IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 31, 2019, accepted April 24, 2019, date of publication April 30, 2019, date of current version May 14, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2914022

Weighted Linear Loss Projection Twin Support Vector Machine for Pattern Classification

SUGEN CHEN[1,2], **JUNFENG CAO**[3], **AND ZHONG HUANG**[4]

[1]School of Mathematics and Computational Science, Anqing Normal University, Anqing 246133, China
[2]Key Laboratory of Modeling, Simulation and Control of Complex Ecosystem in Dabie Mountains of Anhui Higher Education Institutes, Anqing Normal University, Anqing 246133, China
[3]School of Science, Jiangnan University, Wuxi 214122, China
[4]School of Physics and Electrical Engineering, Anqing Normal University, Anqing 246133, China

Corresponding author: Sugen Chen (chensugen@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702012, in part by the University Outstanding Young Talent Support Project of Anhui Province of China under Grant gxyq2017026, in part by the Natural Science Foundation of Anhui Province of China under Grant 1908085MF195, in part by the University Natural Science Research Project of Anhui Province of China under Grant KJ2016A431, Grant KJ2017A361 and Grant KJ2017A368, and in part by the Program for Innovative Research Team in Anqing Normal University.

ABSTRACT Based on the recently proposed projection twin support vector machine (PTSVM) and least squares projection twin support vector machine (LSPTSVM), in this paper, we propose a weighted linear loss projection twin support vector machine, namely WLPTSVM for short. By introducing the weighted linear loss function, the proposed WLPTSVM not only solves systems of linear equations with lower computational cost but also obtains comparable classification accuracy. In addition, it is able to dispose of large scale classification problems efficiently without any extra external optimizers. The experiments conducted on synthetic and several benchmark datasets illustrate the effectiveness of our WLPTSVM.

INDEX TERMS Pattern classification, twin support vector machine, projection twin support vector machine, weighted linear loss function.

I. INTRODUCTION

Traditional support vector machine (SVM) is an excellent kernel-based method for pattern classification and regression [1], [2], which has already been successfully applied to a variety of real-world problems such as image classification [3], bioinformatics [4] and text categorization [5]. However, the training stage involves solving a quadratic programming problem (QPP) with high computational complexity $O(m^3)$, where m is the total size of training samples. This drawback restricts the application of SVM in large-scale problems. On the one hand, many efficient algorithms such as Chunking [2], SMO [6], LIBSVM [7], PSVM [8] and LS-SVM [9] have been proposed to improve the training speed. Recently, on the other hand, multiple surface support vector machines such as twin support vector machine (TWSVM) [10] and projection twin support vector machine (PTSVM) [11], as an extension direction of SVM, have been studied extensively. In 2006, Mangasarian and Wild [12] proposed generalized eigenvalue proximal support vector machine (GEPSVM), which aims at seeking two nonparallel proximal hyperplanes such that each hyperplane is closer to one of two classes and as far as possible from the other. In the spirit of GEPSVM, in 2007, Jayadeva *et al.* [10] proposed another nonparallel hyperplane classifier for pattern classification, namely twin support vector machine (TWSVM). It seeks two nonparallel hyperplanes by resolving two smaller and related SVM-type problems. From then on, many variants of TWSVM are proposed, such as least square TWSVM (LSTSVM) [13], twin bounded support vector machine (TBSVM) [14], twin parametric-margin SVM (TPMSVM) [15], robust TWSVM (RTSVM) [16], nonparallel SVM (NPSVM) [17], L2P-norm distance TWSVM [18], angle-based TWSVM [19] and fuzzy TWSVM (FTSVM) [20]. The more recent extensions and developments in TWSVMs have been discussed in [21], [22]. Meanwhile, in order to avoid solving the QPPs in TWSVM, Ye *et al.* proposed the multi-weight vector projection support vector machine (MVSVM) [23] based on GEPSVM, which seeks one weight vector instead of a hyperplane for each class. The weight vectors of MVSVM can be found by solving a pair of eigenvalue problems. Inspired by MVSVM and TWSVM, in 2011, Chen *et al.* proposed the projection twin support vector machine (PTSVM) [11], which aims at seeking two projection directions by solving a pair of SVM-type problems rather than eigenvalue

