector machine (TWSVM) was initially designed for binary classification. However, real-world problems often require the discrimination more than two categories. To tackle multi-class classification problem, in this paper, a multiple least squares twin support vector machine is proposed. Our Multi-LSTSVM solves K quadratic programming problems (QPPs) to obtain K hyperplanes, each problem is similar to binary LSTSVM. Comparison against the Multi-LSSVM, Multi-GEPSVM, Multi-TWSVM and our Multi-LSTSVM on both UCI datasets and ORL, YALE face datasets illustrate the effectiveness of the proposed method.

Keywords: *Pattern classification, Least squares support vector machine, Twin support vector machine, Multi-class classification*

1. Introduction

Support vector machine (SVM) was originally introduced by Vapnik and his co-workers in the early 1990s [1-2] for binary classification and regression. SVM has already been widely applied to a variety of real-world problems ranging from image classification [3], text categorization [4] and bioinformatics [5], *etc*. However, one of the main challenges for SVM is the high computational complexity of quadratic programming problem (QPP) [6]. This drawback restricts the application of SVM to large-scale problems. Recently, nonparallel support vector machines have attracted widely attentions, and many nonparallel hyperplane classifiers were proposed for binary classification. For example, in 2006, Mangasarian and Wild [7] proposed the first nonparallel hyperplane classifier termed as generalized eigenvalue proximal support vector machine (GEPSVM), which seeks two nonparallel hyperplanes such that each hyperplane is close to one of the two classes and is as far as possible from the other class. From another aspect, Jayadeva *et al*. [8] proposed a twin support vector machine (TWSVM) which also aims at seeking two nonparallel hyperplanes such that each hyperplane is close to one of the two classes and depart from the other simultaneously. The idea of solving two smaller-sized QPPs rather than a single larger-sized QPP in SVM makes the learning of TWSVM four times faster than SVM. From then on, some of extensions of TWSVM have been widely investigated [9-15], *e.g*. TBSVM [9], PTSVM [10], TPMSVM [11], Robust TWSVM [12], RPTSVM [13], NHSVM [14] and NPSVM [15]. To improve the learning speed of TWSVM, after combining the spirit of TWSVM [8] and LSSVM [17], least squares twin support vector machine (LSTSVM) [16] has been proposed as a way to replace the QPPs in TWSVM with a linear system by using a squared loss function instead of the hinge one. Inspired by LSTSVM, linear LSPTSVM [18] and nonlinear LSPTSVM [19] have been introduced as a least squares version of PTSVM [10].

SVM and TWSVM are originally designed for binary classification problems. However, multi-class classification problem is often encountered in practical scenarios. Therefore, how to effectively extend classical SVM and TWSVM to multi-class classification

ISSN: 2005-4270 IJDTA

problems are still ongoing research issues. In the SVM and TWSVM framework, the following models are widely investigated: One-Versus-Rest SVMs (OVR-SVMs) [20], One-Versus-One SVMs (OVO-SVMs) [21], Error-correcting-output code SVMs (ECOC SVMs) [22] and Directed acyclic graph SVMs (DAGSVMs) [23]. Based on "one-versus-one-versus-rest" strategy, another form of multi-class classification algorithms such as K-SVCR [24], Twin-KSVR [25] and LST-KSVC [26] were proposed. In addition, Suykens and Vandewalle [27] extended the LSSVM methodology to the multi-class case. However, the speed in learning a model and the method for dealing with the potential unbalance of samples in different classes are still two critical problems for multi-class classification problems in SVMs. Furthermore, LSTSVM overcomes the sample unbalance problem in two classes by choosing two different penalty variables for different classes and faster in learning a model by solving system of linear equations.

Based on the above analysis, in this paper, we aim to extend the LSTSVM to multi-class classification problem, named Multi-LSTSVM. Moreover, regularization terms are added to control the complexity for finding proper hyperplanes and to make sure each hyperplane is closer to its own class and is as far as possible from the other class. Experimental results obtained on both UCI datasets and ORL, YALE face datasets illustrate the superiority of our Multi-LSTSVM over LSSVM, GEPSVM and TWSVM, which will be referred to as Multi-LSSVM, Multi-GEPSVM and Multi-TWSVM in the following when dealing with multi-class classification, respectively.

The rest of this paper is organized as follows. In Section 2, background knowledge is introduced, such as LSSVM and LSTSVM. Section 3 presents the details of our linear Multi-LSTSVM and its nonlinear version. Experimental results on real-world datasets are described in Section 4, and Section 5 contains concluding remarks.

2. Background

In this Section, we give a brief outline of LSSVM [17] and LSTSVM. [16].

2.1. Least Squares Support Vector Machine (LSSVM)

Consider the binary classification problem with the in training set

$$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\} \in (R^n \times Y)^m \tag{1}$$

Where $x_i \in R^n, y_i \in Y = \{1,-1\}, i = 1,2,\cdots,m$. For the given training set (1), the primal problem of standard LSSVM to be solved is

$$\begin{aligned} &\min_{w,b,\xi_i} \frac{1}{2} \| w \|^2 + \frac{C}{2} \sum_{i=1}^{m} \xi_i^2 \\ &s.t. \quad y_i((w \cdot x_i) + b) = 1 - \xi_i, \ i = 1,2,\cdots,m \end{aligned} \tag{2}$$

Where $C \geq 0$ is the penalty factor, ξ_i are slack variables.

