J. Cent. South Univ. (2015) 22: 2638−2645
DOI: 10.1007/s11771-015-2794-1

Springer

Least squares weighted twin support vector machines with local information

HUA Xiao-peng(花小朋)[1, 2], XU Sen(徐森)[1], LI Xian-feng(李先锋)[1]

1. School of Information Engineering, Yancheng Institute of Technology, Yancheng 224051, China;
2. School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China

Abstract: A least squares version of the recently proposed weighted twin support vector machine with local information (WLTSVM) for binary classification is formulated. This formulation leads to an extremely simple and fast algorithm, called least squares weighted twin support vector machine with local information (LSWLTSVM), for generating binary classifiers based on two non-parallel hyperplanes. Two modified primal problems of WLTSVM are attempted to solve, instead of two dual problems usually solved. The solution of the two modified problems reduces to solving just two systems of linear equations as opposed to solving two quadratic programming problems along with two systems of linear equations in WLTSVM. Moreover, two extra modifications were proposed in LSWLTSVM to improve the generalization capability. One is that a hot kernel function, not the simple-minded definition in WLTSVM, is used to define the weight matrix of adjacency graph, which ensures that the underlying similarity information between any pair of data points in the same class can be fully reflected. The other is that the weight for each point in the contrary class is considered in constructing equality constraints, which makes LSWLTSVM less sensitive to noise points than WLTSVM. Experimental results indicate that LSWLTSVM has comparable classification accuracy to that of WLTSVM but with remarkably less computational time.

Key words: least squares; similarity information; hot kernel function; noise points

1 Introduction

MANGASARIAN and WILD [1] proposed a fast classifier for binary classification, termed generalized eigenvalues proximal SVM (GEPSVM), which is an extension of proximal SVM (PSVM) [2]. The geometric interpretation of GEPSVM is that each plane is the closest to the samples for its own class and at the same time is the furthest from the samples for the other classes [1]. Such a method has lower computational complexity and works better on XOR examples in comparison to standard SVM [1].

During the past few years, a family of novel nonparallel hyperplane classifiers has been developed to improve the generalization of GEPSVM. JAYADEVA et al [3] introduced a stand-alone nonparallel hyperplane classifier, called twin support vector machine (TWSVM). This algorithm aims at generating two nonparallel hyperplanes such that each one is closer to one class and is at least one far from the other class for any given binary data set [3]. The strategy of solving a pair of smaller sized quadratic programming problems (QPPs) instead of a large one as in a standard SVM makes the learning speed of TWSVM approximately four times faster than standard SVM [3]. The experimental results in Ref. [3] have shown that TWSVM compares favorably with SVM and GEPSVM in terms of generalization performance. Some extensions to TWSVM include the least squares TWSVM (LSTWSVM) [4], localized TWSVM (LCTSVM) [5], twin parametric-margin SVM (TPMSVM) [6], structural TWSVM (STWSVM) [7], twin mahalanobios distance-based SVM (TMSVM) [8], and twin bounded SVM (TBSVM) [9]. Different from TWSVM which improves GEPSVM by seeking a hyperplane for each class using SVM-type formulation, a multi-weight vector projection support vector machine (MVSVM) [10] was proposed to enhance the performance of GEPSVM by seeking one weight vector, such that the data points of one class are the closest to its class mean while the data points of different classes are seprarated as far as possible. Later, the projection twin support vector machine (PTSVM) was proposed in the light of MVSVM and TWSVM [11]. Experimental results in Ref. [11] show that PTSVM has comparable or better performance compared with GEPSVM, TWSVM and MVSVM. In Ref. [12] PTSVM was further extended to the nolinear classification which was ignored in Ref. [11].

Foundation item: Project(61105057) supported by the National Natural Science Foundation of China; Project(13KJB520024) supported by the Natural Science Foundation of Jiangsu Higher Education Institutes of China; Project supported by Jiangsu Province Qing Lan Project, China
Received date: 2014−06−03; **Accepted date:** 2014−09−05
Corresponding author: HUA Xiao-peng, Associate Professor, Doctoral Candidate; Tel: +86−13921805878; E-mail: xp_hua@163.com

Unfortunately, TWSVM and its extension algorithms fail to exploit similarity information between any pair of data points [13]. In fact, most of the samples of a data set are highly correlated, at least locally, or the data set has an inherent geometrical property [13]. Motivated by this conclusion, YE et al [13] developed a novel classification method, referred to as WLTSVM, which stands for weighted twin support vector machine with local information. By making full use of similarity information in terms of data affinity, WLTSVM uses two graphs (intra-class graph and inter-class graph) to characterize the intra-class compactness and inter-class separability, respectively [13]. The experimental results in Ref. [13] have revealed that WLTSVM is better than TWSVM in terms of both classification performance and lower computational cost. However, although WLTSVM performs faster than TWSVM, a limitation is that it can not handle large-scale problems, since its solution still follows from solving two QPPs along with two systems of linear equations. Moreover, WLTSVM uses a simple-minded definition to define the weight matrix of within-class adjacency graph. This may lead to poor generalization ability because the underlying similarity information between any pair of data points in the same class cannot be fully reflected [14−15].

In this work, we enhance WLTSVM to least squares WLTSVM (LSWLTSVM). First, we modify the primal problems of WLTSVM in least squares sense and solve them with equality constraints instead of inequalities of WLTSVM. As a result, the solution of LSWLTSVM follows directly from solving two systems of linear equations as opposed to solving two QPPs in WLTSVM and the algorithm can accurately solve large datasets without any external optimizers. Secondly, we use a hot kernel function, not the simple-minded definition in WLTSVM, to define the weight matrix of within-class adjacency graph. This strategy ensures that the underlying similarity information between any pair of data points in the same class can be fully reflected and results in better generalization ability. Thirdly, we consider the weight of each point in the contrary class in contructing equality constraints, which makes LSWLTSVM be less sensitive to noise.

