Pattern Recognition 48 (2015) 984–992

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Least squares twin multi-class classification support vector machine

Jalal A. Nasiri, Nasrollah Moghadam Charkari *, Saeed Jalili

Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran

ARTICLE INFO

Article history:
Received 6 July 2013
Received in revised form
20 July 2014
Accepted 18 September 2014
Available online 30 September 2014

Keywords:
Twin support vector machine
Least squares
Multi-class classification
Nonparallel plane
K-SVCR

ABSTRACT

Twin K-class support vector classification (Twin-KSVC) is a novel multi-class method based on twin support vector machine (TWSVM). In this paper, we formulate a least squares version of Twin-KSVC called as LST-KSVC. This formulation leads to extremely simple and fast algorithm. LST-KSVC, same as the Twin-KSVC, evaluates all the training data into a "1-versus-1-versus-rest" structure, so it generates ternary output $\{-1, 0, +1\}$. In LST-KSVC, the solution of the two modified primal problems is reduced to solving only two systems of linear equations whereas Twin-KSVC needs to solve two quadratic programming problems (QPPs) along with two systems of linear equations. Our experiments on UCI and face datasets indicate that the proposed method has comparable accuracy in classification to that of Twin-KSVC but with remarkably less computational time. Also, because of the structure "1-versus-1-versus-rest", the classification accuracy of LST-KSVC is higher than typical multi-class method based on SVMs.

1. Introduction

Support vector machine (SVM) was originally proposed by Vapnik [1,2] for binary classification. In contrast with other machine learning approach like artificial neural network which aims at reducing empirical risk, SVM implements the structural risk minimization (SRM) that minimizes the upper bound of generation error. SVM has been successfully applied in wide spectrum of research areas like face recognition, text categorization, and biomedicine [3–9]. One of the main challenges in the classical SVM is the high computational complexity of quadratic programming problem (QPP) [10]. The computational complexity of SVM is $O(l^3)$, where l denotes as the total size of training data. This drawback restricts the application of SVM to large-scale problem domains.

Twin support vector machines (TWSVM) were proposed by Jayadeva et al. in [11] for binary classification. TWSVM generates two nonparallel hyper-planes by solving two smaller-sized QPPs such that each hyper-plane is closer to one class and as far as possible from the other. 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 the conventional SVM [11]. Some extensions of TWSVM such as twin bounded support vector machines (TBSVM) [12], Robust TWSVM [13] and Projection TWSVM [14] have been proposed to achieve higher accuracy with lower computational time in comparing with SVM families. Least squares twin support vector machine (LS-TWSVM) [15] has been proposed as a way to replace the convex QPPs in TWSVM with a convex linear system by using a squared loss function instead of the hinge one. Inspired by LS-TWSVM, LS-PTWSVM has been introduced as a least squares version of projection twin support vector machine [16]. LS-TWSVM and LS-PTWSVM have extremely fast training speed since their separating hyper-planes are determined by solving a single system of linear equations.

SVM and TWSVM are suitable for binary classification problems. However, Multi-class classification problem is often occurred in real life. In the SVM and TWSVM family framework, "1-versus-rest" [17] and "1-versus-1" [18] approaches are usually resolve multi-class classification. In "1-versus-rest", K binary SVM classifiers are constructed. Each binary SVM is trained with all of the patterns, so it easily leads to the class imbalance problem, Whereas TWSVM address this problem. The second structure, "1-versus-1", needs to construct K(K-1)/2 binary (Twin) SVMs. Each classifier is involved with the training data of two classes. In this case, the information of the remaining samples is omitted in each binary classification. Therefore, unfavorable results may be received [19]. A new multi-class method based on "1-versus-1-versus-rest" structure called K-SVCR (support vector classification regression for K-class classification) was proposed in [19]. It constructs K(K- 1)/2 binary K-SVCR classifiers for a K-class classification. This structure provides better forecasting results. However, as all the training data are utilized in constricting the decision classification, its time complexity is higher than the former structures. Twin-KSVC based on K-SVCR and TWSVM was proposed in [20]. It takes the advantage of both TWSVM and

* Corresponding author. Tel.: +98 21 82883301.
E-mail addresses: j.nasiri@modares.ac.ir (J.A. Nasiri), charkari@modares.ac.ir (N. Moghadam Charkari), sjalili@modares.ac.ir (S. Jalili).

http://dx.doi.org/10.1016/j.patcog.2014.09.020
0031-3203/

K-SVCR. In the term of computational time, Twin-KSVC requires nearly the same run-time as "1-versus-rest" structure of TWSVM, while its runtime is far lower than K-SVCR.

In this paper, following the line of research in [11,15,20], we propose a least squares version of Twin-KSVC, called least squares twin K-class support vector classification (LST-KSVC) using the strategy of LS-TWSVM and K-SVCR. The QPPs of our LST-KSVC have only equality constraints while inequality constraints appear in the Twin-KSVC. Thus, the solution of LST-KSVC follows directly from solving two systems of linear equation as opposed to solving two QPPs and two systems of linear equation in Twin-KSVC. It takes both the advantages of LS-TWSVM in time complexity and K-SVCR in higher multi-class classification accuracy based on "1-versus-1-versus-rest" structure. The experimental results on benchmark datasets show that the proposed LST-KSVC has comparable classification accuracy to that of the Twin-KSVC but with remarkably less computational time. In addition, the proposed algorithm can properly cope with large dataset without any external optimizers.

This paper is organized as follows. Section 2 briefly dwells on the TWSVM, K-SVCR and Twin-KSVC. LST-KSVC is formulated and described in Section 3, which includes linear, nonlinear cases and classification decision rule. Section 4 provides some interesting experimental results on datasets to investigate our proposed multi-class algorithm and concluding remarks are given in Section 5.

2. Preliminaries

In this section, we give a brief description of TWSVM and Multi-Class SVM based on "1-versus-1-versus-rest" structure for classification purposes.

2.1. TWSVM

TWSVM is a binary classifier that performs classification by the use of two non-parallel hyperplanes unlike SVM which used a single hyperplane [11]. Let us consider dataset D which d^+ is training set with label $+1$ and d^- is training set with label -1 in the m-dimensional real space R^m. Let matrix $A \in R^{d^+ \times m}$ represent the training data belong $+1$ and matrix $B \in R^{d^- \times m}$ represents the training data belong to the class -1. The linear TWSVM search for two non-parallel hyper-planes in R^m as follows:

$$x^T w_{(1)} + b_{(1)} = 0 \quad \text{and} \quad x^T w_{(2)} + b_{(2)} = 0 \tag{1}$$

Such that each hyperplane is closest to the training data of one class and farthest from the training data of another class. A new data sample is assigned to class $+1$ or -1 depends on which of the two planes is closest to it. The linear TWSVM solves two QPPs (2) and (3) with objective function corresponding to one class and constraints corresponding to the other class.

$$\min_{w_{(1)},b_{(1)}} \ \frac{1}{2}\|Aw_{(1)} + e_1 b_{(1)}\|^2 + c_1 e_2^T \lambda_2$$
$$\text{s.t.} \quad -(Bw_{(1)} + e_2 b_{(1)}) + \lambda_2 \geq e_2, \quad \lambda_2 \geq 0. \tag{2}$$

and

$$\min_{w_{(2)},b_{(2)}} \ \frac{1}{2}\|Bw_{(2)} + e_2 b_{(2)}\|^2 + c_2 e_1^T \lambda_1$$
$$\text{s.t.} \quad (Aw_{(2)} + e_2 b_{(2)}) + \lambda_1 \geq e_1, \lambda_1 \geq 0. \tag{3}$$

where c_1, $c_2 > 0$ are penalty parameters, e_1 and e_2 are vectors of ones of appropriate dimensions and λ_1 and λ_2 are vectors of slack variables respectively. Let $P = [B\ e_2]$ and $Q = [A\ e_1]$. The Wolf dual problems of (2) and (3) have been shown to be

$$\max_{\alpha} \ e_2^T \alpha - \frac{1}{2}\alpha^T P \left(Q^T Q\right)^{-1} P^T \alpha$$
$$\text{s.t.} \quad 0 \leq \alpha \leq c_1 e_2, \tag{4}$$

and

$$\max_{\alpha} \ e_1^T \beta - \frac{1}{2}\beta^T Q \left(P^T P\right)^{-1} Q^T \beta$$
$$\text{s.t.} \quad 0 \leq \beta \leq c_2 e_1, \tag{5}$$

where Lagrangian multipliers are $\alpha \in R^{m_2}$ and $\beta \in R^{m_1}$. In order to deal with the case when $P^T P$ or $Q^T Q$ becomes singular and to avoid the possible ill-conditioning of $P^T P$ and $Q^T Q$, TWSVM introduces a term εI ($\varepsilon > 0$) where I is an identity matrix of appropriate dimensions. The non-parallel hyperplanes (1) can be obtained from the solutions α and β of (4) and (5) by

$$z_1 = -\left(Q^T Q + \varepsilon I\right)^{-1} P^T \alpha \quad \text{and} \quad z_2 = -\left(P^T P + \varepsilon I\right)^{-1} Q^T \beta, \tag{6}$$

where $z_{(i)} = \left[w_{(i)}^T\ b_{(i)}\right]^T$, $(i = 1, 2)$.