For this primal problem, LSSVM solves its Lagrangian dual problem

$$\begin{aligned} &\min_{\alpha} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j (K(x_i, x_j) + \frac{\delta_{ij}}{C}) - \sum_{i=1}^{m} \alpha_i \\ &s.t. \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \end{aligned} \tag{3}$$

Where $K(x, x')$ is the kernel function and

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{4}$$

The solution of the above problem is given by the following system of linear equations

$$\begin{bmatrix} 0 & Y^T \\ Y & \Omega + C^{-1} I \end{bmatrix} \cdot \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ e \end{bmatrix} \tag{5}$$

Where $Y = (y_1, y_2, \cdots, y_m)^T$, $\Omega = (\Omega_{ij})_{m \times m} = (y_i y_j K(x_i, x_j))_{m \times m}$, I is the identity matrix and $e = (1,1,\cdots,1)^T \in R^m$. Therefore the decision function is

$$f(x) = \text{sgn}(g(x)) = \text{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i K(x_i, x) + b \right) \tag{6}$$

2.2. Least Squares Twin Support Vector Machine (LSTSVM)

Consider the following binary classification problem. Suppose that all of the data points in positive class are organized as the matrix $A \in R^{m_1 \times n}$ and the data points in negative class are denoted by a matrix $B \in R^{m_2 \times n}$.

Different from LSSVM [17], linear LSTSVM [16] seeks a pair of nonparallel hyperplanes.

$$f_1(x) = w_1^T x + b_1 = 0 \quad \text{and} \quad f_2(x) = w_2^T x + b_2 = 0 \tag{7}$$

The primal problems of linear LSTSVM are

$$\begin{aligned} &\min_{w_1, b_1, \xi_2} \frac{1}{2} \| Aw_1 + e_1 b_1 \|_2^2 + \frac{c_1}{2} \xi_2^T \xi_2 \\ &s.t. \quad -(Bw_1 + e_2 b_1) + \xi_2 = e_2 \end{aligned} \tag{8}$$

and

$$\begin{aligned} &\min_{w_2, b_2, \xi_1} \frac{1}{2} \| Bw_2 + e_2 b_2 \|_2^2 + \frac{c_2}{2} \xi_1^T \xi_1 \\ &s.t. \quad (Aw_2 + e_1 b_2) + \xi_1 = e_1 \end{aligned} \tag{9}$$

Where $c_1, c_2 \geq 0$ are the penalty factors, ξ_1, ξ_2 are slack variables, and e_1, e_2 are vectors of ones.

On substituting the equality constraint into the objective function and we can obtain the unconstrained optimization problem. Then, we derive the linear equations

$$[A\ e_1]^T [A\ e_1] [w_1^T\ b_1]^T + c_1 [B\ e_2]^T [B\ e_2] [w_1^T\ b_1]^T + c_1 [B\ e_2]^T e_2 = 0 \tag{10}$$

$$[B\ e_2]^T [B\ e_2] [w_2^T\ b_2]^T + c_2 [A\ e_1]^T [A\ e_1] [w_2^T\ b_2]^T - c_2 [A\ e_1]^T e_1 = 0 \tag{11}$$

Let $E = [A\ e_1]$, $F = [B\ e_2]$, from (10) and (11), we get

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -\left(F^T F + \frac{1}{c_1} E^T E\right)^{-1} F^T e_2 \tag{12}$$

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = \left(E^T E + \frac{1}{c_2} F^T F\right)^{-1} E^T e_1 \tag{13}$$

Note that the solutions to the pair of QPPs (8) and (9) can be found directly by solving two systems of linear equations (12) and (13), more details can be seen in reference [16]. Once w_1, b_1 and w_2, b_2 are obtained from (12) and (13), the two nonparallel hyperplanes (7) are known. A new data point $x \in R^n$ is then assigned to the positive or negative class by

$$Class \quad k = \arg \min_{i=1,2} \{ | w_1^T x + b_1 |, | w_2^T x + b_2 | \} \tag{14}$$

Where $|\cdot|$ is the absolute value.

3. Multi-LSTSVM

In this Section, we present our multi-class classifier for a *K*-class classification problem. The proposed algorithm evaluates all training points into a "One-Versus-All" structure and it solves *K* small-sized QPPs simultaneously. For convenience, we denote the number of data points of the *k*-th class as m_k and define the following matrices: the patterns belonging to the *k*-th class are represented by the matrix $A_k \in R^{m_k \times n}, k = 1,2,\cdots,K$. Moreover, define the matrix

$$B_k = [A_1^T, \cdots, A_{k-1}^T, A_{k+1}^T, \cdots, A_K^T]^T \in R^{\tilde{m}_k \times n}, \tilde{m}_k = m - m_k \tag{15}$$

Where m is the total number of training samples and B_k is composed of the patterns belonging to all classes except the *k*-th class.

3.1. Linear Multi-LSTSVM

Our linear Multi-LSTSVM seeks *K* hyperplanes, one for each class, and assigned class label according to which hyperplane is nearest to. In our proposed Multi-LSTSVM, the regularization term is added to objective function similar to references [9, 13]. Then, we have the following QPP

$$\begin{aligned} &\min_{w_k, b_k, \xi_k} \frac{1}{2} \| A_k w_k + e_{k1} b_k \|_2^2 + \frac{\upsilon_k}{2} (\| w_k \|^2 + b_k^2) + \frac{c_k}{2} \xi_k^T \xi_k \\ &s.t. \quad -(B_k w_k + e_{k2} b_k) + \xi_k = e_{k2} \end{aligned} \tag{16}$$

Where $\upsilon_k, c_k, \xi_k, e_{ki}$ $(k = 1,2,\cdots,K, i = 1,2)$ are regularization parameters, penalty parameters, slack variables and the vectors of ones, respectively.