2 Related work

2.1 TWSVM

Let the binary samples be classified and denoted by a set of N column vectors $\boldsymbol{x}_i$, (i=1, 2, …, N) in the n-dimensional real space $\mathbf{R}^n$, and $y_i \in \{1, 2\}$ denote the class to which the ith sample belongs. Without loss of generality, we assume that the matrix $\boldsymbol{X}_1=[\boldsymbol{x}_1^{(1)},\cdots,\boldsymbol{x}_{N_1}^{(1)}]^{\mathrm{T}}$ with size of $N_1\times n$ represents the data points of class 1 (class +1), the matrix $\boldsymbol{X}_2=[\boldsymbol{x}_1^{(2)},\cdots,\boldsymbol{x}_{N_2}^{(2)}]^{\mathrm{T}}$ with size of $N_2\times n$ represents the data points of class 2 (class −1), and the matrix $\boldsymbol{X}$ with size of $N\times n$ represents the all training data points, where $N=N_1+N_2$. All vectors will be column vectors unless transformed to a row vector by a prime superscript T.

For the linear case, TWSVM determines two nonparallel hyperplanes [3]:

$$\boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{x}+b_1=0,\ \boldsymbol{w}_2^{\mathrm{T}}\boldsymbol{x}+b_2=0 \tag{1}$$

where $\boldsymbol{w}_1$, $\boldsymbol{w}_2\in\mathbf{R}^n$, b_1, $b_2\in\mathbf{R}$. Here, each hyperplane is closer to one of the two classes and is at least one distance from the other. A new data point $\boldsymbol{x}$ is assigned to the class +1 or class −1 depending upon its proximity to the two nonparallel hyperplanes. Formally, for finding the positive and negative hyperplanes, TWSVM solves the following two primal problems [3]:

$$\text{TWSVM-1:}\begin{cases}\min & \dfrac{1}{2}\left\|\boldsymbol{X}_1\boldsymbol{w}_1+\boldsymbol{e}_1b_1\right\|_2^2+c_1\boldsymbol{e}_2^{\mathrm{T}}\boldsymbol{\xi}_2\\ \text{s.t.:} & -(\boldsymbol{X}_2\boldsymbol{w}_1+\boldsymbol{e}_2b_1)+\boldsymbol{\xi}_2\geq\boldsymbol{e}_2,\quad \boldsymbol{\xi}_2\geq 0\end{cases} \tag{2}$$

and

$$\text{TWSVM-2:}\begin{cases}\min & \dfrac{1}{2}\left\|\boldsymbol{X}_2\boldsymbol{w}_2+\boldsymbol{e}_2b_2\right\|_2^2+c_2\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{\xi}_1\\ \text{s.t.:} & (\boldsymbol{X}_1\boldsymbol{w}_2+\boldsymbol{e}_1b_2)+\boldsymbol{\xi}_1\geq\boldsymbol{e}_1,\quad \boldsymbol{\xi}_1\geq 0\end{cases} \tag{3}$$

where $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are vectors of ones of appropriate dimensions; ξ_1 and ξ_2 are the slack vectors; c_1 and c_2 are two nonnegative penalty coefficients; $\|\cdot\|_2$ is the 2-norm.

Let $\boldsymbol{H}=[\boldsymbol{X}_1\ \ \boldsymbol{e}_1]$, $\boldsymbol{G}=[\boldsymbol{X}_2\ \ \boldsymbol{e}_2]$, and $\boldsymbol{v}_i=[\boldsymbol{w}_i^{\mathrm{T}}\ \ b_i]^{\mathrm{T}}$, i=1, 2. The Wolfe's dual problems of Eqs. (2) and (3) are given by Eqs. (4) and (5) in terms of the Lagrangian multipliers $\boldsymbol{\alpha}\in\mathbf{R}^{N_2}$ and $\boldsymbol{\gamma}\in\mathbf{R}^{N_1}$, respectively.

$$\text{DTWSVM-1:}\begin{cases}\min & \dfrac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{G}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{\alpha}-\boldsymbol{e}_2^{\mathrm{T}}\boldsymbol{\alpha}\\ \text{s.t.:} & 0\leq\boldsymbol{\alpha}\leq\boldsymbol{e}_2c_1\end{cases} \tag{4}$$

$$\text{DTWSVM-2:}\begin{cases}\min & \dfrac{1}{2}\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\gamma}-\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{\gamma}\\ \text{s.t.:} & 0\leq\boldsymbol{\gamma}\leq\boldsymbol{e}_1c_2\end{cases} \tag{5}$$

The nonparallel proximal hperplanes are obtained from the solutions $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ of Eqs. (4) and (5) by

$$\boldsymbol{v}_1=\begin{bmatrix}\boldsymbol{w}_1\\ b_1\end{bmatrix}=-(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{\alpha} \tag{6}$$

and

$$\boldsymbol{v}_2=\begin{bmatrix}\boldsymbol{w}_2\\ b_2\end{bmatrix}=(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\gamma} \tag{7}$$

In order to deal with the case when $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}$ or $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}$ is singular and avoid the possible ill-conditioning, the inverse matrices $(\boldsymbol{H}'\boldsymbol{H})^{-1}$ and $(\boldsymbol{G}'\boldsymbol{G})^{-1}$ are approximately replaced by $(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}+\varepsilon\boldsymbol{I})^{-1}$ and $(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}+\varepsilon\boldsymbol{I})^{-1}$ respectively [3], where $\boldsymbol{I}$ is an identity matrix of appropriate

dimensions; and ε is a positive scalar, small to keep the structure of data.

For the nonlinear case, we can refer to Ref. [3].

2.2 WLTSVM

Obviously, TWSVM only considers the global geometry structure of the data space from Eqs. (2) and (3), and ignores the similarity information between any pair of data points. By making full use of similarity information in terms of data affinity, WLTSVM constructs two graphs for each of the TWSVM pair, a within-class graph G_s and a between-class graph G_d to model the intra-class compactness and inter-class separability, respectively [13].