The associate editor coordinating the review of this manuscript and approving it for publication was Tallha Akram.

problems. From then on, various improved algorithms based on PTSVM are proposed [24]–[34], e.g. RPTSVM [24], LSPTSVM [25], [26], IPTSVM [27], LIWLSPTSVM [28], PNPSVM [29], NPTSVM [30], PTSVR [31] and other variants PTSVM algorithms [32]–[34]. Although LSTSVM has been presented by using the squared loss function instead of hinge loss function in TWSVM and obtains very fast training speed since two QPPs are replaced by two systems of linear equations, but may result in the reduction of classification ability and the characteristic of constructing two nonparallel hyperplanes may be weakened [35]. In order to mitigate this problem, Shao *et al.* [36] proposed a twin-type support vector machine with weighted linear loss function, called weighted linear loss twin support vector machine (WLTSVM), which achieved the comparable classification accuracy but with less computational time.

Based on the above analysis and inspired by WLTSVM, in this paper, we propose a novel projection twin support vector machine with weighted linear loss function for pattern classification, termed as WLPTSVM. However, different from WLTSVM, the linear version of WLPTSVM aims at seeking a projection direction instead of a hyperplane for each class by solving a system of linear equations, which inherits the main idea of PTSVM. Different from PTSVM, in the linear version of our WLPTSVM, a weighted linear loss function is introduced. Specifically, our WLPTSVM has the following advantages: First, different from TWSVM and PTSVM, weighted linear loss function is utilized to replace the hinge loss function leading to solve two systems of linear equations that are much simpler than that of TWSVM and PTSVM, where two QPPs are solved. Second, different from LSPTSVM, weighted linear loss function is used to replace quadratic loss function, improving the classification accuracy of LSPTSVM. Third, different from PTSVM and LSPTSVM, nonlinear version of our WLPTSVM is also presented, which is missing in original PTSVM and LSPTSVM. At last, the systems of linear equations in our WLPTSVM are solved efficiently by utilizing the well-known conjugate gradient (CG) algorithm [37] such that our WLPTSVM can deal with large-scale classification problems without any extra external optimizers. Then, comparing to the existing algorithms, e.g. TWSVM [10], PTSVM [11], RPTSVM [24], LSPTSVM [25] and WLTSVM [36], some experimental results on synthetic and benchmark datasets illustrate the effectiveness of our WLPTSVM.

The rest of this paper is organized as follows. In Section II, a brief review of PTSVM, LSPTSVM and WLTSVM are given. Section III proposes linear and nonlinear version of our WLPTSVM, respectively. And the experimental results on both synthetic datasets and real-world benchmark datasets are reported in Section IV. Last, Section V gives the conclusion.

II. RELATED WORKS

Let us consider a binary classification problem in the n-dimensional real space R^n and a set of training data samples is represented by $T = \{(x_j^{(i)}, y_j)|j = 1, 2, \cdots, m_i; i = 1, 2.\}$, where $x_j^{(i)} \in R^n$ is the j-th input belongs to class W_i and $y_j \in \{+1, -1\}$ are corresponding outputs. In addition, we set $m = m_1 + m_2$ and organize the m_1 samples of positive class W_1 by a $m_1 \times n$ matrix $A \in R^{m_1 \times n}$ and the m_2 samples of negative class W_2 by a $m_2 \times n$ matrix $B \in R^{m_2 \times n}$.