A new point $x \in R^m$ is assigned to class i ($i = +1, -1$), depending on which of the two hyperplanes in (1) is closer to, i.e.

$$Class(i) = \arg\min_{i=1,2} \frac{|x^T w_{(i)} + b_{(i)}|}{\|w_{(i)}\|} \tag{7}$$

where $|.|$ is the absolute value.

TWSVM was also extended in [11] to handle nonlinear kernels by considering two non-parallel kernel generated surfaces.

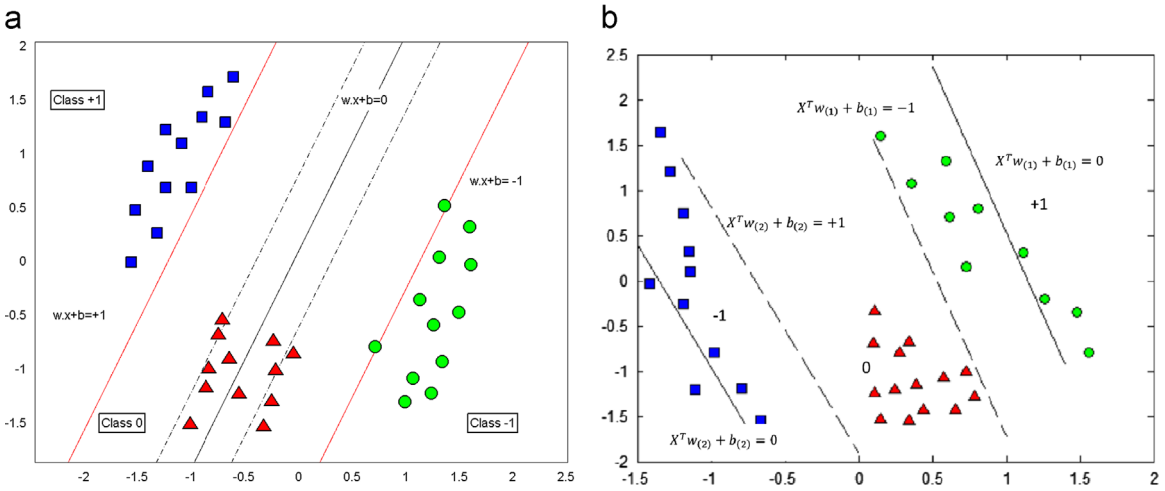

Fig. 1. Illustration of multi-class SVM and TWSVM with ternary output $\{-1, 0, +1\}$: (a) K-SVCR and (b) Twin-KSVC.

2.2. Multi-class SVMs

In this subsection, we briefly introduce two multi-class methods based on "1-versus-1-versus-rest" structure for the training set.

K-SVCR: a multi-class classification support vector machine: K-SVCR was introduced in [14] based on SVM theory and ternary output {−1, 0, +1}. As illustrated in Fig. 1(a), it evaluates all the training data in the decomposition phase by using mixed classification and regression machine formulation. K-SVCR can be achieved by solving the following QPP:

$$\begin{aligned}\min_{w,b,\xi,n,n^*}\ & \frac{1}{2}\|w\|^2+c_1\sum_{i=1}^{l}\xi_i+c_2\sum_{i=1}^{o}(n_i+n_i^*)\\ \text{s.t}\ & y_i(w.\phi(x_i)+b)\geq 1-\xi_i,\quad i=1,2,\ldots,l.\\ & -\varepsilon-n_i^*<w.\phi(x_i)+b,\quad i=1,2\ldots,o.\\ & w.\phi(x_i)+b\leq\varepsilon+n_i,\quad i=1,2\ldots,o.\\ & \xi_i,n_i,n_i^*\geq 0,\end{aligned}\tag{8}$$

where ξ_i, n_i and n_i^* are slack variables. The positive parameter ε is restricted to be lower than 1 to avoid overlapping. l is the number of patterns belong to the two classes to be separated and o is the number of patterns belong to other classes that is labeled "0". The hyperplane decision function could be written as

$$f(X)=\begin{cases}+1 & \text{if } \sum_{i=1}^{no.sv}\alpha_i\ k(x_i,x)+b\geq\varepsilon\\ -1 & \text{if } \sum_{i=1}^{no.sv}\alpha_i\ k(x_i,x)+b\leq-\varepsilon\\ 0 & \text{otherwise}\end{cases}\tag{9}$$

In the above α_i and $no.sv$ are Lagrangian multipliers and the number of support vectors respectively. K-SVCR improved the standard two-class classifiers in the multi-classification structure since all the data are given full consideration while the focus of algorithm is a two-class partition, but (8) is composed of a single larger-sized QPP. Moreover, the two focused kinds of samples are used once in constraints, whereas the remaining samples are used twice in constraints, which leads to low computational speed.

Twin multi-class classification support vector machine: Twin-KSVC [20] is a new multi-class classification based on TWSVM formulation. Like K-SVCR, the proposed algorithm evaluates all training points into a "1-versus-1-versus-rest" structure with ternary output {−1, 0, +1}. It finds two nonparallel planes for each of two kinds of samples selected from K classes. The rest of the samples are mapped into a region between the two nonparallel planes.

Let matrix $A\in R^{l_1\times m}$ represent the training data belong to "+1", $B\in R^{l_2\times m}$ represents the training data belong to the class "−1" and $C\in R^{l_3\times m}$ indicates the rest training data which are labeled "0". The patterns are denoted in rows and the features of each pattern are shown as m column. The two nonparallel hyperplanes are defined as follows:

$$x^Tw_{(1)}+b_{(1)}=0\quad\text{and}\quad x^Tw_{(2)}+b_{(2)}=0\tag{10}$$

They can be obtained by resolving the following pair of QPPs,

$$\begin{aligned}\underset{w_{(1)},b_{(1)},\xi,\eta}{\text{Min}}\ & \frac{1}{2}\|Aw_{(1)}+e_1b_{(1)}\|^2+c_1e_2^T\xi+c_2e_3^T\eta\\ \text{s.t.}\ & -(Bw_{(1)}+e_2b_{(1)})+\xi\geq e_2,\\ & -(Cw_{(1)}+e_3b_{(1)})+\eta\geq e_3(1-\varepsilon),\\ & \xi\geq 0e,\quad \eta\geq 0e,\end{aligned}\tag{11}$$

and

$$\begin{aligned}\underset{w_{(2)},b_{(2)},\xi^*,\eta^*}{\text{Min}}\ & \frac{1}{2}\|Bw_{(2)}+e_2b_{(2)}\|^2+c_3e_1^T\xi^*+c_4e_3^T\eta^*\\ \text{s.t.}\ & (Aw_{(2)}+e_1b_{(2)})+\xi^*\geq e_1,\\ & (Cw_{(2)}+e_3b_{(2)})+\eta^*\geq e_3(1-\varepsilon),\\ & \xi^*\geq 0e,\quad \eta^*\geq 0e,\end{aligned}\tag{12}$$