Given any pair of points ($\boldsymbol{x}_i$, $\boldsymbol{x}_j$) in the same class and an arbitrary point $\boldsymbol{x}_l$ in the contrary class, the weight matrics for G_s and G_d are respectively defined as

$$\boldsymbol{W}_{s,ij}=\begin{cases}1, \text{ if } \boldsymbol{x}_i \text{ is } k_1 \text{ nearest neighbors of } \boldsymbol{x}_j \text{ or}\\ \quad \boldsymbol{x}_j \text{ is } k_1 \text{ nearest neighbors of } \boldsymbol{x}_i\\ 0, \text{otherwise}\end{cases} \tag{8}$$

and

$$\boldsymbol{W}_{d,il}=\begin{cases}1, \text{ if } \boldsymbol{x}_l \text{ is } k_2 \text{ nearest neighbors of } \boldsymbol{x}_i\\ 0, \text{otherwise}\end{cases} \tag{9}$$

For the linear case, WLTSVM solves the following two primal problems [13]:

WLTSVM-1:

$$\begin{cases}\min \quad \dfrac{1}{2}\sum\limits_{i=1}^{N_1}\rho_i^{(1)}(\boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{x}_i^{(1)}+b_1)^2+c_1\sum\limits_{l=1}^{N_2}\xi_l^{(2)}\\ \text{s.t.:} \quad -f_l^{(1)}(\boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{x}_l^{(2)}+b_1)+\xi_l^{(2)}\geq f_l^{(1)}, \quad \xi_l^{(2)}\geq 0\end{cases} \tag{10}$$

and

WLTSVM-2:

$$\begin{cases}\min \quad \dfrac{1}{2}\sum\limits_{i=1}^{N_2}\rho_i^{(2)}(\boldsymbol{w}_2^{\mathrm{T}}\boldsymbol{x}_i^{(2)}+b_2)^2+c_2\sum\limits_{l=1}^{N_1}\xi_l^{(1)}\\ \text{s.t.:} \quad f_l^{(2)}(\boldsymbol{w}_2^{\mathrm{T}}\boldsymbol{x}_l^{(1)}+b_2)+\xi_l^{(1)}\geq f_l^{(2)}, \quad \xi_l^{(1)}\geq 0\end{cases} \tag{11}$$

where $\rho_i^{(c)}=\sum\limits_{j=1}^{N_c}\boldsymbol{W}_{s,ij}$ (c=1, 2) is the weight of sample $\boldsymbol{x}_i^{(c)}$, and $f_l^{(c)}=\begin{cases}1, & \exists i, W_{d,il}\neq 0\\ 0, & \text{therwise}\end{cases}$ is the weight of sample $\boldsymbol{x}_l^{(c)}$. In fact, $\rho_i^{(c)}$ is introduced to exploit the similarity information between pairs of samples in the same class, and $f_l^{(c)}$ is introduced to choose the margin points in the countrary class. It is evident that WLTSVM only takes into account all the weighted samples of the same class whose weights are at least one and the margin points of the ountrary class whose weights are equal to one [13].

Simplifying the primal problems (Eqs. (10) and (11)) gives two explicit expressions in matrix form as follows:

WLTSVM-1:

$$\begin{cases}\min \quad \dfrac{1}{2}(\boldsymbol{X}_1\boldsymbol{w}_1+\boldsymbol{e}_1b_1)^{\mathrm{T}}\boldsymbol{D}^{(1)}(\boldsymbol{X}_1\boldsymbol{w}_1+\boldsymbol{e}_1b_1)+c_1\boldsymbol{e}_2^{\mathrm{T}}\boldsymbol{\xi}_2\\ \text{s.t.} \quad -\boldsymbol{F}^{(1)}(\boldsymbol{X}_2\boldsymbol{w}_1+\boldsymbol{e}_2b_1)+\boldsymbol{\xi}_2\geq \boldsymbol{F}^{(1)}\boldsymbol{e}_2, \quad \boldsymbol{\xi}_2\geq 0\end{cases} \tag{12}$$

and WLTSVM-2:

$$\begin{cases}\min \quad \dfrac{1}{2}(\boldsymbol{X}_2\boldsymbol{w}_2+\boldsymbol{e}_2b_2)^{\mathrm{T}}\boldsymbol{D}^{(2)}(\boldsymbol{X}_2\boldsymbol{w}_2+\boldsymbol{e}_2b_2)+c_2\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{\xi}_1\\ \text{s.t.} \quad \boldsymbol{F}^{(2)}(\boldsymbol{X}_1\boldsymbol{w}_2+\boldsymbol{e}_1b_2)+\boldsymbol{\xi}_1\geq \boldsymbol{F}^{(2)}\boldsymbol{e}_1, \quad \boldsymbol{\xi}_1\geq 0\end{cases} \tag{13}$$

where $\boldsymbol{D}^{(c)}=\mathrm{diag}(\rho_1^{(c)},\cdots,\rho_{N_c}^{c})$, $\boldsymbol{F}^{(c)}=\mathrm{diag}(f_1^{(c)}, \cdots, f_{N_c}^{(c)})$.

Similar to Refs. [3, 11, 16], two nonparallel hyperplanes in Eq. (1) can be obtained by introducing Lagrangian functions for Eqs. (12) and (13), respectively, and solving the corresponding dual QPPs. Compared to TWSVM, WLTSVM has better classification ability because of taking into account the similarity information between any pair of data points in the same class, and achieving lower computation complexity for only considering margin points in the countrary class during solving the dual QPPs [13]. However, WLTSVM cannot handle large-scale problems since its solution still follows from solving two QPPs. In addition, simple-minded definition of the weight matrix of within-class adjacency graph may lead to poor generalization ability because the underlying similarity information between any pair of data points in the same class cannot be fully reflected.

3 Least squares WLTSVM (LSWLTSVM)

In this section, we enhance WLTSVMs to least squares WLTSVM (LSWLTSVM). Firstly, we introduce LSWLTSVM in the linear case. Then, we extend LSWLTSVM to the nonlinear case.