A. PTSVM

The key idea of projection twin support vector machine (PTSVM) [11] is to find a projection axis for each class, such that within-class variance of the projected data points of its own class is minimized meanwhile the projected data points of the other class scatter away as far as possible. Thus, the primal problems of linear PTSVM are expressed as follows.

$$\min_{w_1} \frac{1}{2} w_1^T S_1 w_1 + c_1 e_2^T \xi_2$$
$$s.t.\ Bw_1 - \frac{1}{m_1} e_2 e_1^T A w_1 + \xi_2 \geq e_2, \quad \xi_2 \geq 0, \tag{1}$$
$$\min_{w_2} \frac{1}{2} w_2^T S_2 w_2 + c_2 e_1^T \xi_1$$
$$s.t.\ -(Aw_2 - \frac{1}{m_2} e_1 e_2^T B w_2) + \xi_1 \geq e_1, \quad \xi_1 \geq 0, \tag{2}$$

where $c_1 > 0$ and $c_2 > 0$ are trade-off parameters, $e_1 \in R^{m_1}$ and $e_2 \in R^{m_2}$ are vectors of ones, ξ_1 and ξ_2 are both non-negative slack variables. S_1 and S_2 are within-class variance matrices defined by

$$S_1 = \sum_{i=1}^{m_1} (x_i^{(1)} - \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)})(x_i^{(1)} - \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)})^T \tag{3}$$
$$S_2 = \sum_{i=1}^{m_2} (x_i^{(2)} - \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)})(x_i^{(2)} - \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)})^T \tag{4}$$

It has been shown that when S_1 and S_2 are nonsingular or invertible, the solutions of the primal problems (1) and (2) are obtained by solving the Wolfe dual problems

$$\max_{\alpha} e_2^T \alpha - \frac{1}{2} \alpha^T (B - \frac{1}{m_1} e_2 e_1^T A) S_1^{-1} (B - \frac{1}{m_1} e_2 e_1^T A)^T \alpha$$
$$s.t.\ 0 \leq \alpha \leq c_1 e_2 \tag{5}$$
$$\max_{\gamma} e_1^T \gamma - \frac{1}{2} \gamma^T (A - \frac{1}{m_2} e_1 e_2^T B) S_2^{-1} (A - \frac{1}{m_2} e_1 e_2^T B)^T \gamma$$
$$s.t.\ 0 \leq \gamma \leq c_2 e_1 \tag{6}$$

where $\alpha \in R^{m_2}$ and $\gamma \in R^{m_1}$ are the Lagrangian multipliers. Therefore, the projection axes are obtained from the solution α and γ in (5) and (6) by

$$w_1 = S_1^{-1} (B - \frac{1}{m_1} e_2 e_1^T A)^T \alpha \tag{7}$$
$$w_2 = -S_2^{-1} (A - \frac{1}{m_2} e_1 e_2^T B)^T \gamma \tag{8}$$

After the optimal projection axes are obtained according to (7) and (8), the training stage of PTSVM is completed. For testing, the label of a new coming data point $x \in R^n$ is assigned to class W_i, depending on the distance between the projection of x and projected class mean which is expressed as

$$x \in W_i, i = \arg\min_{k=1,2} \left| w_k^T (x - \frac{1}{m_k} \sum_{j=1}^{m_k} x_j^{(k)}) \right| \tag{9}$$

The above procedure seeks a single direction for each class to make the corresponding projected data points well separated. Meanwhile, it has been extended to find multiple orthogonal directions to further enhance its performance. The detailed content can be seen in [11].

B. LSPTSVM

Different from PTSVM, LSPTSVM [25] has been presented by using the squared loss function instead of the hinge loss function in PTSVM and by using equality constraints instead of inequality constrains. Thus, the primal problems of linear LSPTSVM are expressed as

$$\begin{aligned} &\min_{w_1} \tfrac{1}{2}w_1^T S_1 w_1 + \tfrac{c_1}{2}\xi_2^T\xi_2 + \tfrac{c_3}{2}w_1^2 \\ &s.t.\ Bw_1 - \tfrac{1}{m_1}e_2e_1^T A w_1 + \xi_2 = e_2, \end{aligned} \tag{10}$$

$$\begin{aligned} &\min_{w_2} \tfrac{1}{2}w_2^T S_2 w_2 + \tfrac{c_2}{2}\xi_1^T\xi_1 + \tfrac{c_4}{2}w_2^2 \\ &s.t.\ -(Aw_2 - \tfrac{1}{m_2}e_1e_2^T B w_2) + \xi_1 = e_1, \end{aligned} \tag{11}$$

where $c_1 > 0, c_2 > 0, c_3 > 0$ and $c_4 > 0$ are positive trade-off parameters, $e_1 \in R^{m_1}$ and $e_2 \in R^{m_2}$ are vectors of ones, ξ_1 and ξ_2 are both nonnegative slack variables. S_1 and S_2 are within-class variance matrices defined by (3) and (4).