On substituting the equality constraint into the objective function, we get the unconstrained optimization problem as follows.

$$\min_{w_k, b_k} L = \frac{1}{2} \| A_k w_k + e_{k1} b_k \|^2 + \frac{\upsilon_k}{2} (\| w_k \|^2 + b_k^2) + \frac{c_k}{2} (e_{k2} + (B_k w_k + e_{k2} b_k))^2 \tag{17}$$

Differentiating L with respect to w_k, b_k yields the following KKT conditions,

$$\frac{\partial L}{\partial w_k} = A_k^T (A_k w_k + e_{k1} b_k) + \upsilon_k w_k + c_k B_k^T [e_{k2} + (B_k w_k + e_{k2} b_k)] = 0 \tag{18}$$

$$\frac{\partial L}{\partial b_k} = e_{k1}^T (A_k w_k + e_{k1} b_k) + \upsilon_k b_k + c_k e_{k2}^T [e_{k2} + (B_k w_k + e_{k2} b_k)] = 0 \tag{19}$$

After combining (18) and (19), we can achieve the following equation,

$$\begin{bmatrix} A_k^T \\ e_{k1}^T \end{bmatrix} \cdot [A_k \quad e_{k1}] \cdot \begin{bmatrix} w_k \\ b_k \end{bmatrix} + \upsilon_k \begin{bmatrix} w_k \\ b_k \end{bmatrix} + c_k \begin{bmatrix} B_k^T \\ e_{k2}^T \end{bmatrix} \cdot [B_k \quad e_{k2}] \cdot \begin{bmatrix} w_k \\ b_k \end{bmatrix} + c_k \begin{bmatrix} B_k^T \\ e_{k2}^T \end{bmatrix} e_{k2} = 0 \tag{20}$$

Define $G_k = [A_k \quad e_{k1}], H_k = [B_k \quad e_{k2}], u_k = [w_k^T \quad b_k]^T$, we achieve

$$u_k = -c_k [G_k^T G_k + \upsilon_k I_k + c_k H_k^T H_k] \cdot H_k^T e_{k2} \tag{21}$$

Where I_k is an identity matrix.

Once $u_k = \begin{bmatrix} w_k^T & b_k \end{bmatrix}^T$ is obtained from (21), the *k*-th hyperplane is known. A new pattern $x \in R^n$ is assigned to class k ($k = 1,2,\cdots,K$), depending on which of the *K* hyperplanes $f_k(x) = (w_k \cdot x) + b_k$ it lies nearest to, *i.e.*, the decision function is represented as

$$x \in Class \quad k, \quad k = \arg\min_{k=1,2,\cdots,K} \frac{|(w_k \cdot x) + b_k|}{\| w_k \|} \tag{22}$$

Where $|\cdot|$ is the absolute value.

3.2. Nonlinear Multi-LSTSVM

In this subSection, we extend the linear Multi-LSTSVM to the nonlinear case by kernel trick. The input data are mapped into a high dimensional feature space by some nonlinear kernel functions. Here, we consider the following kernel-generated hyperplanes.

$$K(x^T, C^T)w_k + b_k = 0, k = 1,2,\cdots,K \tag{23}$$

Where $C = [A_1^T, A_2^T, \cdots, A_K^T]$ and K is an appropriate kernel. Similar to linear case, the nonlinear optimization problems can be expressed as

$$\begin{aligned} &\min_{w_k, b_k, \xi_k} \frac{1}{2} \| K(A_k, C^T)w_k + e_{k1}b_k \|_2^2 + \frac{\upsilon_k}{2}(\| w_k \|^2 + b_k^2) + \frac{c_k}{2}\xi_k^T \xi_k \\ &s.t. \quad -(K(B_k, C^T)w_k + e_{k2}b_k) + \xi_k = e_{k2} \end{aligned} \tag{24}$$

Where $\upsilon_k, c_k, \xi_k, e_{ki}$ $(k = 1,2,\cdots,K, i = 1,2)$ are respectively regularization parameters, penalty parameters, slack variables and the vectors of ones.

On substituting the equality constraint into the objective function, we get the unconstrained optimization problem as follows.

$$\min_{w_k, b_k} \tilde{L} = \frac{1}{2} \| K(A_k, C^T)w_k + e_{k1}b_k \|^2 + \frac{\upsilon_k}{2}(\| w_k \|^2 + b_k^2) + \frac{c_k}{2}(e_{k2} + (K(B_k, C^T)w_k + e_{k2}b_k))^2 \tag{25}$$