3.1 Linear LSWLTSVM

Given any pair of points ($\boldsymbol{x}_i$, $\boldsymbol{x}_j$) in the same class, the weight matrix for a within-class adjacency graph G_s is defined as

$$\boldsymbol{W}_{s,ij}^{\text{new}}=\begin{cases}\exp(-\left\|\boldsymbol{x}_i-\boldsymbol{x}_j\right\|^2/t), \text{ if } \boldsymbol{x}_j \text{is } k \text{ nearest}\\ \qquad\qquad\qquad\quad \text{neighbors of } \boldsymbol{x}_i \text{ or } \boldsymbol{x}_i \text{ is } k\\ \qquad\qquad\qquad\quad \text{nearest neighbors of } \boldsymbol{x}_j\\ 0, \text{otherwise}\end{cases} \tag{14}$$

where t is the hot kernel parameter. Compared with to simple-minded definition of the matrix (Eq. (8)), the

matrix Eq. (14) can better reflect the similarity information between any pair of data points in the same class [14−15].

Different from WLTSVM, our decision hyperplanes are obtained from the primal problems directly. The primal problems are modified versions of the primal problems Eqs. (12) and (13) of WLTSVM in least squares sense and constructed following the idea of PSVM proposed in Ref. [2]. Different from the primal problems Eqs. (12) and (13) with the inequality constrains, our primal problems have only equality constrains as follows.

LSWLTSVM-1:

$$\begin{cases}\min & \dfrac{1}{2}(\boldsymbol{X}_1\boldsymbol{w}_1+\boldsymbol{e}_1b_1)^{\mathrm{T}}\boldsymbol{D}_{\mathrm{new}}^{(1)}(\boldsymbol{X}_1\boldsymbol{w}_1+\boldsymbol{e}_1b_1)+\dfrac{c_1}{2}\boldsymbol{\xi}_2^{\mathrm{T}}\boldsymbol{\xi}_2\\ \text{s.t.} & -(\boldsymbol{X}_2\boldsymbol{w}_1+\boldsymbol{e}_2b_1)+\boldsymbol{\xi}_2=\boldsymbol{D}_{\mathrm{new}}^{(2)}\boldsymbol{e}_2\end{cases} \tag{15}$$

and LSWLTSVM-2:

$$\begin{cases}\min & \dfrac{1}{2}(\boldsymbol{X}_2\boldsymbol{w}_2+\boldsymbol{e}_2b_2)^{\mathrm{T}}\boldsymbol{D}_{\mathrm{new}}^{(2)}(\boldsymbol{X}_2\boldsymbol{w}_2+\boldsymbol{e}_2b_2)+\dfrac{c_2}{2}\boldsymbol{\xi}_1^{\mathrm{T}}\boldsymbol{\xi}_1\\ \text{s.t.} & (\boldsymbol{X}_1\boldsymbol{w}_2+\boldsymbol{e}_1b_2)+\boldsymbol{\xi}_1=\boldsymbol{D}_{\mathrm{new}}^{(1)}\boldsymbol{e}_1\end{cases} \tag{16}$$

where $\boldsymbol{D}_{\mathrm{new}}^{(c)}=\mathrm{diag}(\rho_1^{(c)},\cdots,\rho_{N_c}^{c})$, $\rho_i^{(c)}=\sum_{j=1}^{N_c}\boldsymbol{W}_{s,ij}^{\mathrm{new}}$ (c=1, 2).

Note that, there are extra three modifications. The first one is that in the objective functions of Eqs. (15) and (16), the diagonal matrics $\boldsymbol{D}_{\mathrm{new}}^{(1)}$ and $\boldsymbol{D}_{\mathrm{new}}^{(2)}$ are introduced to replace $\boldsymbol{D}^{(1)}$ and $\boldsymbol{D}^{(2)}$ in the objective functions of Eqs. (12) and (13), respectively. This strategy will lead to the underlying similarity information between any pair of data points in the same class can be fully reflected. The second one is that the loss function in Eqs. (15) and (16) is the aquare of 2-norm of slack variables ξ_2 and ξ_1 instead of 1-norm of ξ_2 and ξ_1 used in Eqs. (12) and (13), which makes the constraints $\xi_2\geq 0$ and $\xi_1\geq 0$ redundant. The very simple modification allows us to solve the primal problems Eqs. (15) and (16) by solving a simultaneous system of linear equations. The third one is that in the objective functions of Eqs. (15) and (16), the constraint condition aims at all data points in the contrary class instead of margin points used in Eqs. (12) and (13) and requires the hyperplane to be at a distance of $\rho_i^{(c)}$ from the points of the contrary class, which makes our method less sensitive to noise. In fact, on substituting the equality constraints into the objective function, the primal problem Eq. (15) becomes

$$\min \quad \frac{1}{2}(\boldsymbol{X}_1\boldsymbol{w}_1+\boldsymbol{e}_1b_1)^{\mathrm{T}}\boldsymbol{D}_{new}^{(1)}(\boldsymbol{X}_1\boldsymbol{w}_1+\boldsymbol{e}_1b_1)+\frac{c_1}{2}\left\|\boldsymbol{D}_{\mathrm{new}}^{(2)}\boldsymbol{e}_2+(\boldsymbol{X}_2\boldsymbol{w}_1+\boldsymbol{e}_2b_1)\right\|^2 \tag{17}$$

Setting the gradient of Eq. (17) with respect to $\boldsymbol{w}_1$ and b_1 to zero gives

$$\boldsymbol{X}_1^{\mathrm{T}}\boldsymbol{D}_{\mathrm{new}}^{(1)}(\boldsymbol{X}_1\boldsymbol{w}_1+\boldsymbol{e}_1b_1)+c_1\boldsymbol{X}_2^{\mathrm{T}}(\boldsymbol{D}_{\mathrm{new}}^{(2)}\boldsymbol{e}_2+(\boldsymbol{X}_2\boldsymbol{w}_1+\boldsymbol{e}_2b_1))=0 \tag{18}$$

and

$$\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{D}_{\mathrm{new}}^{(1)}(\boldsymbol{X}_1\boldsymbol{w}_1+\boldsymbol{e}_1b_1)+c_1\boldsymbol{e}_2^{\mathrm{T}}(\boldsymbol{D}_{\mathrm{new}}^{(2)}\boldsymbol{e}_2+(\boldsymbol{X}_2\boldsymbol{w}_1+\boldsymbol{e}_2b_1))=0 \tag{19}$$