On substituting the equality constraints into the objective functions, the formulas become

$$\min_{w_1} \tfrac{1}{2}w_1^T S_1 w_1 + \tfrac{c_1}{2}||e_2 - Bw_1 + \tfrac{1}{m_1}e_2e_1^T Aw_1||^2 + \tfrac{c_3}{2}w_1^2 \tag{12}$$

$$\min_{w_2} \tfrac{1}{2}w_2^T S_2 w_2 + \tfrac{c_2}{2}||e_1 + Aw_2 - \tfrac{1}{m_2}e_1e_2^T Bw_2||^2 + \tfrac{c_4}{2}w_2^2 \tag{13}$$

Setting the gradient of (12) with respect to w_1 to zero and setting the gradient of (13) with respect to w_2 to zero, the following solutions are obtained

$$\begin{aligned} w_1 = &\left(\frac{S_1}{c_1} + (-B+\frac{1}{m_1}e_2e_1^T A)^T(-B+\frac{1}{m_1}e_2e_1^T A)+\frac{c_3}{c_1}I\right)^{-1} \\ &\times(B - \frac{1}{m_1}e_2e_1^T A)^T e_2 \end{aligned} \tag{14}$$

$$\begin{aligned} w_2 = -&\left(\frac{S_2}{c_2} + (A-\frac{1}{m_2}e_1e_2^T B)^T(A-\frac{1}{m_2}e_1e_2^T B)+\frac{c_4}{c_2}I\right)^{-1} \\ &\times(A - \frac{1}{m_2}e_1e_2^T B)^T e_1 \end{aligned} \tag{15}$$

After the optimal projection axes are obtained according to (14) and (15), the training stage of linear LSPTSVM is completed and then the testing stage is similar to PTSVM. In addition, the nonlinear version of LSPTSVM was proposed by Ding and Hua [26]. More detail about LSPTSVM can be seen in [25], [26].

C. WLTSVM

For a binary classification problem, linear loss TSVM also seeks two nonparallel hyperplanes. However, different from TSVM, linear loss TSVM adopts linear loss to estimate the misclassification loss and adds a modified regularization item to minimize the structural risk. The primal problems of linear loss TSVM are expressed as follows.

$$\begin{aligned} &\min_{w_1,b_1} \frac{c_3}{2}(||w_1||^2 + b_1^2) + \frac{1}{2}(\xi_1^T\xi_1 + c_1e_2^T\xi_2) \\ &s.t.\ Aw_1 + e_1b_1 = \xi_1, \\ &\quad Bw_1 + e_2b_1 + e_2 = \xi_2, \end{aligned} \tag{16}$$

$$\begin{aligned} &\min_{w_2,b_2} \frac{c_4}{2}(||w_2||^2 + b_2^2) + \frac{1}{2}(\eta_2^T\eta_2 + c_2e_1^T\eta_1) \\ &s.t.\ Bw_2 + e_2b_2 = \eta_2, \\ &\quad -(Aw_2 + e_1b_2) + e_1 = \eta_1, \end{aligned} \tag{17}$$