Differentiating $\tilde{L}$ with respect to w_k, b_k yields the following Karush-Kuhn-Tucker (KKT) conditions, we get

$$\begin{aligned} \frac{\partial \tilde{L}}{\partial w_k} &= K(A_k, C^T)^T (K(A_k, C^T)w_k + e_{k1}b_k) + \upsilon_k w_k \\ &\quad + c_k K(B_k, C^T)^T [e_{k2} + (K(B_k, C^T)w_k + e_{k2}b_k)] = 0 \end{aligned} \tag{26}$$

$$\begin{aligned} \frac{\partial \tilde{L}}{\partial b_k} &= e_{k1}^T (K(A_k, C^T)w_k + e_{k1}b_k) + \upsilon_k b_k \\ &\quad + c_k e_{k2}^T [e_{k2} + (K(B_k, C^T)w_k + e_{k2}b_k)] = 0 \end{aligned} \tag{27}$$

After combining (26) and (27), we can achieve the following equation,

$$\begin{aligned} &\begin{bmatrix} K(A_k, C^T)^T \\ e_{k1}^T \end{bmatrix} \cdot \begin{bmatrix} K(A_k, C^T) & e_{k1} \end{bmatrix} \cdot \begin{bmatrix} w_k \\ b_k \end{bmatrix} + \upsilon_k \begin{bmatrix} w_k \\ b_k \end{bmatrix} \\ &+ c_k \begin{bmatrix} K(B_k, C^T)^T \\ e_{k2}^T \end{bmatrix} \cdot \begin{bmatrix} K(B_k, C^T) & e_{k2} \end{bmatrix} \cdot \begin{bmatrix} w_k \\ b_k \end{bmatrix} + c_k \begin{bmatrix} K(B_k, C^T)^T \\ e_{k2}^T \end{bmatrix} e_{k2} = 0 \end{aligned} \tag{28}$$

Define $\tilde{G}_k = \begin{bmatrix} K(A_k, C^T) & e_{k1} \end{bmatrix}$, $\tilde{H}_k = \begin{bmatrix} K(B_k, C^T) & e_{k2} \end{bmatrix}$, $u_k = \begin{bmatrix} w_k^T & b_k \end{bmatrix}^T$, we can obtain

$$u_k = -c_k[\tilde{G}_k^T \tilde{G}_k + \upsilon_k I_k + c_k \tilde{H}_k^T \tilde{H}_k] \cdot \tilde{H}_k^T e_{k2} \tag{29}$$

Where I_k is an identity matrix.

Once $u_k = \begin{bmatrix} w_k^T & b_k \end{bmatrix}^T$ is obtained from (29), the *k*-th hyperplane is known. A new pattern $x \in R^n$ is then assigned to class k ($k = 1,2,\cdots,K$), depending on which of the *K* hyperplanes (23) it lies nearest to, *i.e.*, the decision function is represented as

$$x \in Class \quad k, \quad k = \arg\min_{k=1,2,\cdots,K} \frac{|K(x, C^T)w_k + b_k|}{\sqrt{w_k^T K(C, C^T) w_k}} \tag{30}$$

Where $|\cdot|$ is the absolute value.

4. Experimental Results

In order to evaluate our Multi-LSTSVM, we investigate its classification accuracy and computational efficiency on real-world UCI benchmark datasets and ORL, YALE face datasets. In our implementation, we focus on the comparison between our Multi-LSTSVM and several state-of-the-art binary classification methods being used for multi-class classification, including Multi-LSSVM, Multi-GEPSVM and Multi-TWSVM. All these four methods are carried out by using the "One-Versus-All" structure and implemented in MATLAB R2013a on a personal computer (PC) with an Intel (R) Core (TM) processor (3.40GHz) and 4 GB random-access memory (RAM). We perform Multi-LSSVM by employing LSSVM toolbox [28], and Multi-GEPSVM is implemented by using a MATLAB function "eig", and QPPs in Multi-TWSVM are solved by SOR algorithm, which is also used to solve QPPs in references [9, 13, 15]. Our Multi-LSTSVM is solved by operator '\' in Matlab. As for parameters selecting, the standard 10-fold cross-validation technique is employed. In addition, the parameters for all methods, including penalty parameter, regularization parameter, kernel parameter etc, are selected from the set $\{2^{-8}, 2^{-7}, \cdots, 2^7, 2^8\}$. We repeat all the experiments five times on each dataset and record the corresponding mean values.

4.1. UCI Datasets

In this subSection, in order to demonstrate the superiority of our Multi-LSTSVM over Multi-LSSVM, Multi-GEPSVM and Multi-TWSVM, we choose 9 datasets from the UCI machine learning repository [28]. For the linear case, in order to compare the performance of the proposed Multi-LSTSVM with the rest of three methods, the results of numerical experiments are listed in Table 1, including the mean and standard deviation of classification accuracies and training time. In Table 1, the best accuracy is shown by boldface and the shortest CPU time is shown by underline for each dataset. It is impressive that Multi-LSTSVM obtains better accuracy than other methods in 5 out of 9 datasets, and takes less time than other methods in 7 out of 9. Therefore, it is evident that the performance of Multi-LSTSVM is comparable or better than the other three methods. For example, for the Dermatology dataset, the accuracy of our linear Multi-LSTSVM is 97.21%, while Multi-LSSVM is 93.18%, Multi-GEPSVM is 93.80% and Multi-TWSVM is 95.36%.