where ε is a positive parameter chosen prior while the other parameters are same as defined in TWSVM. Twin-KSVC seeks to nonparallel in (10). Meanwhile, it obtains two corresponding hyperplanes $x^Tw_1+b_1=-1$ and $x^Tw_2+b_2=+1$, and they are at a distance of 1 from two focused hyperplane. As illustrated in Fig. 1(b), these two hyperplanes divide the whole plane into three parts. Twin-KSVC classifies sample points according to which region they belong. By introducing the lagrangian vectors, the dual QPPs of (11) and (12) can be represented as follows:

$$\begin{aligned}\max_{\gamma}\ & -\frac{1}{2}\gamma^TN\left(H^TH\right)^{-1}N^T\gamma+e_4^T\gamma\\ \text{s.t.}\ & 0\leq\gamma\leq F,\end{aligned}\tag{13}$$

where $H=[A\ e_1]$, $G=[B\ e_2]$, $M=[C\ e_3]$, $N=[G;M]$, $F=[c_1e_2;c_2e_3]$ and $e_4=[e_2;e_3(1-\epsilon)]$

$$\begin{aligned}\max_{\rho}\ & -\frac{1}{2}\rho^TP\left(G^TG\right)^{-1}P^T\gamma+e_5^T\rho\\ \text{s.t.}\ & 0\leq\rho\leq F^*,\end{aligned}\tag{14}$$

where, $P=[H;M]$, $F^*=[c_3e_1;c_4e_3]$, $e_5=[e_1;e_3(1-\epsilon)]$.

For a new testing point x_i, Twin-KSVC determines its class label by the following decision function.

$$f(x_i)=\begin{cases}+1 & \text{if } x_iw_{(1)}+eb_{(1)}>-1+\varepsilon\\ -1 & \text{if } x_iw_{(2)}+eb_{(2)}<1-\varepsilon\\ 0 & \text{otherwise}\end{cases}\tag{15}$$

It is shown that Twin-KSVC requires nearly the same run-time as "1-versus-rest" structure of TWSVM. Moreover, it requires far lower computational time than K-SVCR [20].

3. Least squared twin multi-class support vector machine

In this section, we introduce a least squares version of Twin-KSVC called least squared twin K-class support vector classification (LST-KSVC). Following the idea of PSVM that proposed in [21], the decision function of LST-KSVC is obtained by the primal problem directly. Similar to Twin-KSVC, let matrix $A\in R^{l_1\times m}$ represent the training data belong to "+1", $B\in R^{l_2\times m}$ represents the training data belong to the class "−1" and $C\in R^{l_3\times m}$ indicates the rest training data which are labeled "0".

3.1. Linear LST-KSVC

We modify the primal problem (11) of linear Twin-KSVC in least squares sense at (16), with the inequality constraint replaced with equality constraints as follows:

$$\begin{aligned}\underset{w_{(1)},b_{(1)}}{\text{Min}}\ & \frac{1}{2}\|Aw_{(1)}+e_1b_{(1)}\|^2+\frac{c_1}{2}y^Ty+\frac{c_2}{2}z^Tz\\ \text{s.t.}\ & -(Bw_{(1)}+e_2b_{(1)})+y=e_2,\\ & -(Cw_{(1)}+e_3b_{(1)})+z=e_3(1-\varepsilon),\end{aligned}\tag{16}$$

Note that the loss function in (16) is the square of 2-norm of slack variables y and z with weights $c_1/2$ and $c_2/2$ instead of 1-norm of y and z with weights c_1 and c_2 as used in (11), which makes the constraint $y\geq 0$ and $z\geq 0$ redundant [33]. This simple modification allows us to solve the QPPs (16) by solving a simultaneous system of linear equations. By substituting the equality constraints into the objective function of QPP (16) we obtain:

$$\begin{aligned}\underset{w_{(1)},b_{(1)}}{\text{Min}}\ & \frac{1}{2}\|Aw_{(1)}+e_1b_{(1)}\|^2+\frac{c_1}{2}\|Bw_{(1)}+e_2b_{(1)}+e_2\|^2\\ & +\frac{c_1}{2}\|Cw_{(1)}+e_3b_{(1)}+e_3(1-\varepsilon)\|^2\end{aligned}\tag{17}$$

Setting the gradient of (17) with respect to $w_{(1)}$ and $b_{(1)}$ to zero gives

$$A^T(Aw_{(1)}+e_1b_{(1)})+c_1B^T(Bw_{(1)}+e_2b_{(1)}+e_2) +c_2C^T(Cw_{(1)}+e_3b_{(1)}+e_3(1-\varepsilon))=0, \quad (18)$$

$$e_1^T(Aw_{(1)}+e_1b_{(1)})+c_1e_2^T(Bw_{(1)}+e_2b_{(1)}+e_2) +c_2e_3^T(Cw_{(1)}+e_3b_{(1)}+e_3(1-\varepsilon))=0, \quad (19)$$

Arranging (18) and (19) in matrix form and solving for $w_{(1)}$ and $b_{(1)}$ gives

$$c_1\begin{bmatrix}B^TB & B^Te_2\\ e_2^TB & l_2\end{bmatrix}\begin{bmatrix}w_{(1)}\\ b_{(1)}\end{bmatrix}+\begin{bmatrix}A^TA & A^Te_1\\ e_1^TA & l_1\end{bmatrix}\begin{bmatrix}w_{(1)}\\ b_{(1)}\end{bmatrix} +c_2\begin{bmatrix}C^TC & C^Te_3\\ e_3^TC & l_3\end{bmatrix}\begin{bmatrix}w_{(1)}\\ b_{(1)}\end{bmatrix}+\begin{bmatrix}c_1B^Te_2+c_2C^Te_3(1-\varepsilon)\\ c_1l_2+c_2l_3(1-\varepsilon)\end{bmatrix}=0 \quad (20)$$

$$\begin{bmatrix}w_{(1)}\\ b_{(1)}\end{bmatrix}=\begin{bmatrix}c_1B^TB+A^TA+c_2C^TC & c_1B^Te_2+A^Te_1+c_2C^Te_3\\ c_1e_2^TB+e_1^TA+c_2e_3^TC & c_1l_2+l_1+c_2l_3\end{bmatrix}^{-1} \times\begin{bmatrix}-c_1B^Te_2-c_2C^Te_3(1-\varepsilon)\\ -c_1l_2-c_2l_3(1-\varepsilon)\end{bmatrix} \quad (21)$$

$$\begin{bmatrix}w_{(1)}\\ b_{(1)}\end{bmatrix}=-\left[c_1\begin{bmatrix}B^T\\ e_2^T\end{bmatrix}[B\ \ e_2]+\begin{bmatrix}A^T\\ e_1^T\end{bmatrix}[A\ \ e_1]+c_2\begin{bmatrix}C^T\\ e_3^T\end{bmatrix}[C\ \ e_3]\right]^{-1} \times\left[c_1\begin{bmatrix}B^Te_2\\ l_2\end{bmatrix}+c_2\begin{bmatrix}C^Te_3(1-\varepsilon)\\ l_3(1-\varepsilon)\end{bmatrix}\right] \quad (22)$$

Lets $E=[A\ e_1]$, $F=[B\ e_2]$ and $G=[C\ e_3]$, the solution becomes:

$$\begin{bmatrix}w_{(1)}\\ b_{(1)}\end{bmatrix}=-\left(c_1F^TF+E^TE+c_2G^TG\right)^{-1}\left(c_1F^Te_5+c_2G^Te_6(1-\varepsilon)\right). \quad (23)$$

similarly, the solution of QPP (24) can be shown to be (25) as follows:

$$\begin{aligned}\underset{w_{(2)},b_{(2)}}{\text{Min}}\quad & \frac{1}{2}\|Bw_{(2)}+e_2b_{(2)}\|^2+\frac{c_3}{2}y^Ty+\frac{c_4}{2}z^Tz\\ \text{s.t.}\quad & (Aw_{(2)}+e_1b_{(2)})+y=e_1,\\ & (Cw_{(2)}+e_3b_{(2)})+z=e_3(1-\varepsilon),\end{aligned} \quad (24)$$

$$\begin{bmatrix}w_{(2)}\\ b_{(2)}\end{bmatrix}=\left(c_3E^TE+F^TF+c_4G^TG\right)^{-1}\left(c_3E^Te_4+c_4G^Te_6(1-\varepsilon)\right). \quad (25)$$

In this regard, two nonparallel separating hyperplanes of (10) are obtained. The linear LST-KSVC completely solves the classification problem with just two matrix inverses of much smaller dimensional matrix rather than solving two QPPs in Twin-KSVC or two larger sized of QPPs in K-SVCR.