Arranging Eqs. (18) and (19) in matrix form and solving for $\boldsymbol{w}_1$ and b_1 gives

$$\frac{1}{c_1}\begin{bmatrix}\boldsymbol{X}_1^{\mathrm{T}}\\ \boldsymbol{e}_1^{\mathrm{T}}\end{bmatrix}\boldsymbol{D}_{\mathrm{new}}^{(1)}\begin{bmatrix}\boldsymbol{X}_1 & \boldsymbol{e}_1\end{bmatrix}\begin{bmatrix}\boldsymbol{w}_1\\ b_1\end{bmatrix}+\begin{bmatrix}\boldsymbol{X}_2^{\mathrm{T}}\\ \boldsymbol{e}_2^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}\boldsymbol{X}_2 & \boldsymbol{e}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{w}_1\\ b_1\end{bmatrix}+\begin{bmatrix}\boldsymbol{X}_2^{\mathrm{T}}\\ \boldsymbol{e}_2^{\mathrm{T}}\end{bmatrix}\boldsymbol{D}_{\mathrm{new}}^{(2)}\boldsymbol{e}_2=0. \tag{20}$$

Let $\boldsymbol{E}=[\boldsymbol{X}_1\ \boldsymbol{e}_1]$, $\boldsymbol{F}=[\boldsymbol{X}_2\ \boldsymbol{e}_2]$. The solution becomes

$$\begin{bmatrix}\boldsymbol{w}_1\\ b_1\end{bmatrix}=-\left[\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F}+\frac{1}{c_1}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{D}_{\mathrm{new}}^{(1)}\boldsymbol{E}\right]^{-1}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{D}_{\mathrm{new}}^{(2)}\boldsymbol{e}_2 \tag{21}$$

In an exactly similar way the solution of the problem (Eq. (16)) can be shown as

$$\begin{bmatrix}\boldsymbol{w}_2\\ b_2\end{bmatrix}=\left[\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E}+\frac{1}{c_2}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{D}_{\mathrm{new}}^{(2)}\boldsymbol{F}\right]^{-1}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{D}_{\mathrm{new}}^{(1)}\boldsymbol{e}_1 \tag{22}$$

Similar to TWSVM, we can introduce a regularization term $\varepsilon\boldsymbol{I}(\varepsilon>0)$ to deal with the case when ($\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F}+\boldsymbol{E}^{\mathrm{T}}\boldsymbol{D}_{\mathrm{new}}^{(1)}\boldsymbol{E}/c_1$) and ($\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E}+\boldsymbol{F}^{\mathrm{T}}\boldsymbol{D}_{\mathrm{new}}^{(2)}\boldsymbol{F}/c_2$) are singular and avoid the possible ill-conditioning.

Once the augmented vectors of Eqs. (21) and (22) are known, the two separating planes of Eq. (1) are obtained. The class of an unknown data point $\boldsymbol{x}\in\mathbf{R}^n$ is determined as

$$\mathrm{class}(\boldsymbol{x})=\underset{i=1,2}{\mathrm{argmin}}\left|\boldsymbol{x}^{\mathrm{T}}\boldsymbol{w}_i+b_i\right| \tag{23}$$

where $|\cdot|$ denotes the perpendicular distance of the point $\boldsymbol{x}$ from the plane.

3.2 Nonlinear LSWLTSVM

In the real world, classification problems cannot always be handled by linear kernel methods. Thus, we extend our LSWLTSVM to a nonlinear version by considering the following kernel-based hyperplanes:

$$K(\boldsymbol{x}^{\mathrm{T}},\boldsymbol{X}^{\mathrm{T}})\boldsymbol{u}_1+b_1=0,\ K(\boldsymbol{x}^{\mathrm{T}},\boldsymbol{X}^{\mathrm{T}})\boldsymbol{u}_2+b_2=0 \tag{24}$$

where K is an appropriately chosen kernel.

The optimization problems for an nonlinear LSWLTSVM (NLSWLTSVM) instead of the primal ones in the input space as Eqs. (15) and (16) can be reformulated as

NLSWLTSVM-1:

$$\begin{cases} \min \quad \frac{1}{2}(K(\boldsymbol{X}_1,\boldsymbol{X}^{\mathrm{T}})\boldsymbol{u}_1+\boldsymbol{e}_1b_1)^{\mathrm{T}}\boldsymbol{D}_{\mathrm{new}}^{(1)}(K(\boldsymbol{X}_1,\boldsymbol{X}^{\mathrm{T}})\cdot \\ \qquad \boldsymbol{u}_1+\boldsymbol{e}_1b_1)+\frac{c_1}{2}\boldsymbol{\xi}_2^{\mathrm{T}}\boldsymbol{\xi}_2 \\ \text{s.t.} \quad -(K(\boldsymbol{X}_2,\boldsymbol{X}^{\mathrm{T}})\boldsymbol{u}_1+\boldsymbol{e}_2b_1)+\boldsymbol{\xi}_2=\boldsymbol{D}_{\mathrm{new}}^{(2)}\boldsymbol{e}_2 \end{cases} \tag{25}$$

and NLSWLTSVM-2:

$$\begin{cases} \min \quad \frac{1}{2}(K(\boldsymbol{X}_2,\boldsymbol{X}^{\mathrm{T}})\boldsymbol{u}_2+\boldsymbol{e}_2b_2)^{\mathrm{T}}\boldsymbol{D}_{\mathrm{new}}^{(2)}(K(\boldsymbol{X}_2,\boldsymbol{X}^{\mathrm{T}})\cdot \\ \qquad \boldsymbol{u}_2+\boldsymbol{e}_2b_2)+\frac{c_2}{2}\boldsymbol{\xi}_1^{T}\boldsymbol{\xi}_1 \\ \text{s.t.} \quad (K(\boldsymbol{X}_1,\boldsymbol{X}^{\mathrm{T}})\boldsymbol{u}_2+\boldsymbol{e}_1b_2)+\boldsymbol{\xi}_1=\boldsymbol{D}_{\mathrm{new}}^{(1)}\boldsymbol{e}_1 \end{cases} \tag{26}$$