For the nonlinear case, Gaussian kernel is employed, which is defined by $K(x,z) = e^{-\frac{\|x-z\|^2}{2\sigma^2}}$, σ is the kernel parameter. Table 2 displays the experimental results for four nonlinear methods on the above 9 UCI datasets. From Table 2, we can observe that in most cases, the accuracies of all the nonlinear methods are obviously better than those of the linear ones. As in Table 1, the best accuracy is shown by boldface and the shortest CPU time is shown by underline for each dataset. From Table 1 and Table 2, we can find

that Multi-TWSVM also gets comparable performance on classification in some case, but its training time is longer than our multi-LSTSVM. For example, for the Segment dataset, the training time of our linear Multi-LSTSVM is 0.0127 second, while Multi-TWSVM is 21.3856 second, and the training time of our nonlinear Multi-LSTSVM is 6.0102 second, while Multi-TWSVM is 26.55 second. Thus, our method is more suitable for large-scale dataset classification problems.

Table 1. Performance Comparison on Multi-Class Classification Accuracies Using Linear Kernel

Datasets	Multi-LSSVM	Multi-GEPSVM	Multi-TWSVM	Multi-LSTSVM
	Acc+Std(%) Time（s）	Acc+Std(%) Time（s）	Acc+Std(%) Time（s）	Acc+Std(%) Time（s）
Iris 150*4*3	73.07±1.38 T=0.0134	**97.60±0.37** T=0.0015	94.53±0.30 T=0.0216	93.20±0.30 T=0.0033
Glass 214*9*6	37.76±1.21 T=0.0279	45.70±0.21 T=0.0029	58.88±0.99 T=0.2980	**59.44±0.77** T=0.0029
Wine 178*13*3	97.98±0.31 T=0.0186	93.03±0.75 T=0.0016	98.88±0.40 T=0.0438	**99.66±0.31** T=0.0016
Zoo 101*16*7	91.49±1.33 T=0.0275	83.96±4.92 T=0.0039	**98.85±1.08** T=0.0103	96.33±1.13 T=0.0039
Vehicle 846*18*4	62.20±0.45 T=0.1517	62.48±0.52 T=0.0043	77.71±0.43 T=1.7413	**78.42±0.20** T=0.0040
Dermatology 358*34*6	93.18±0.32 T=0.0541	93.80±0.78 T=0.0095	95.36±0.75 T=0.1141	**97.21±0.20** T=0.0049
Seeds 210*7*3	94.95±0.54 T=0.0174	89.43±0.62 T=0.0014	**95.24±0.34** T=0.0398	93.81±0.48 T=0.0014
Balance 625*4*3	84.48±0.04 T=0.0611	**91.65±0.07** T=0.0019	87.46±0.65 T=0.2954	87.14±0.46 T=0.0020
Segment 2310*18*7	73.20±0.16 T=2.5432	72.52±0.13 T=0.0131	69.98±0.67 T=21.3856	**73.81±0.19** T=0.0127

4.2. Image Classification

In order to further validate the performance of our proposed Multi-LSTSVM, two popular databases ORL and YALE are selected for evaluation. In our experiments, we use the data provided by Deng Cai [http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html]. For ORL database, PCA is exploited to reduce the dimensionality of features into 50, 60, …, 110,120. For YALE database, feature dimensionality is reduced to 30, 40, …, 90, 100 by PCA. The optimal parameters are selected by 10-fold cross validation method. The classification accuracy and training time of different methods with linear kernel and nonlinear kernel are reported in Table 3 to Table 6. The best accuracy is shown by boldface and the shortest CPU time is shown by underline for each dataset.

From Table 3, Table 4 and Table 6, we can find that our Multi-LSTSVM not only gets the best accuracy but also takes the least CPU time in all cases. In Table 5, our Multi-LSTSVM also obtains comparable performance on classification in most cases, although its CPU time is more than Multi-LSSVM. To get an intuitive view of the superiority of our proposed method on ORL database using nonlinear kernel, Figure1 and Figure2 are plotted. Figure1 shows the recognition rates of different methods versus the dimensions and Figure2 depicts the CPU time of different methods. The results of Figure 1 and Figure 2 further verify the conclusion above.

Table 2. Performance Comparison on Multi-Class Classification Accuracies Using Nonlinear Kernel

Datasets	Multi-LSSVM	Multi-GEPSVM	Multi-TWSVM	Multi-LSTSVM
	Acc+Std(%) Time（s）	Acc+Std(%) Time（s）	Acc+Std(%) Time（s）	Acc+Std(%) Time（s）
Iris 150*4*3	96.53±0.56 T=0.0142	96.27±0.37 T=0.0462	97.33±0.67 T=0.0255	**98.00±0.47** T=0.0066
Glass 214*9*6	63.55±1.58 T=0.0285	56.26±1.77 T=0.2458	68.60±0.61 T=0.2688	**69.35±1.17** T=0.0241
Wine 178*13*3	97.53±0.75 T=0.0206	94.94±0.03 T=0.0821	98.99±0.25 T=0.0223	**99.89±0.25** T=0.0083
Zoo 101*16*7	94.65±0.89 T=0.0266	93.47±1.50 T=0.0497	**96.04±0.16** T=0.0162	95.84±0.44 T=0.0098
Vehicle 846*18*4	77.35±0.52 T=0.1793	.52.05±1.15 T=18.1021	**85.70±0.55** T=1.9659	85.63±0.49 T=0.3076
Dermatology 358*34*6	**96.26±0.47** T=0.0530	96.09±0.34 T=1.2260	93.69±0.58 T=0.0842	93.35±0.36 T=0.0618
Seeds 210*7*3	94.29±0.48 T=0.0188	92.86±0.34 T=0.1370	96.38±0.26 T=0.0388	**96.57±0.71** T=0.0114
Balance 625*4*3	88.77±0.21 T=0.0719	92.16±2.03 T=7.0802	**99.10±0.14** T=0.2598	92.22±0.37 T=0.1159
Segment 2310*18*7	94.20±0.12 T=2.7592	95.26±0.14 T=87.2531	96.49±0.16 T=26.5500	**96.77±0.13** T=6.0102