3.2. Nonlinear LST-KSVC

We have extended the linear LST-KSVC to the nonlinear one by considering the following kernel generated surfaces:

$$K\left(x^T,D^T\right)u_{(1)}+\gamma_{(1)}=0 \quad \text{and} \quad K\left(x^T,D^T\right)u_{(2)}+\gamma_{(2)}=0 \quad (26)$$

where, $D=[A;B;C]$ and K is an arbitrary kernel. The primal QPPs of the nonlinear LST-KSVC can be modified in the same way with 2-norm of slack variables and equality constraints corresponding to surfaces (26) are given in (27) and (28).

$$\begin{aligned}\min_{u_{(1)},\gamma_{(1)}}\quad & \frac{1}{2}\|K\left(A,D^T\right)u_{(1)}+e_1\gamma_{(1)}\|^2+\frac{c_1}{2}y^Ty+\frac{c_2}{2}z^Tz\\ \text{s.t.}\quad & -\left(K\left(B,D^T\right)u_{(1)}+e_2\gamma_{(1)}\right)+y=e_2,\\ & -\left(K\left(C,D^T\right)u_{(1)}+e_3\gamma_{(1)}\right)+z=e_3(1-\varepsilon),\end{aligned} \quad (27)$$

and

$$\begin{aligned}\min_{u_{(2)},\gamma_{(2)}}\quad & \frac{1}{2}\|K\left(B,D^T\right)u_{(2)}+e_2\gamma_{(2)}\|^2+\frac{c_3}{2}y^Ty+\frac{c_4}{2}z^Tz\\ \text{s.t.}\quad & \left(K\left(A,D^T\right)u_{(2)}+e_1\gamma_{(2)}\right)+y=e_1,\\ & \left(K\left(C,D^T\right)u_{(2)}+e_3\gamma_{(2)}\right)+z=e_3(1-\varepsilon),\end{aligned} \quad (28)$$

By substituting the constraints into the objective function, these QPPs become

$$\underset{u_{(1)},\gamma_{(1)}}{\text{Min}}\quad \frac{1}{2}\|K\left(A,D^T\right)u_{(1)}+e_1\gamma_{(1)}\|^2+\frac{c_1}{2}\|K\left(B,D^T\right)u_{(1)}+e_2\gamma_{(1)}+e_2\|^2 +\frac{c_2}{2}\|K\left(C,D^T\right)u_{(1)}+e_3\gamma_{(1)}+e_3(1-\varepsilon)\|^2 \quad (29)$$

$$\underset{u_{(2)},\gamma_{(2)}}{\text{Min}}\quad \frac{1}{2}\|K\left(B,D^T\right)u_{(2)}+e_2\gamma_{(2)}\|^2+\frac{c_3}{2}\|-K\left(A,D^T\right)u_{(2)}-e_1\gamma_{(2)}+e_1\|^2 +\frac{c_4}{2}\|-K\left(C,D^T\right)u_{(2)}-e_3\gamma_{(2)}+e_3(1-\varepsilon)\|^2 \quad (30)$$

The solution of QPPs (29) and (30) can be derived as

$$\begin{bmatrix}u_{(1)}\\ \gamma_{(1)}\end{bmatrix}=-\left(c_1N^TN+M^TM+c_2O^TO\right)^{-1}\left(c_1N^Te_5+c_2O^Te_6(1-\varepsilon)\right) \quad (31)$$

$$\begin{bmatrix}u_{(2)}\\ \gamma_{(2)}\end{bmatrix}=\left(c_3M^TM+N^TN+c_4O^TO\right)^{-1}\left(c_3M^Te_4+c_4O^Te_6(1-\varepsilon)\right) \quad (32)$$

where $M=[K\left(A,D^T\right)\ e_1]$, $N=[K\left(B,D^T\right)\ e_2]$, and $O=[K\left(C,D^T\right)\ e_3]$. It can be noted that the solution of nonlinear LST-KSVC requires inversion of matrix of size $(l+1)\times(l+1)$ twice. Therefore, to reduce the computation cost, the Sherman–Morrison–Woodbury (SMW) formula [22] is used to recast (31) and (32) as

$$\begin{bmatrix}u_{(1)}\\ \gamma_{(1)}\end{bmatrix}=-\left(Z-ZN^T\left(\frac{I}{c_1}+NZN^T\right)^{-1}NZ\right) \times\left(c_1N^Te_5+c_2O^Te_6(1-\varepsilon)\right) \quad (33)$$

$$\begin{bmatrix}u_{(2)}\\ \gamma_{(2)}\end{bmatrix}=\left(F-FM^T\left(\frac{I}{c_3}+MFM^T\right)^{-1}MF\right) \times\left(c_3M^Te_4+c_4O^Te_6(1-\varepsilon)\right) \quad (34)$$

where $Z=(M^TM+c_2O^TO)^{-1}$ and $F=(N^TN+c_4O^TO)^{-1}$ can be found using SMW formula as

$$Z=\frac{1}{c_2}\left(Y-YM^T\left(c_2I+MYM^T\right)^{-1}MY\right) \quad (35)$$

$$F=\frac{1}{c_4}\left(Y-YN^T\left(c_4I+NYN^T\right)^{-1}NY\right) \quad (36)$$

here $Y=(O^TO)^{-1}$. Regarding to [11,15], we use a regularization term $\alpha I, \alpha>0$ to Y to take care of problems due to the possible ill-conditioning of O^TO.

$$Y=\frac{1}{\alpha}\left(I-O^T\left(\alpha I+OO^T\right)^{-1}O\right) \quad (37)$$

3.3. Classification decision rule

As shown in Sections 3.1 and 3.2, LST-KSVC evaluates all training points into the "1-versus-1-versus-rest" structure with ternary output $\{-1,\ 0,\ +1\}$. For a new testing point x_i, we determine its class label by the following decision function in

the linear case:

$$f(x_i)=\begin{cases}+1 & \text{if } x_i w_{(1)}+eb_{(1)}>-1+\varepsilon\\ -1 & \text{if } x_i w_{(2)}+eb_{(2)}<1-\varepsilon\\ 0 & \text{otherwise}\end{cases} \tag{38}$$

In the case of nonlinear LST-KSVC, the corresponding decision function is designed as

$$f(x_i)=\begin{cases}+1 & \text{if } K\left(x_i,D^T\right)u_{(1)}+e\gamma_{(1)}>-1+\varepsilon\\ -1 & \text{if } K\left(x_i,D^T\right)u_{(2)}+e\gamma_{(2)}<1-\varepsilon\\ 0 & \text{otherwise}\end{cases} \tag{39}$$

In the "1-versus-1-versus-rest" structure, the proposed method constructs $K(k\text{-}1)/2$ LST-KSVC classifiers for K-class classification. Each (i,j)-LST-KSVC is trained over all the pattern set while considering "specialized training" on two classes from ensemble [19,20]. The labels "+1", "−1", and "0" are assigned to samples of class(i), class(j), and all remaining classes, respectively. For a new testing point x_i, a vote is given to the class(i) or class(j) based on which condition is satisfied. Finally, the given testing point x_i is assigned to the class label that gets the most votes.

3.4. Computational complexity

It has been shown that the computational complexity of SVM is $O(l^3)$ and TWSVM is $O(l^3/4)$, where l is the total size of training data points, which implies that TWSVM is approximately 4 times faster than SVM [11]. As it has been discussed in [20], the learning run-time of Twin-KSVC is approximately 4 times faster than K-SVCR.