However, the optimal values of (16) and (17) may be very small since ξ_2 and η_1 could be negative. In this case, the QPPs may suffer negative infinity problem. In order to address this problem, following the notion of rough sets, a weighted linear loss function with weighted vectors v_1 and v_2 is designed. Then, the primal problems of weighted linear loss TSVM (WLTSVM) [36] can be written as

$$\begin{aligned} &\min_{w_1,b_1} \frac{c_3}{2}(||w_1||^2 + b_1^2) + \frac{1}{2}(\xi_1^T\xi_1 + c_1v_2^T\xi_2) \\ &s.t.\ Aw_1 + e_1b_1 = \xi_1, \\ &\quad Bw_1 + e_2b_1 + e_2 = \xi_2, \end{aligned} \tag{18}$$

$$\begin{aligned} &\min_{w_2,b_2} \frac{c_4}{2}(||w_2||^2 + b_2^2) + \frac{1}{2}(\eta_2^T\eta_2 + c_2v_1^T\eta_1) \\ &s.t.\ Bw_2 + e_2b_2 = \eta_2, \\ &\quad -(Aw_2 + e_1b_2) + e_1 = \eta_1, \end{aligned} \tag{19}$$

where $\upsilon_2 = (\upsilon_{21}, \upsilon_{22}, \cdots, \upsilon_{2m_2})^T$ and $\upsilon_1 = (\upsilon_{11}, \upsilon_{12}, \cdots, \upsilon_{1m_1})^T$ are calculated by the following formula

$$\upsilon_{2i} = \begin{cases} 10^{-4}, & if\ \xi_{2i} \geq J_1, \\ 1, & otherwise, \end{cases} \tag{20}$$

$$\upsilon_{1i} = \begin{cases} 10^{-4}, & if\ \eta_{1i} \geq J_2, \\ 1, & otherwise, \end{cases} \tag{21}$$

where $J_1 \geq 0$ and $J_2 \geq 0$ are parameters.

The solutions of (16) and (17) can be obtained by

$$\begin{cases} c_3w_1 + c_1B^Tv_2 + A^T(Aw_1 + e_1b_1) = 0 \\ c_3b_1 + c_1e_2^Tv_2 + e_1^T(Aw_1 + e_1b_1) = 0 \end{cases} \tag{22}$$

$$\begin{cases} c_4w_2 + c_1A^Tv_1 + B^T(Bw_2 + e_2b_2) = 0 \\ c_4b_2 + c_2e_1^Tv_1 + e_2^T(Bw_2 + e_2b_2) = 0 \end{cases} \tag{23}$$

In general, the training process of linear WLTSVM has been divided into two stages. In the first stage, WLTSVM will be initialized with $v_1 = e_1$ and $v_2 = e_2$. Then, we solve (16) and (17) to obtain ξ_2 and η_1 and set J_1 as the average value of $|\xi_2|$ and J_2 as the average value of $|\eta_1|$. In the second stage, based on the values calculated by (20) and (21), we obtain the solutions to (18) and (19), respectively.

Once the solutions of w_1, b_1 and w_2, b_2 are obtained from (22) and (23), the nonparallel hyperplanes are known. A new data points $x \in R^n$ is then assigned to positive class W_1 or negative class W_2 by

$$x \in W_k,\ k = \arg\min_{k=1,2}\{\frac{|w_1^T x + b_1|}{||w_1||}, \frac{|w_2^T x + b_2|}{||w_2||}\} \quad (24)$$

In addition, more detail about WLTSVM can be seen in [36].

III. WEIGHTED LINEAR LOSS PROJECTION TWIN SUPPORT VECTOR MACHINE

In this section, we present the linear and nonlinear version of weighted linear loss projection twin support vector machine (WLPTSVM) for binary classification, respectively.

A. LINEAR WLPTSVM

For a binary classification problem, similar to WLTSVM [36], by introducing the linear loss function, the primal problems of the linear loss projection twin support vector machine are expressed as

$$\begin{aligned} \min_{w_1}\ & \frac{c_3}{2}w_1^2 + (\frac{1}{2}\xi_1^T\xi_1 + c_1 e_2^T \xi_2) \\ s.t.\ & Aw_1 - \frac{1}{m_1}e_1e_1^T Aw_1 = \xi_1, \\ & e_2 - (Bw_1 - \frac{1}{m_1}e_2e_1^T Aw_1) = \xi_2, \end{aligned} \quad (25)$$