Table 3. Performance Comparison on ORL Database Using Linear Kernel

Dataset	Multi-LSSVM	Multi-GEPSVM	Multi-TWSVM	Multi-LSTSVM
	Acc±Std(%) Time(s)	Acc±Std(%) Time(s)	Acc±Std(%) Time(s)	Acc±Std(%) Time(s)
ORL 400*50	49.75±1.33 T=0.2761	92.25±0.64 T=0.1058	89.45±1.04 T=0.2445	**94.10±0.34** T=0.0442
ORL 400*60	57.25±1.38 T=0.2848	91.95±1.05 T=0.1434	90.55±0.65 T=0.2475	**93.80±0.84** T=0.0484
ORL 400*70	62.70±0.54 T=0.3018	91.55±0.96 T=0.2033	90.90±0.70 T=0.2537	**93.00±0.73** T=0.0534
ORL 400*80	67.60±0.96 T=0.2885	91.75±0.92 T=0.2454	91.10±1.18 T=0.2619	**94.30±0.51** T=0.0580
ORL 400*90	70.75±0.50 T=0.3066	91.50±0.73 T=0.3118	91.20±1.08 T=0.2701	**94.00±0.81** T=0.0662
ORL 400*100	74.45±0.45 T=0.2982	89.70±0.96 T=0.3923	91.05±1.60 T=0.2849	**93.95±1.01** T=0.0750
ORL 400*110	77.45±1.14 T=0.3261	88.55±1.04 T=0.4916	90.55±0.84 T=0.2958	**93.05±0.69** T=0.0841
ORL 400*120	79.60±1.65 T=0.3067	87.80±1.61 T=0.5937	90.30±1.08 T=0.3088	**92.55±0.65** T=0.0905

Table 4. Performance Comparison on YALE Database Using Linear Kernel

Dataset	Multi-LSSVM	Multi-GEPSVM	Multi-TWSVM	Multi-LSTSVM
	Acc±Std(%) Time(s)	Acc±Std(%) Time(s)	Acc±Std(%) Time(s)	Acc±Std(%) Time(s)
Yale 165*30	47.27±2.06 T=0.0575	71.64±0.51 T=0.0167	50.79±1.84 T=0.0324	**72.97±3.05** T=0.0092
Yale 165*40	53.09±2.52 T=0.0618	73.45±0.66 T=0.0229	64.85±1.05 T=0.0323	**77.82±1.95** T=0.0118
Yale 165*50	61.45±0.92 T=0.0666	73.45±2.59 T=0.0326	65.09±0.92 T=0.0341	**79.03±1.85** T=0.0130
Yale 165*60	61.82±3.64 T=0.0761	68.73±1.01 T=0.0464	64.73±1.57 T=0.0359	**77.33±0.92** T=0.0135
Yale 165*70	62.55±1.74 T=0.0809	68.85±1.95 T=0.0720	65.21±1.64 T=0.0367	**79.52±3.47** T=0.0144
Yale 165*80	60.12±1.89 T=0.0860	63.52±2.51 T=0.0906	60.36±1.10 T=0.0366	**76.48±1.63** T=0.0165
Yale 165*90	57.33±1.58 T=0.0957	56.36±2.31 T=0.1178	56.61±2.70 T=0.0388	**72.85±1.68** T=0.0208
Yale 165*100	53.94±1.21 T=0.0948	52.00±2.66 T=0.1475	52.24±2.07 T=0.0405	**67.76±2.48** T=0.0181

Table 5. Performance Comparison on ORL Database Using Nonlinear Kernel

Dataset	Multi-LSSVM	Multi-GEPSVM	Multi-TWSVM	Multi-LSTSVM
	Acc±Std(%) Time(s)	Acc±Std(%) Time(s)	Acc±Std(%) Time(s)	Acc±Std(%) Time(s)
ORL 400*50	87.45±0.99 T=0.2744	94.25±0.54 T=13.2292	92.40±0.91 T=0.8848	**94.55±0.93** T=0.5229
ORL 400*60	87.50±1.21 T=0.3017	**94..80±0.37** T=12.5873	92.20±0.86 T=0.9281	92.60±0.55 T=0.5344
ORL 400*70	85.75±1.10 T=0.3001	**93.50±0.64** T=12.4386	91.15±0.70 T=0.9375	91.75±0.90 T=0.5571
ORL 400*80	86.65±0.95 T=0.2726	**93.25±0.73** T=13.9634	92.30±1.18 T=0.9405	92.90±0.22 T=0.5507
ORL 400*90	85.30±0.37 T=0.2795	92.35±0.84 T=12.6064	90.70±1.27 T=0.9491	**92.40±0.98** T=0.5373
ORL 400*100	83.65±0.88 T=0.3174	89.05±0.97 T=11.8288	90.00±0.64 T=1.0141	**92.05±0.27** T=0.5503
ORL 400*110	82.45±1.08 T=0.3225	87.95±1.18 T=11.6987	88.40±1.55 T=0.9663	**91.10±0.38** T=0.6031
ORL 400*120	81.15±1.14 T=0.3284	87.20±0.65 T=10.1980	89.25±1.25 T=0.9532	**91.35±1.08** T=0.5825