The linear LST-KSVC completely solved with just two matrix inversions with order of $(m+1)\times(m+1)$ where m is the dimension of the input space. The computing time of LST-KSVC is related to the sample dimensionality (m). However, in Twin-KSVC, it is related to the training set size (l) where $m \ll l$.

In nonlinear LST-KSVC , it can be found that the inverses of the matrices with size $(l+1)\times(l+1)$ is required. Sherman–Morrison–Woodbury (SMW) [22] formula have been utilized to reduce the computational cost. We show that formula (31) can be solved by (33), (35) and (37) using three inverses of smaller dimension $(l_1\times l_1)$, $(l_2\times l_2)$ and $(l_3\times l_3)$.

In addition, the rectangular kernel technique [23] in nonlinear least squares algorithm [15,16,21] and nonlinear LST-KSVC have been utilized to reduce the computational cost. Rectangular kernel technique reduces the $l\times l$ dimensionality of kernel matrices to a much smaller $l\times\bar{l}$ where $\bar{l}$ is small as 1% of l. It is worth mentioning that the reduced kernel technique slightly affects on computational time of Twin-KSVC, TWSVM and K-SVCR since in QPPs, the computing time is related to the number of training set.

4. Numerical experiments

Artificial, image (face [24]), several publicly UCI [25] and NDC data generator [26] datasets are used to show the ability of our LST-KSVC. All experiments except those shown in Table 5, have been implemented in Matlab 7.9 on a PC with system configuration Intel core2 Duo CPU at 2.53 GHz with 4 GB of RAM , and Windows 7 operating system. The results of Table 5 have been utilized from a PC with system configuration Intel Core i7 CPU

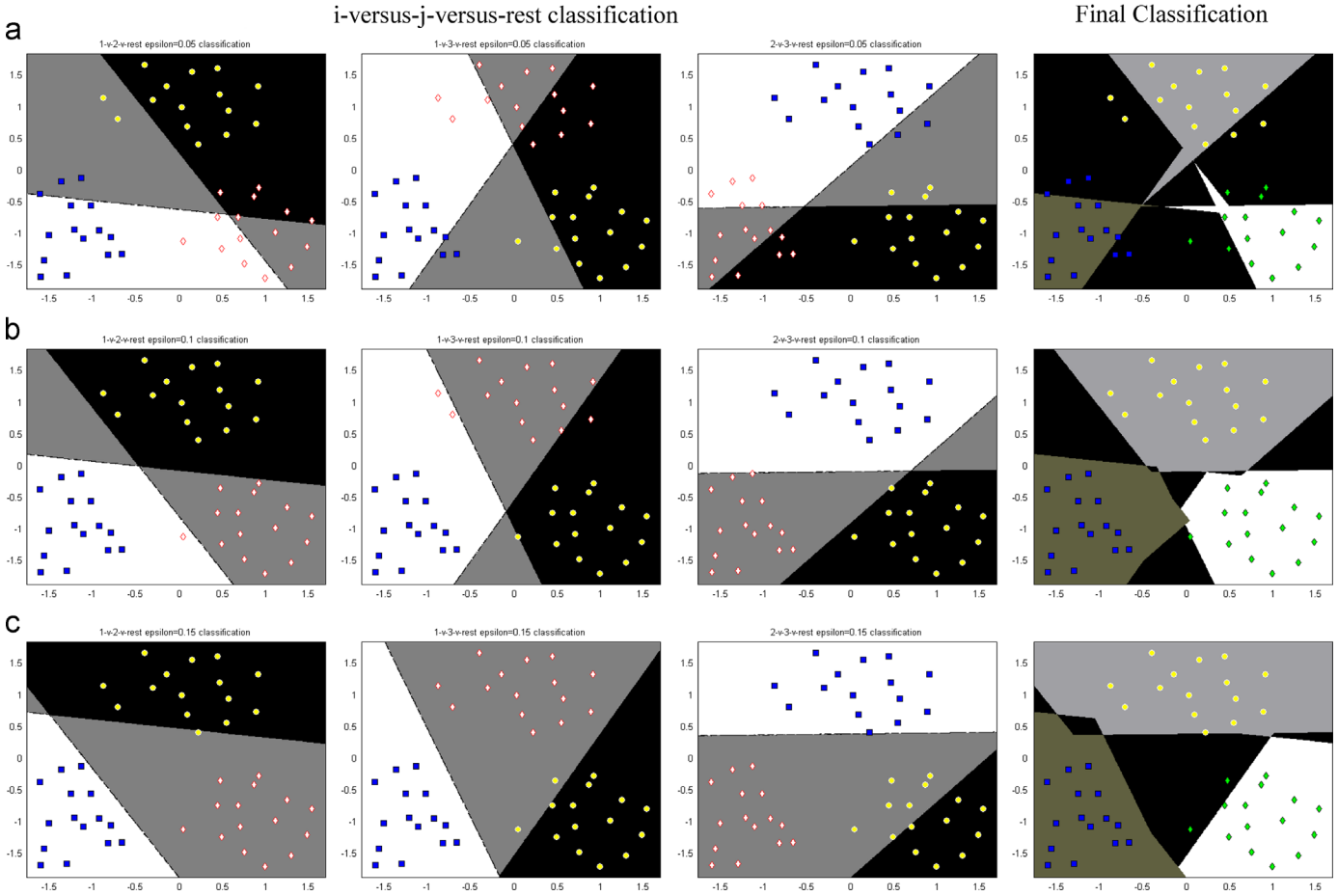

Fig. 2. Illustration of LST-KSVC with different ε insensitivity levels for linear kernel; (a) $\varepsilon=0.05$; (b) $\varepsilon=0.10$; and (c) $\varepsilon=0.15$.