The solution of optimization problems (Eqs. (25) and (26)) can be derived as

$$\begin{bmatrix}\boldsymbol{u}_1\\ b_1\end{bmatrix}=-\left[\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}+\frac{1}{c_1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{D}_{\mathrm{new}}^{(1)}\boldsymbol{G}\right]^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{D}_{\mathrm{new}}^{(2)}\boldsymbol{e}_2 \tag{27}$$

and

$$\begin{bmatrix}\boldsymbol{u}_2\\ b_2\end{bmatrix}=\left[\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}+\frac{1}{c_2}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{D}_{\mathrm{new}}^{(2)}\boldsymbol{H}\right]^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{D}_{\mathrm{new}}^{(1)}\boldsymbol{e}_1 \tag{28}$$

where $\boldsymbol{G}=[K(\boldsymbol{X}_1,\boldsymbol{X}^{\mathrm{T}})\ \boldsymbol{e}_1]$ and $\boldsymbol{H}=[\ K(\boldsymbol{X}_2,\boldsymbol{X}^{\mathrm{T}})\ \boldsymbol{e}_2]$.

Once the augmented vectors of Eqs. (27) and (28) are known, the two separating hyperplanes of Eq. (24) are obtained. The class of an unknown data point $\boldsymbol{x}\in\mathbf{R}^n$ is determined as

$$\mathrm{class}(\boldsymbol{x})=\underset{i=1,2}{\mathrm{argmin}}\left|K(\boldsymbol{x}^{\mathrm{T}},\boldsymbol{X}^{\mathrm{T}})\boldsymbol{u}_i+b_i\right| \tag{29}$$

4 Connection to WLTSVM

4.1 Generalization performance

LSWLTSVM uses a hot kernel function, instead of the simple-minded definition in WLTSVM, to define the weight matrix of within-class adjacency graph. This strategy will make LSWLTSVM achieve better generalization performance because the underlying similarity information between any pair of data points in the same class can be better reflected. Figure 1 describes the decision hyperplanes of LSWLTSVM and WLTSVM on an artificial dataset. As can be seen from the case illustrated in Fig. 1, the LSWLTSVM decision hyperplanes reflect the intrinsic manifold structure of the data and show that they are more reasonable. Although one of the WLTSVM decision hyperplanes reflects the intrinsic manifold structure of the data, the other reflects the average information of the corresponding class distribution.

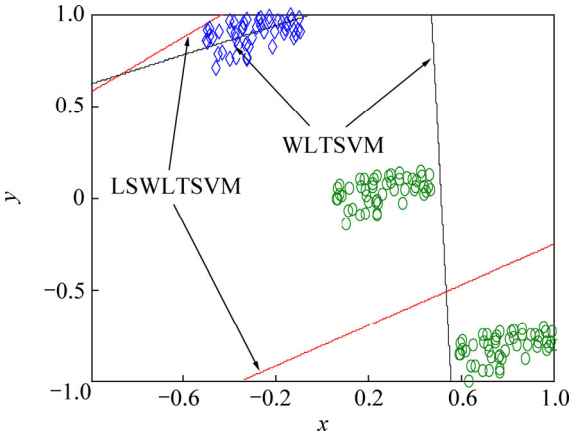

Fig. 1 Illustration of decision hyperplanes generated by WLTSVM and LSWLTSVM

Additionally, WLTSVM only uses part margin points in the contrary class to construct the constraints. This strategy can improve the learning speed of WLTSVM. However, it also possibly reduce the generalization ability since it may use the outliers in the contrary classes to construct hyperplanes. To give further explanation, we rewrite Eq. (10) as

WLTSVM-1:

$$\begin{cases} \min \quad \frac{1}{2}\sum_{i=1}^{N_1}\rho_i^{(1)}(\boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{x}_i^{(1)}+b_1)^2+c_1\sum_{l=1}^{N_2}h(\boldsymbol{x}_l)\eta_l^{(2)} \\ \text{s.t.} \quad \boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{x}_l^{(2)}+b_1\le -1+\eta_l^{(2)}, \quad \eta_l^{(2)}\ge 0 \end{cases} \tag{30}$$

where $\eta_l^{(2)}=\xi_l^{(2)}/h(\boldsymbol{x}_l)$. Obviously, for Eq. (30), the more $\boldsymbol{x}_l$ is near to class 1, the larger its penalty factor is. This means that WLTSVM is possibly sensitive to outliers. However, our LSWLTSVM considers the weight for each point in the contrary class. This strategy obviously reduces the influence of outliers.

4.2 Computational complexity

The solution of our LSWLTSVM follows directly from solving two systems of linear equations as opposed to solving two QPPs in WLTSVM. This strategy obviously makes our LSWLTSVM faster than WLTSVM. Moreover, only within-class adjacency graph is required to be constructed in our LSWLTSVM; while both within-class adjacency graph and between-class adjacency graph are needed in WLTSVM.

5 Experiments

In order to evaluate the proposed LSWLTSVM, we investigate its classification accuracies and computational efficiencies on real-world UCI benchmark datasets [17] and David Musicant's NDC Data Generator datasets [18]. In experiments, we focus on the comparison between the proposed algorithm and some

state-of-the-art classification methods, including TWSVM, PTSVM and WLTSVM. All the classification algorithms are implemented in MATLAB 7.1 and carried out experiments on a PC with an Intel® Core 2 Duo processor (2.3 GHz), 2 GB of RAM.

5.1 UCI datasets

UCI datasets are commonly used in testing machine learning algorithms [19−22]. Table 1 shows the linear kernel comparison of LSWLTSVM versus TWSVM, PTSVM and WLTSVM. We compared them through ten-fold cross-validation and computed the means and standard errors for the results. For the values of parameters in these algorithms, we set $c_1=c_2$ for these methods and selected them from the set of values $\{2^i|i=-8,-6,\cdots,8\}$. For WLTSVM, the neighborhood size k_1 was searched in the range from one to nine and k_2 was set to five. For our LSWLTSVM, the neighborhood k was searched in the same range as k_1 in WLTSVM and the hot kernel parameter t was searched from the set of values $\{2^i|i=-1,0,\cdots,6\}$. The effectiveness of WLTSVM over TWSVM and PTSVM has already been reported in Ref. [13], but Table 1 reveals that the accuracy of LSWLTSVM is similar to or better than that of WLTSVM for the standard datasets.