Table 6. Performance Comparison on YALE Database Using Nonlinear Kernel

Dataset	Multi-LSSVM	Multi-GEPSVM	Multi-TWSVM	Multi-LSTSVM
	Acc±Std(%) Time(s)	Acc±Std(%) Time(s)	Acc±Std(%) Time(s)	Acc±Std(%) Time(s)
Yale 165*30	65.70±2.04 T=0.0626	60.36±1.64 T=0.4160	64.61±2.29 T=0.0621	**73.82±1.57** T=0.0424
Yale 165*40	65.94±1.00 T=0.0691	61.33±3.13 T=0.3801	71.15±2.08 T=0.0634	**76.12±1.10** T=0.0410
Yale 165*50	65.09±1.58 T=0.0725	50.79±1.38 T=0.3530	67.64±2.52 T=0.0620	**76.73±0.69** T=0.0419
Yale 165*60	64.24±2.06 T=0.0824	48.36±1.45 T=0.3382	66.06±1.66 T=0.0627	**78.06±1.08** T=0.0419
Yale 165*70	62.30±1.79 T=0.0868	44.97±1.38 T=0.3079	63.27±1.52 T=0.0628	**79.52±1.45** T=0.0427
Yale 165*80	60.48±1.57 T=0.0895	42.30±0.66 T=0.2943	59.03±1.33 T=0.0641	**77.09±2.51** T=0.0433
Yale 165*90	53.70±1.26 T=0.0935	32.97±0.69 T=0.2895	54.42±2.24 T=0.1051	**72.97±1.95** T=0.0437
Yale 165*100	48.73±2.41 T=0.1018	29.64±2.29 T=0.2829	47.76±2.69 T=0.0784	**67.03±3.11** T=0.0445

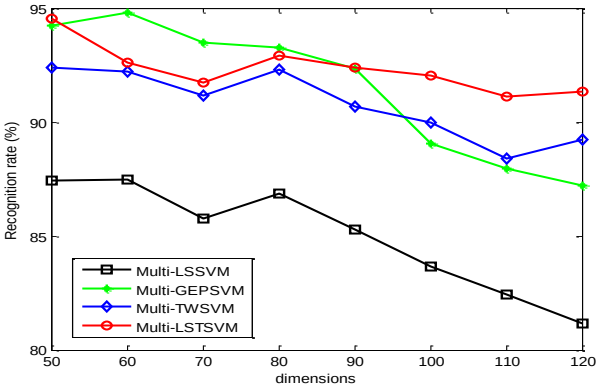

Figure 1. Recognition Rates of Different Methods versus the Dimensions on ORL Database Using Nonlinear Kernel

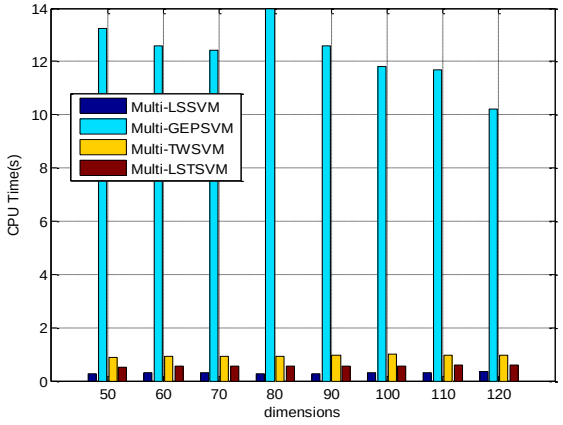

Figure 2. The CPU Times of Different Methods on ORL Database Using Nonlinear Kernel

5. Conclusions

In this paper, we extend least squares twin support vector machine (LSTSVM) to multi-class classification problem, termed as Multi-LSTSVM. Multi-LSTSVM solves *K* QPPs such that the *k*-th QPP aims at making sure the *k*-th hyperplane is closer to its own class and is as far as possible from the other class. Experimental results obtained on real-world UCI datasets, ORL and YALE face datasets illustrate that our proposed Multi-LSTSVM has comparable or better performance. Therefore, one more direction of future work is to apply our Multi-LSTSVM to other practical problems such as text categorization, image analysis and speaker recognition and so on.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under grant No. 11326052 and the Youth fund project of Anqing Normal University under grant No. KJ201308.