$$\begin{aligned} \min_{w_2}\ & \frac{c_4}{2}w_2^2 + (\frac{1}{2}\eta_2^T\eta_2 + c_2 e_1^T \eta_1) \\ s.t.\ & Bw_2 - \frac{1}{m_2}e_2e_2^T Bw_2 = \eta_2, \\ & Aw_2 - \frac{1}{m_2}e_1e_2^T Bw_2 + e_1 = \eta_1, \end{aligned} \quad (26)$$

where c_i $(i = 1, 2, 3, 4)$ are positive parameters, ξ_1, ξ_2, η_1 and η_2 are slack variables.

Observing the above formulas (25) and (26), we can find that the optimal values of empirical risks $\frac{1}{2}\xi_1^T\xi_1 + c_1e_2^T\xi_2$ and $\frac{1}{2}\eta_2^T\eta_2 + c_2e_1^T\eta_1$ may be very small since ξ_2 and η_1 could be negative. In order to avoid this possible infinity problem and balance the influence of each point to the projected class mean, following the notion of rough sets [38], we introduce the weighted linear loss function with the weighted vectors υ_1 and υ_2, and then present our WLPTSVM formulations as follows.

$$\begin{aligned} \min_{w_1}\ & \frac{c_3}{2}w_1^2 + (\frac{1}{2}\xi_1^T\xi_1 + c_1 \upsilon_2^T \xi_2) \\ s.t.\ & Aw_1 - \frac{1}{m_1}e_1e_1^T Aw_1 = \xi_1, \\ & e_2 - (Bw_1 - \frac{1}{m_1}e_2e_1^T Aw_1) = \xi_2, \end{aligned} \quad (27)$$

$$\begin{aligned} \min_{w_2}\ & \frac{c_4}{2}w_2^2 + (\frac{1}{2}\eta_2^T\eta_2 + c_2 \upsilon_1^T \eta_1) \\ s.t.\ & Bw_2 - \frac{1}{m_2}e_2e_2^T Bw_2 = \eta_2, \\ & Aw_2 - \frac{1}{m_2}e_1e_2^T Bw_2 + e_1 = \eta_1, \end{aligned} \quad (28)$$

where $\upsilon_2 = (\upsilon_{21}, \upsilon_{22}, \cdots, \upsilon_{2m_2})^T$ and $\upsilon_1 = (\upsilon_{11}, \upsilon_{12}, \cdots, \upsilon_{1m_1})^T$ are determined by the following formula

$$\upsilon_{2i} = \begin{cases} 10^{-4}, & if\ \xi_{2i} \geq J_1, \\ 1, & otherwise, \end{cases} \quad (29)$$

$$\upsilon_{1i} = \begin{cases} 10^{-4}, & if\ \eta_{1i} \geq J_2, \\ 1, & otherwise, \end{cases} \quad (30)$$

where $J_1 \geq 0$ and $J_2 \geq 0$ are parameters.

Before solving the problems (27) and (28), we give the geometric interpretation of the problem (27) + (29) while the problem (28) + (30) is similar. For (27), the first term in the objective function is to control the model complexity for seeking the optimal projection direction w_1. The second term in the objective function is to minimize the empirical risk, which tries to make the within-class variance of the projected samples of its own class is minimized, and meanwhile, the projected samples of the other class scatter away as far as possible. Moreover, the weighted vector υ_2 is to balance the influence of each point to the projected class mean. In the training process, the empirical risk also tries to achieve the desired consistency. Therefore, from this point of view, the problems (27) and (28) with (29) and (30) are superior to the corresponding ones in PTSVM.