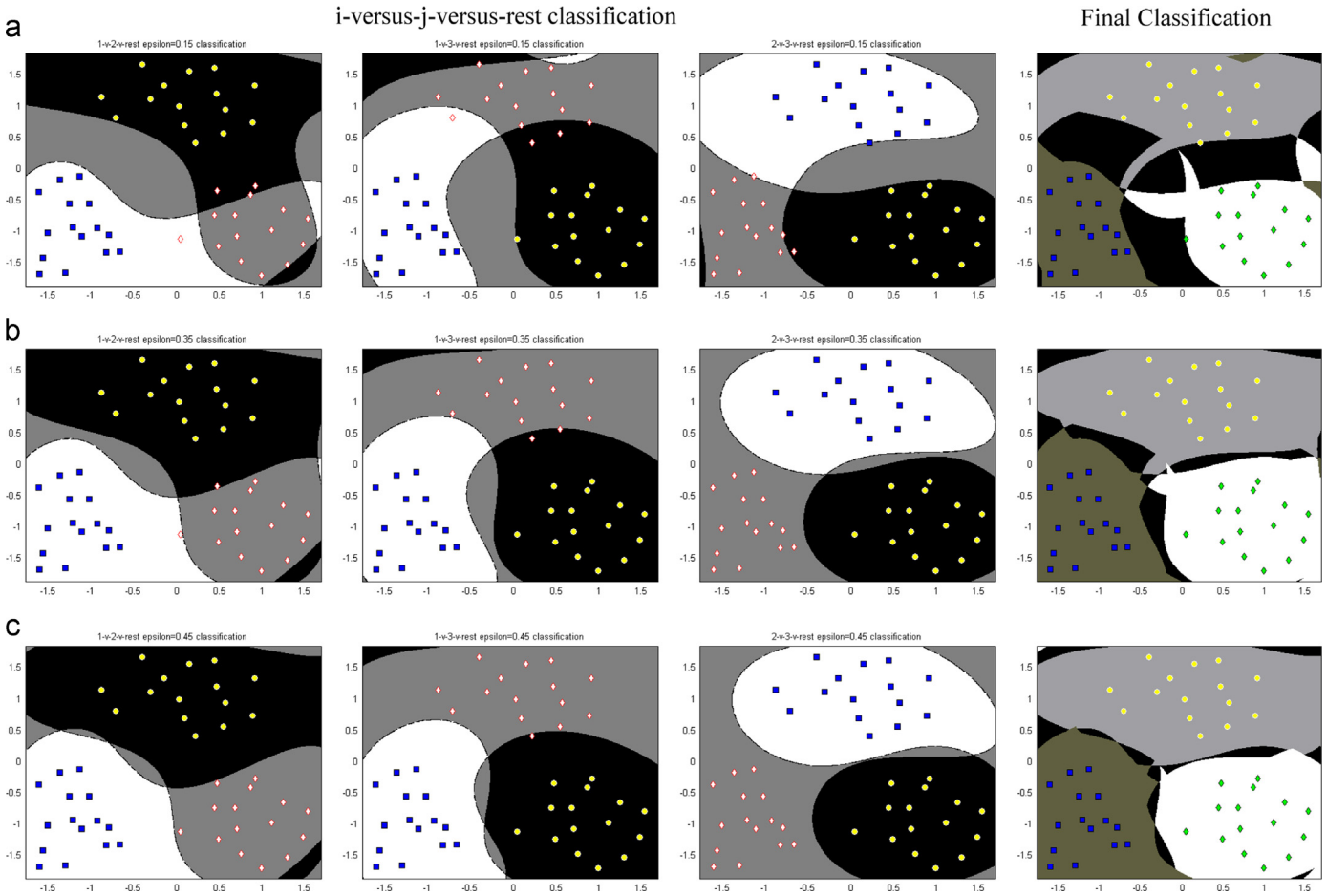

Fig. 3. Illustration of LST-KSVC with different ε insensitivity levels for polynomial kernel; (a) ε=0.15; (b) ε=0.35; and (c) ε=0.45.

Table 1
Performance comparison on face datasets.

Dataset	TWSVM (c_1, c_2) Acc ± Std Time (s)	Twin-KSVC (c_1, c_2, ε) Acc ± Std Time (s)	LST-KSVC (c_1, c_2, ε) Acc ± Std Time (s)
Face-20	$2^{-1}, 2^{1}$ 63.209 ± 6.33 27.0486	$2^{-3}, 2^{0}, 0.4$ 65.083 ± 1.67 5708.6	$2^{3}, 2^{-5}, 0.25$ **66.325 ± 1.09** **0.0110**
Face-50	$2^{5}, 2^{3}$ 73.954 ± 3.9 24.4925	$2^{-5}, 2^{1}$, 0.2 74.831 ± 0.27 5558.9	$2^{4}, 2^{0}$, 0.13 **77.298 ± 1.39** **0.0656**
Face-100	$2^{1}, 2^{3}$ 84.314 ± 5.9 19.2034	$2^{1}, 2^{-5}$, 0.1 **87.734** ± 3.35 2286.4	$2^{3}, 2^{-5}$, 0.3 86.032 ± 1.45 **0.0717**
Face-150	$2^{1}, 2^{3}$ 87.732 ± 5.1 18.6568	$2^{5}, 2^{-6}$, 0.2 89.072 ± 0.51 1157.7	$2^{5}, 2^{-5}$, 0.2 **89.9 ± 0.10** **0.3208**

3.40 GHz and 14 GB RAM. For resolving QPPs to obtain the optimal solution "quadprog.m" function is utilized. In the implementation of our LST-KSVC, as LS-TSVM [15] and LSPTSVM [16], the involved system of linear equations is realized by MATLAB operation "\".

4.1. Artificial dataset

To illustrate the effect of ε parameter on the final classification results, a two-dimensional artificial data set has been generated with 45 patterns equally distributed in three classes. Fig. 2 and 3 show three possible i-versus-j-versus rests LST-KSVC and the final decision function with ternary output $\{-1, 0, +1\}$ when the insensitivity level (ε) is increased. Linear kernel and polynomial kernel of degree 4 have been used in Figs. 2 and 3 respectively. In the case of i-versus-j-versus-rest sub figures; white, black and gray regions belong to class i, j and the rest classes respectively and in final classification sub figures black region does not belong to any class. We can observe that ε (insensitivity) parameter, which is in the constraints of the 0-labelled patterns, has a high effect on optimal hyperplane determination. It is evidence that the optimal value of ε is dependent on dataset that is similar to the C parameter in SVM.

4.2. Face database

Illumination variation is one of the challenging issues in face recognition. We conducted our method in face Illumination variation. Two popular databases Yale B and Extended Yale B [24] have been used for evaluation. In our experiments, only 64 frontal images per person under different illumination conditions are considered. After combining the Extended Yale B with the Yale B, there are 2414 images of 38 subjects named as the Completed Yale B. The images are divided into 5 subsets according to the light source direction and the camera axis. In this section, each subset is called a lighting class. We directly use the cropped and aligned images provided by [27], the size of images is 192 × 168 pixels. The lighting class of each test face image is estimated by using TWSVM, Twin-KSVC and LST-KSVC.

Similar to [27], discrete cosine transform (DCT) was applied to extract proper features. PCA was applied for reducing the dimensions of the features into 20, 50, 100, and 150. The optimal parameters are selected by 10-fold cross validation method. The classification accuracy and training time of different methods with linear kernel are reported in Table 1.

Table 2
Performance comparison of Multi-class algorithms with RBF kernel.

Dataset	K-SVCR $(c_1, c_2, \gamma, \varepsilon)$ Acc ± std Time (s)	TWSVM (c_1, c_2, γ) Acc ± std Time (s)	Twin-KSVC $(c_1, c_2, \gamma, \varepsilon)$ Acc ± std Time (s)	LST-KSVC $(c_1, c_2, \gamma, \varepsilon)$ Acc ± std Time (s)
Teaching evaluation	$2^0, 2^{-2}, 2^{-4}, 0.9$	$2^6, 2^6, 2^{-2}$	$2^0, 2^{-2}, 2^{-2}, 0.2$	$2^{-4}, 2^3, 2^4, 0.1$
$151 \times 5 \times 3$	64.47 ± 26.91	69.76 ± 28.33	71.01 ± 23.07	**72.33 ± 5.38**
	1.8578	0.2552	0.2888	**0.0087**
Iris	$2^2, 2^2, 2^{-2}, 0.5$	$2^{-2}, 2^0, 2^{-2}$	$2^0, 2^0, 2^2, 0.2$	$2^5, 2^5, 2^{-1}, 0.15$
$150 \times 4 \times 3$	98.0 ± 2.26	96.00 ± 3.26	98.13 ± 2.66	**99.27 ± 1.01**
	1.9864	0.2580	0.3037	**0.0193**
Wine	$2^4, 2^4, 2^4, 0.1$	$2^0, 2^6, 2^{-2}$	$2^4, 2^0, 2^6, 0.1$	$2^8, 2^4, 2^8, 0.12$
$178 \times 13 \times 3$	97.70 ± 1.12	92.22 ± 4.77	97.75 ± 3.24	94.27 ± 3.77
	2.7624	0.2413	0.8916	**0.0125**
Soybean	$2^2, 2^0, 2^2, 0.1$	$2^0, 2^{-2}, 2^2$	$2^2, 2^{-4}, 2^2, 0.2$	$2^2, 2^7, 2^4, 0.2$
$47 \times 35 \times 4$	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
	0.2935	0.2507	0.4022	**0.0032**
Ecoli	$2^0, 2^0, 2^6, 0.5$	$2^{-4}, 2^{-2}, 2^6$	$2^0, 2^2, 2^4, 0.2$	$2^1, 2^{-5}, 2^4, 0.4$
$327 \times 7 \times 5$	79.32 ± 4.62	78.22 ± 5.24	86.36 ± 4.51	**88.89 ± 1.16**
	75.39	1.1499	5.5621	**0.2007**
Glass	$2^0, 2^2, 2^{-4}, 0.1$	$2^{-2}, 2^8, 2^{-4}$	$2^0, 2^0, 2^2, 0.2$	$2^2, 2^{-1}, 2^6, 0.2$
$214 \times 9 \times 6$	57.85 ± 9.56	52.86 ± 5.08	63.21 ± 4.83	**65.76 ± 2.00**
	32.1611	2.6340	11.5605	**0.1179**
Car	$2^0, 2^0, 2^4, 0.3$	$2^{-2}, 2^0, 2^4$	$2^{-2}, 2^6, 2^4, 0.2$	$2^6, 2^{-5}, 2^2, 0.1$
$1728 \times 6 \times 4$	71.8 ± 5.28	71.8 ± 4.68	72.3 ± 5.04	**94.13 ± 0.17**
	6477.1	306.7994	128.1238	**13.5458**
Balance	$2^0, 2^0, 2^3, 0.4$	$2^{-2}, 2^2, 2^4$	$2^2, 2^3, 2^6, 0.2$	$2^{-3}, 2^3, 2^4, 0.3$
$625 \times 4 \times 3$	89.35 ± 6.02	90.42 ± 5.69	90.33 ± 5.48	**96.83 ± 5.18**
	56.88	8.3999	50.5070	**0.2910**
Dermatology	$2^{-1}, 2^{-1}, 2^2, 0.4$	$2^{-2}, 2^2, 2^2$	$2^1, 2^{-2}, 2^2, 0.1$	$2^0, 2^5, 2^5, 0.1$
$358 \times 34 \times 6$	84.18 ± 2.13	84.08 ± 2.06	84.26 ± 2.01	**95.18 ± 1.54**
	584.14	1.6727	7.1127	**0.3523**