Table 2 compares the classification accuracy for the nonlinear TWSVM, PTSVM, WLTSVM and LSWLTSVM. For each algorithm, a Gaussian kernel (i.e. $K(x_i,x_j)=\exp(-\|x_i-x_j\|^2/\sigma)$) was selected and the parameter σ was searched in the range from 2^{-1} to 2^8. Note that we used rectangular kernel using 10% of total data points. The effectiveness of nonlinear WLTSVM over TWSVM and PTSVM has already been reported in Ref. [13], but Table 2 shows that the accuracy of nonlinear LSWLTSVM is comparable to or better than that of the other two algorithms. In fact, the accuracy of nonlinear LSWLTSVM is slightly better than that of the other classifiers on many data sets.

Table 1 Classification comparison for TWSVM, PTSVM, WLTSVM and LSWLTSVM with linear kernel

Data set ($N\times n$)	(Test±Std)/%			
	TWSVM	PTSVM	WLTSVM	LSWLTSVM
wpbc (194×33)	80.72±8.57	79.67±7.34	81.52±9.14	**82.30±7.91**
cleve (297×13)	82.42±4.21	82.08±3.38	83.11±4.74	**83.46±5.52**
bupa-liver (345×6)	68.69±3.23	67.59±5.30	**69.86±4.52**	69.65±3.33
monks2 (432×6)	67.14±0.89	67.84±3.15	67.14±0.89	**67.38±1.19**
vertebral (310×6)	85.48±6.49	84.19±5.85	**85.81±6.32**	85.16±5.98
breast_gy (277×9)	73.05±4.86	70.83±6.45	74.46±5.63	**75.94±5.81**
hepatitis (155×19)	84.17±7.79	83.67±10.8	85.33±7.77	**87.67±6.33**
monks3 (432×6)	77.41±11.32	80.74±13.05	80.74±13.05	**83.62±18.39**
heart (270×14)	84.07±5.25	84.07±5.51	84.44±3.98	**85.19±4.38**
cmc (1473×9)	67.77±3.63	66.08±4.92	68.38±3.79	**68.81±4.00**
tic-tac-toe (958×9)	64.93±2.31	63.11±5.33	65.23±4.09	**73.83±6.99**
pima (768×8)	77.17±2.80	76.41±3.53	77.33±3.09	**77.70±3.12**

Table 2 Classification comparison for TWSVM, PTSVM, WLTSVM and LSWLTSVM with nonlinear kernel

Data set ($N\times n$)	(Test±Std)/%			
	TWSVM	PTSVM	WLTSVM	LSWLTSVM
wpbc (194×33)	79.36±5.52	78.03±6.59	80.35±7.39	**80.88±1.08**
cleve (297×13)	83.80±4.37	85.93±4.90	84.90±5.54	**86.16±4.88**
bupa-liver (345×6)	74.13±4.53	72.18±5.90	73.58±5.12	**74.79±2.98**
haberman (306×3)	76.78±4.96	74.17±3.89	76.83±2.83	**77.83±3.80**
monks2 (432×6)	66.44±3.01	68.71±3.43	66.21±2.42	**68.78±17.92**
vertebral (310×6)	85.81±6.32	84.52±6.42	86.13±5.96	**88.39±6.15**
breast_gy (277×9)	75.86±5.89	73.57±4.10	76.23±6.11	**77.34±7.58**
monks3 (432×6)	98.70±3.89	98.52±4.44	**99.26±2.22**	98.70±3.89
sonar (208×60)	65.71±18.79	64.50±13.5	69.00±18.14	**70.79±17.48**
spect (267×22)	85.49±8.34	84.42±7.43	85.19±7.60	**85.87±7.04**

5.2 NDC datasets

We also conducted experiments on lage datsets, generated using David Musicants NDC Data Generator to demonstrate the computational efficiency of all these algorithms. For all experiments, the parameters for all algorithms are fixed to be one. Figure 2 illustrates the training time comparison for four algorithms with linear kernel. WLTSVM is faster than TWSVM or PTSVM, which has already been reported in Ref. [13]. However, compared with WLTSVM, LSWLTSVM consumes fewer training time in the training stage. There are two main reasons. One is that the solution of our LSWLTSVM follows directly from solving two systems of linear equations as opposed to solving two QPPs in WLTSVM. The other is that only the within-class adjacency graph is required to be constructed in our LSWLTSVM; while both within-class adjacency graph and between-class adjacency graph are needed to be done in WLTSVM. Figure 3 illustrates the training time comparison for four algorithms with nonlinear kernel. Note that we used rectangular kernel using 10% of total data points. From Figure 3, similar results can be obtained.

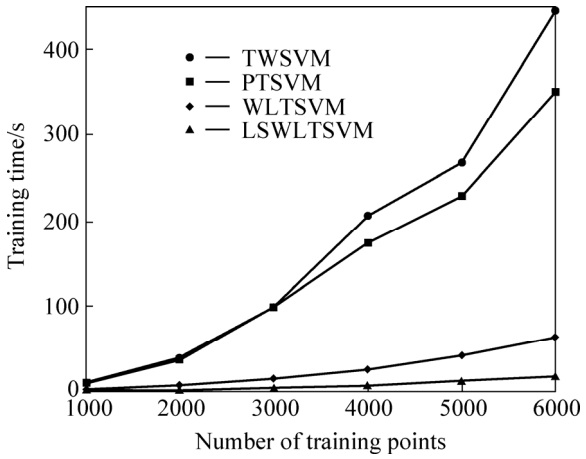

Fig. 2 Illustration of training time comparision for four algorithms with linear kernel