References

[1] C. Cortes and V. N. Vapnik, "Support vector machine", Machine Learning, vol. 20, no. 3, **(1995)**, pp. 273-297.
[2] V. N. Vapnik, "The nature of statistical learning theory", Springer-Verlag, New York Incorporated, **(2000)**.
[3] E. Osuna, R. Freund and F. Girosi, "Training support vector machines: An application to face detection", Proceedings of Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, June 17-19, **(1997)**, pp.130-136.
[4] D. Isa, L. H. Lee, V. P. Kallimani and R. Rajkumar, "Text document preprocessing with the Bayes formula for classification using the support vector machine", IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 9, **(2008)**, pp. 1264-1272.
[5] W. S. Noble, "Kernel methods in computational biology, In: Support Vector Machine Applications in Computational Biology", MIT Press, Cambridge, **(2004)**, pp.71-92.
[6] S. Zafeiriou, A. Tefas and I. Pitas, "Minimum class variance support vector machine", IEEE Transactions on Image Processing, vol. 16, no. 10, **(2007)**, pp. 2551-2564.
[7] O. L. Mangasarian and E. W. Wild, "Multisurface proximal support vector machine classification via generalized eigenvalues", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no.1, **(2006)**, pp. 69-74.
[8] J. R. Khemchandai and S. Chandra, "Twin support vector machine classification for pattern classification", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 5, **(2007)**, pp. 905-910.
[9] Y. H. Shao, C. H. Zhang, X. B. Wang and N. Y. Deng, "Improvements on twin support vector machines", IEEE Transactions on Neural Networks, vol. 22, no. 6, **(2011)**, pp. 962-968.
[10] X. B. Chen, J. Yang, Q. L. Ye and J. Liang, "Recursive projection twin support vector machine via within-class variance minimization", Pattern Recognition, vol. 44, no.10, **(2011)**, pp. 2643-2655.
[11] X. J. Peng, "TPMSVM: a novel twin parametric-margin support vector machine for pattern recognition", Pattern Recognition, vol. 44, no. 10, **(2011)**, pp. 2678-2692.
[12] Z. Q. Qi, Y. J. Tian and Y. Shi, "Robust twin support vector machine for pattern classification", Pattern Recognition, vol. 46, no. 1, **(2013)**, pp. 305-316.
[13] Y. H. Shao, Z. Wang, W. J. Chen and N. Y. Deng, "A regularization for the projection twin support vector machine", Knowledge-Based Systems, vol. 37, **(2013)**, pp. 203-210.
[14] Y. H. Shao, W. J. Chen and N. Y. Deng, "Nonparallel hyperplane support vector machine for binary classification problems", Information Sciences, vol. 263, **(2014)**, pp.22-35.
[15] Y. J. Tian, Z. Q. Qi, X. C. Ju, Y. Shi and X. H. Liu, "Nonparallel support vector machines for pattern classification", IEEE Transaction on Cybernetics, vol. 44, no. 7, **(2014)**, pp.1067-1079.
[16] M. A. Kumar and M. Gopal, "Least squares twin support vector machines for pattern classification", Expert Systems with Applications, vol. 36, no. 4, **(2009)**, pp.7535-7543.
[17] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers", Neural processing letters, vol. 9, no. 3, **(1999)**, pp. 293-300.
[18] Y. H. Shao, N. Y. Deng and Z. M. Yang, "Least squares recursive projection twin support vector machine for classification", Pattern Recognition, vol. 45, no. 6, **(2012)**, pp. 2299-2307.
[19] S. F. Ding and X. P. Hua, "Recursive least squares projection twin support vector machines for nonlinear classification", Neurocomputing, vol. 130, **(2014)**, pp. 3-9.

[20] L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon and L. D. Jackel, "Comparison of classifier methods: a case study in handwritten digit recognition", Proceedings of International Conference on Pattern Recognition, IEEE Computer Society Press, **(1994)**, pp. 77-77.
[21] Kreßel and Ulrich H. G., "Pairwise classification and support vector machines", in Advances in kernel methods, MIT Press, Cambridge, MA, **(1999)**, pp.255-268.
[22] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes", Journal of Artificial Intelligence Research, vol. 2, **(1995)**, pp. 263-286.
[23] J. Platt, N. Cristianini and J. S. Taylor, "Large margin dags for multiclass classification", In: Advances in Neural Information Processing Systems, MIT Press, Cambridge, vol. 12, **(2000)**, pp. 547-553.
[24] C. Angulo, X. Parra and A. Catala, "K-SVCR: a support vector machine for multi-class classification", Neurocomputing, vol.55, no.1, **(2003)**, pp. 57-77.\
[25] Y. T. Xu, R. Guo and L. S. Wang, "A twin multi-class classification support vector machine", Cognitive Computation, vol. 5, no. 4, **(2013)**, pp. 580-588.
[26] J. A. Nasiri, N. M. Charkari and S. Jalili, "Least squares twin multi-class classification support vector machine", Pattern Recognition, vol. 48, no. 3, **(2015)**, pp. 984-992.
[27] J. A. K. Suykens and J. Vandewalle, "Multiclass least squares support vector machines", IJCNN'99: Proceedings of International Joint Conference on Neural Networks, world Scientific, Washington, DC, vol. 2, **(1999)**, pp. 900-903.
[28] De Brabanter K., Karsmakers P., Ojeda F., Alzate C., De Brabanter J., Pelckmans K., De Moor B., Vandewalle J., Suykens J. A. K., "LS-SVMlab Toolbox User's Guide version 1.8", **(2010)**, available from < http://www.esat.kuleuven.be/sista/lssvmlab>.
[29] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases", **(1992)**. http://www.ics.uci.edu/mlearn/MLRepository.html.

Authors

Sugen Chen, He is a lecturer in Anqing Normal University, Anhui, China. He received his MS degree from HeFei University of Technology, China, in 2009. His current interests include pattern recognition and machine learning.

Juan Xu, She is a lecturer in Anqing Normal University, Anhui, China. She received her MS degree from Nanjing University of Science and Technology, China, in 2009. Her current interests include pattern recognition and image modeling.