As it is shown, LST-LSVC performed several orders of magnitude faster than TWSVM and Twin-KSVC on all dimensions of face datasets while the accuracy has been kept similar to other methods.

4.3. UCI datasets

In the third experiment, we conduct some experiments on nine benchmark datasets: teaching evaluation, Iris, Wine, Soybean, Ecoli, Glass, Car, Balance, and Dermatology from the UCI machine learning Repository [25].

4.3.1. Parameter selection

It is clear that the performance of SVMs and TSVMs methods depends heavily on the choices of parameters [20]. In the experiment, 5-fold cross validation and the Gaussian kernel function $k(x_i, x_j) = \exp(-\|x_i - x_j\|^2/\gamma^2)$ are selected to evaluate the performance of different multiclass classification. The optimal values for the parameters were found by the grid search method. In Twin-KSVC and LST-KSVC, we set $c_1 = c_3$ and $c_2 = c_4$ to reduce the computational complexity of parameter selection. The optimal values for c parameters in the all methods were selected from the range of $\{2^i | i = -4, -2, 0, 1, 2, 4, 6, 8\}$ and Gaussian kernel parameter γ was selected from $\{2^i | i = -4, -2, 0, 2, 4, 6, 8\}$.

In K-SVCR, the optimal value of parameter ε ranged from set $\{0.1, 0.3, \ldots, 0.9\}$ but in Twin-KSVC and LST-KSVC this parameter was set to a small value and chosen from set $\{0, \ldots, 0.4\}$. The result of the former three methods comes from [20].

4.3.2. Result comparisons and discussion

From the result of Table 2, it is found that the performances of Twin-KSVC and LST-KSVC are better than that of the original TWSVM. It emphasizes the necessity of designing multi-class TWSVM base model for the multi-class classification. It also reveals that the classification accuracy of LST-KSVC and Twin-KSVC is almost the same except for the Car, Balance, and Dermatology data sets.

There are two kinds of constraints in each QPP of Twin-KSVC (focused class and other class). In general, the number of samples of focused class is smaller than class "other". It becomes worse in imbalanced data set. In addition, it is not easy to take into account unequal misclassification costs (c_1, c_2, c_3, c_4) in the large samples as well as large class number datasets. So, Twin-KSVC may lead to suboptimal solutions in this situation [28].

Different classification techniques employ different loss functions to get better classification accuracy [29]. Twin-KSVC minimizes the hinge loss function and LST-KSVC uses the least square (LS) loss function. LST-KSVC does not have quadratic programming problem (QPP) and classification solution is learned by two linear equations. It seems that LST-KSVC is able to reduce the above-mentioned problems in Twin-KSVC by using analytical solution.

Characteristics of UCI datasets summarized in Table 3. The class imbalance of the datasets is indicated by the high ratio between the number of instances of the majority class and the minority class. The scale of search space for finding the optimal misclassification costs is defined by the number of samples × class. For Car dataset, the imbalance ratio and scale of search space is 18.91 and 6912, respectively. The mentioned parameter is much larger than other UCI datasets where the accuracy improved by 21.83% in LST-KSVC. The Dermatology dataset shows that the imbalance ratio and scale of search space is 5.63 and 2148, respectively. In this case, the accuracy improved by 10.92%. In the Teaching evaluate, Iris, Wine, and Soybean UCI datasets the imbalance ratios are much smaller, so the classification accuracy of LST-KSVC and Twin-KSVC is almost the same.

By keeping same or better accuracy, LST-KSVC performed several orders of magnitude faster than K-SVCR, TWSVM and

Table 3
Characteristics of the UCI data sets.

Data set	Scale	Data per class	Imbalance ratio	Improvement (%)
Teaching	151 × 3=453	49, 50, 52	1.06	1.33
Iris	150 × 3=450	All 50	1	1.14
Wine	178 × 3=534	59, 71, 48	1.48	−3.48
Soybean	47 × 4=188	10, 10, 10, 17	1.71	0
Ecoli	327 × 5=1635	143, 77, 35, 20, 52	7.16	2.53
Glass	214 × 6=1284	70, 76, 17, 13, 9, 29	8.45	2.55
Car	1728 × 4=6912	1210, 384, 69 65	18.91	21.83
Balance	625 × 3=1875	49, 288, 288	5.89	6.5
Dermatology	358 × 6=2148	111, 60, 71, 48, 48, 20	5.63	10.92

Twin-KSVC on all datasets. It is worth mentioning that LST-KSVC does not require any special optimizers whereas others need. In this experiment, fast interior point solvers of Mosek optimization toolbox for MATLAB have been used. For Car dataset our method is able to train the classifier in 13.5 s whereas the training time of Twin-KSVC, TWSVM and K-SVCR on the same dataset are 128.1 s, 306.7 s and 6471.1 s respectively. The results undoubtedly prove the boost of computational efficiency of LST-KSVC over K-SVCR, TWSVM, and Twin-KSVC.

Table 4
Comparison on NDC datasets with RBF kernel.

NDC datasets	K-SVCR Time (s)	TWSVM Time (s)	Twin-KSVC Time (s)	LST-KSVC Time (s)
100 × 32 × 3	1.5471	0.20608	0.51786	0.00332
500 × 32 × 3	75.3495	1.4213	2.1915	0.19932
1000 × 32 × 3	674.8210	6.6804	11.0879	1.1002
5000 × 32 × 3	[a]	255.5694	350.1599	73.3299
10 k[c] × 32 × 3	[a]	796.8556	1242.31	5.032
50 k[c] × 32 × 3	[a]	[b]	[b]	6.1745
100 k[d] × 32 × 3	[b]	[b]	[b]	1.1601

[a] Experiment was stopped as computing time was very high.
[b] Terminate because of out of memory.
[c] The rectangular kernel with ratio of 1% was used.
[d] The rectangular kernel with ratio of 0.1% was used.

Table 5
The effect of different sample ratios of rectangular kernel $K=(l,\bar{l})$.

Sample ratio	$\bar{l}=10\%$ Accuracy% Time (s)	$\bar{l}=20\%$ Accuracy% Time (s)	$\bar{l}=30\%$ Accuracy% Time (s)	$\bar{l}=40\%$ Accuracy% Time (s)	$\bar{l}=50\%$ Accuracy% Time (s)
5000 × 32 × 3	90.89 0.18	91.33 0.78	86.04 1.76	75.93 3.17	66.44 4.96

4.4. NDC datasets

To further show the advantage of our LST-KSVC in the training speed, we have compared the CPU time in the training process of our LST-KSVC with SVCR, TWSVM and Twin-KSVC on large datasets. We extended the NDC database generator for multi class classification [26] and produce several 3 class datasets with the size increased from 10^2 to 10^6, while the feature number was fixed at 32. The NDC datasets are divided into a training set and prediction set. We report the training speed and prediction accuracy, respectively.