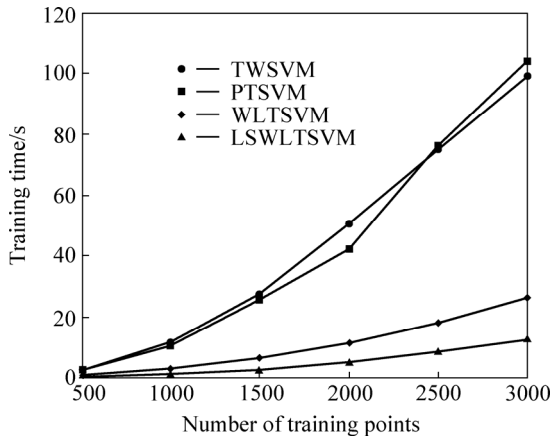

Fig. 3 Illustration of training time comparision for four algorithms with nonlinear kernel

6 Conclusions and future work

In this work, we have enhanced WLTSVM to least squares WLTSVM (LSWLTSVM). LSWLTSVM is formulated as an extremely simple algorithm for generating linear or nonlinear binary classifiers using two non-parallel hyperplanes. In LSWLTSVM, we solve the two primal problems of WLTSVM instead of two dual problems usually solved in WLTSVM and TWSVM. LSWLTSVM requires just the solution of two systems of linear equations for both linear and nonlinear cases in contrast to WLTSVM, which requires solving two quadratic programming problems in addition to two systems of linear equations. Moreover, two extra modifications are proposed in LSWLTSVM to improve the generalization capability. One is that LSWLTSVM uses a hot kernel function instead of simple-minded definition to define the weight matrix of within-class adjacency graph. The other is that LSWLTSVM considers the weight for each point in the contrary class instead of margin points in WLTSVM. Experimental results on several benchmarch datasets reveal that LSWLTSVM is better than WLTSVM in terms of both classification effectiveness and lower computational cost. However, a limitation is that the sparseness in the solution of our LSWLTSVM is lost. Thus, the further work will include how to improve the sparseness of the solution.

References

[1] MANGASARIAN O L, WILD E. Multisur face proximal support vector machine classification via generalized eigenvalues [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28(1): 69−74.

[2] FUNG G, MANGASARIAN O L. Proximal support vector machine classifiers [C]// Proceedings KDD-2001: Knowledge Discovery and Data Mining. New York: ACM Press, 2001: 77−86.

[3] JAYADEVA, KHEMCHANDAI R, CHANDRA S. Twin support vector machines for pattern classification [J]. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2007, 29(5): 905−910.

[4] KUMAR M A, GOPAL M. Least squares twin support vector machines for pattern classification [J]. Expert Systems with Applications, 2009, 36(4): 7535−7543.

[5] YE Qiao-lin, ZHAO Chun-xia, YE Ning, CHEN Xiao-bo. Localized twin SVM via convex minimization [J]. Neurocomputing, 2011, 74(4): 580−587.

[6] PENG Xin-jun. TPMSVM: A novel twin parametric-margin support vector machine for pattern recognition [J]. Pattern Recognition, 2011, 44(10/11): 2678−2692.

[7] QI Zhi-quan, TIAN Ying-jie, SHI Yong. Structural twin support vector machine for classification [J]. Knowledge-Based Systems, 2013, 43(5): 74−81.

[8] PENG Xin-jun, XU Dong. Twin mahalanobis distance-based support vector machines for pattern recognition [J]. Information Science, 2012, 200(10): 22−37.

[9] SHAO Yuan-hai, ZHANG Chun-hua, WANG Xiao-bo, DENG

Nai-yang. Improvements on twin support vector machines [J]. IEEE Transactions on Neural Networks, 2011, 22(6): 962−968.

[10] YE Qiao-lin, ZHAO Chun-xia, YE Ning, CHEN Yan-nan. Multi-weight vector projection support vector machines [J]. Pattern Recognition Letters, 2010, 31(13): 2006−2011.

[11] CHEN Xiao-bo, YANG Jian, YE Qiao-lin, LIANG Jun. Recursive projection twin support vector machine via within-class variance minimization [J]. Pattern Recognition, 2011, 44(10): 2643−2655.

[12] SHAO Yun-hai, WANG Zhen, CHEN Wei-jie, DENG Nai-yang. A regularization for the projection twin support vector machine [J]. Knowledge-Based Systems, 2013, 37: 203−210.

[13] YE Qiao-lin, ZHAO Chun-xia, GAO Shang-bing, ZHENG Hao. Weighted twin support vector machines with local information and its application [J]. Neural Networks, 2012, 35(11): 31−39.

[14] HE Xiao-fei, NIYOGI P. Locality preserving projection [OL]. [2012−07−08]. http://www.docin.com/p-202458452.html.

[15] WANG Xiao-ming, CHUNG Fu-lai, WANG Shi-tong. On minimum class locality preserving variance support vector machine [J]. Patter Recognition, 2010, 43(8): 2753−2762

[16] HUA Xiao-peng, DING Shi-fei. Locality preserving twin support vector machines [J]. Journal of Computer Research and Development, 2014, 51(3): 590−597. (in Chinese)

[17] BLAKE C, MERZ C. UCI repository of machine learning databases [OL]. [1998−05−15]. http://www.ics.uci.edu/~mlearn/MLRepository.html.

[18] MUSICANT D R. NDC: Normally distributed clustered datasets [OL]. [1998−08−15]. http://research.cs.wisc.edu/dmi/svm/ndc/.

[19] SHAO Yuan-hai, DENG Nai-yang, YANG Zhi-min, CHEN Wei-jie, WANG Zhen. Probabilistic outputs for twin support vector machines [J]. Knowledge-Based Systems, 2012, 33(9): 145−151.

[20] XUE Hui, CHEN Song-can. Glocalization pursuit support vector machine [J]. Neural Computing and Applications, 2011, 20(7): 1043−1053.

[21] DING Li-zhong, LIAO Shi-zhong. KMA-a: A kernel matrix approximation algorithm for support vector machines [J]. Journal of Computer Research and Development, 2012, 49(4): 746−753. (in Chinese)

[22] PENG Xin-jun, XU Dong. Norm-mixed twin support vector machine classifier and its geometric algorithm [J]. Neurocomputing, 2013, 99(1): 486−495.

(Edited by YANG Hua)