The parameters of all algorithms have been fixed in advance ($c_{1,2,3,4}=1$, $\gamma=0.125$ and $\varepsilon=0.1$). For the NDC dataset with 10k, 50k and 100k samples, we have employed rectangular kernel [23] with 1%,1% and 0.1% of total data points respectively. According to Table 4, we found that when the training size increases, the proposed LST-KSVC becomes much faster than SVCR, TWSVM and Twin-KSVC. For example, LST-KSVC easily classified 100k patterns with 32 features in 1.16 s.

Table 5 shows how different sample ratios affect on prediction accuracy. Although the parameters of LST-KSVC have been fixed in advance ($c_{1,2,3,4}=1$, $\gamma=2^{-17}$ and $\varepsilon=0.3$), but results confirm that using rectangular kernel not only makes large dataset tractable, but it also leads to improved generalization by avoiding data over fitting [23].

5. Conclusion

Twin K-class support vector classification (Twin-KSVC) is a novel multi-class method based on twin support vector machine (TWSVM). In this paper, we formulate a least squares version of Twin-KSVC called as LST-KSVC for multi-class classification. This formulation leads to extremely simple and fast algorithm. LST-KSVC, similar to the Twin-KSVC, evaluates all the training data into a "1-versus-1-versus-rest" structure, so it generates ternary output $\{-1,\ 0,\ +1\}$. In LST-KSVC, we solve two primal problems by solving just two systems of linear equations. This allows LST-KSVC to classify large datasets in shorter time, that is not true for Twin-KSVC which requires large training time. Computational results on Face, UCI and NDC datasets demonstrate that our LST-KSVC obtains classification accuracy comparable to that of Twin-KSVC, but at reduced computational effort. There are four parameters in our LST-KSVC same as Twin-KSVC, so the parameter selection is a practical problem and should be addressed in the future.

Conflict of interest

None declared.

Acknowledgments

We thank the anonymous reviewers for their valuable suggestions. This research is partially supported by ITRC (Iran Telecommunication Research Center) under contract No. 6979/500.

References

[1] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[2] V.N. Vapnik, V. Vapnik, Statistical Learning Theory, vol. 2.
[3] N.M. Khan, R. Ksantini, I.S. Ahmad, B. Boufama, A novel SVM+ NDA model for classification with an application to face recognition, Pattern Recognit. 45 (1) (2012) 66–79.
[4] M.M. Adankon, M. Cheriet, Model selection for the LS-SVM. Application to handwriting recognition, Pattern Recognit. 42 (12) (2009) 3264–3270.
[5] Y.-C. Wu, Y.-S. Lee, J.-C. Yang, Robust and efficient multiclass SVM models for phrase pattern recognition, Pattern Recognit. 41 (9) (2008) 2874–2889.
[6] R. Liu, Y. Wang, T. Baba, D. Masumoto, S. Nagata, SVM-based active feedback in image retrieval using clustering and unlabeled data, Pattern Recognit. 41 (8) (2008) 2645–2655.
[7] Z. Xue, D. Ming, W. Song, B. Wan, S. Jin, Infrared gait recognition based on wavelet transform and support vector machine, Pattern Recognit. 43 (8) (2010) 2904–2910.
[8] S. Li, J.T. Kwok, H. Zhu, Y. Wang, Texture classification using the support vector machines, Pattern Recognit. 36 (12) (2003) 2883–2893.
[9] X.-Y. Wang, T. Wang, J. Bu, Color image segmentation using pixel wise support vector machine classification, Pattern Recognit. 44 (4) (2011) 777–787.

[10] X. Peng, TPMSVM: a novel twin parametric-margin support vector machine for pattern recognition, Pattern Recognit. 44 (10) (2011) 2678–2692.
[11] R. Khemchandani, S. Chandra, et al., Twin support vector machines for pattern classification, IEEE Trans. Pattern Anal. Mach. Intell. 29 (5) (2007) 905–910.
[12] Y.-H. Shao, C.-H. Zhang, X.-B. Wang, N.-Y. Deng, Improvements on twin support vector machines, IEEE Trans. Neural Netw. 22 (6) (2011) 962–968.
[13] Z. Qi, Y. Tian, Y. Shi, Robust twin support vector machine for pattern classification, Pattern Recognit. 46 (1) (2013) 305–316.
[14] X. Chen, J. Yang, Q. Ye, J. Liang, Recursive projection twin support vector machine via within-class variance minimization, Pattern Recognit. 44 (10) (2011) 2643–2655.
[15] M. Arun Kumar, M. Gopal, Least squares twin support vector machines for pattern classification, Expert Syst. Appl. 36 (4) (2009) 7535–7543.
[16] Y.-H. Shao, N.-Y. Deng, Z.-M. Yang, Least squares recursive projection twin support vector machine for classification, Pattern Recognit. 45 (6) (2012) 2299–2307.
[17] L. Bottou, C. Cortes, J.S. Denker, H. Drucker, I. Guyon, L.D. Jackel, Y. LeCun, U.A. Muller, E. Sackinger, P. Simard, et al., Comparison of classifier methods: a case study in handwritten digit recognition, in: International Conference on Pattern Recognition, IEEE Computer Society Press, 1994, p. 77.
[18] U.H.-G. Kreßel, Pairwise classification and support vector machines, in: Advances in Kernel Methods, MIT Press, Cambridge, MA, 1999, pp. 255–268.
[19] C. Angulo, X. Parra, A. Catala, SVCR: a support vector machine for multi-class classification, Neurocomputing 55 (1) (2003) 57–77.
[20] Y. Xu, R. Guo, L. Wang, A twin multi-class classification support vector machine, Cogn. Comput. 5 (4) (2013) 580–588.
[21] G. Fung, O.L. Mangasarian, Proximal support vector machine classifiers, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2001, pp. 77–86.
[22] G.H. Golub, C.F. van Loan, Matrix Computations, 3rd ed., John Hopkins University Press, Baltimore, London, 2012.
[23] Y.-J. Lee, S.-Y. Huang, Reduced support vector machines: a statistical theory, IEEE Trans. Neural Netw. 18 (1) (2007) 1–13.
[24] D.J. Jobson, Z.-U. Rahman, G.A. Woodell, Properties and performance of a center/surround retinex, IEEE Trans. Image Process. 6 (3) (1997) 451–462.
[25] A. Asuncion, D. Newman, Uci machine learning repository, 2007.
[26] D. Musicant, NDC: Normally Distributed Clustered Datasets, Computer Sciences Department, University of Wisconsin, Madison.
[27] W. Chen, M.J. Er, S. Wu, Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain, IEEE Trans. Syst. Man Cybern. B: Cybern. 36 (2) (2006) 458–466.
[28] H.-N. Qu, G.-Z. Li, W.-S. Xu, An asymmetric classifier based on partial least squares, Pattern Recognit. 43 (10) (2010) 3448–3457.
[29] S. Suresh, N. Sundararajan, P. Saratchandran, Risk-sensitive loss functions for sparse multi-category classification problems, Inf. Sci. 178 (12) (2008) 2621–2638.

Jalal A. Nasiri received BS degree in Computer Engineering from IAUM, in 2006 and received the M.S degree in Computer Engineering at Ferdowsi University of Mashhad, Iran, in 2009. Currently he is a PhD candidate of computer engineering in Tarbiat Modares University. His research includes support vector machines, optimization theory and application.

Nasrollah Moghadam Charkari received his BS in computer science from Shahid Beheshti University, Tehran, Iran in 1986 and his MS and PhD degrees in computer engineering and information systems engineering from Yamanashi University, Japan in 1992 and 1995, respectively. Currently he is an assistant professor in faculty of electrical and computer engineering, Tarbiat Modares University, Tehran, Iran. His research interests include machine learning, computer vision, and image processing.

Saeed Jalili received the PhD degree from Bradford University (UK) in 1991 and the MSc degree in computer science from Sharif University of Technology in 1985. Since 1992, he has been an Associate Professor at Tarbiat Modares University (TMU). His main research interests are Machine Learning, Privacy Preserving Data Publishing, and Quantitative Evaluation of Software Architecture.