The above problems can be solved by the following approximation algorithm. Consider the problem (27), and substitute the equality constrains into the objective function. Thus, we obtain

$$\begin{aligned} L(w_1) =& \frac{c_3}{2}w_1^2 + \frac{1}{2}(Aw_1 - \frac{1}{m_1}e_1e_1^TAw_1)^T(Aw_1 - \frac{1}{m_1}e_1e_1^TAw_1) \\ &+ c_1\upsilon_2^T[e_2 - (Bw_1 - \frac{1}{m_1}e_2e_1^TAw_1)] \\ =& \frac{c_3}{2}w_1^2 + \frac{1}{2}w_1^T(A - \frac{1}{m_1}e_1e_1^TA)^T(A - \frac{1}{m_1}e_1e_1^TA)w_1 \\ &+ c_1\upsilon_2^T[e_2 - (B - \frac{1}{m_1}e_2e_1^TA)w_1], \end{aligned} \quad (31)$$

Let $S_1 = (A - \frac{1}{m_1}e_1e_1^TA)^T(A - \frac{1}{m_1}e_1e_1^TA)$, $R_1 = B - \frac{1}{m_1}e_2e_1^TA$, the formula (31) translates into

$$L(w_1) = \frac{c_3}{2}w_1^2 + \frac{1}{2}w_1^TS_1w_1 + c_1\upsilon_2^T(e_2 - R_1w_1), \quad (32)$$

Setting the gradient of (32) with respect to w_1 to be zero, we can get

$$\frac{\partial L}{\partial w_1} = S_1w_1 + c_3w_1 - c_1R_1^T\upsilon_2 = 0, \quad (33)$$

Then, the solution to QPP (27) can be obtained from the systems of linear equation as follows.

$$(S_1 + c_3I_1)w_1 = c_1 \cdot R_1^T\upsilon_2, \quad (34)$$

where I_1 is an identity matrix.

Consider the problem (28), and substitute the equality constrains into the objective function. Similarly, we obtain

$$\begin{aligned}L(w_2)=&\frac{c_4}{2}w_2^2+\frac{1}{2}(Bw_2-\frac{1}{m_2}e_2e_2^TBw_2)^T(Bw_2-\frac{1}{m_2}e_2e_2^TBw_2)\\&+c_2\upsilon_1^T[e_1+(Aw_2-\frac{1}{m_2}e_1e_2^TBw_2)]\\=&\frac{c_4}{2}w_2^2+\frac{1}{2}w_2^T(B-\frac{1}{m_2}e_2e_2^TB)^T(B-\frac{1}{m_2}e_2e_2^TB)w_2\\&+c_2\upsilon_1^T[e_1+(A-\frac{1}{m_2}e_1e_2^TB)w_2],\end{aligned} \tag{35}$$

Let $S_2 = (B - \frac{1}{m_2}e_2e_2^TB)^T(B - \frac{1}{m_2}e_2e_2^TB)$, $R_2 = A - \frac{1}{m_2}e_1e_2^TB$, the formula (35) translates into

$$L(w_2) = \frac{c_4}{2}w_2^2 + \frac{1}{2}w_2^TS_2w_2 + c_2\upsilon_1^T(e_1 + R_2w_2), \tag{36}$$

Setting the gradient of (36) with respect to w_2 to be zero, we can get

$$\frac{\partial L}{\partial w_2} = S_2w_2 + c_4w_2 + c_2R_2^T\upsilon_1 = 0, \tag{37}$$

Then, the solution to QPP (28) can be obtained from the systems of linear equation as follows.

$$(S_2 + c_4I_2)w_2 = -c_2 \cdot R_2^T\upsilon_1, \tag{38}$$

where I_2 is an identity matrix.

In order to find suitable υ_2 and υ_1 defined in (29) and (30) and the approximate solutions of problems (27) + (29) and (28) + (30), a weight-setting method with two steps is constructed. Generally speaking, the first step is to solve problems (25) and (26) with the linear loss function and find the corresponding ξ_2 and η_1. The second step is to calculate υ_1 and υ_2 using the obtained ξ_2 and η_1, and then find the solutions of problems (27) and (28) using the obtained υ_1 and υ_2, and take these solutions as the approximate solutions required. Thus, the detailed algorithm is given as follows.

Algorithm 1 Linear WLPTSVM

Step 1. Given the training input matrices A and B. Set $\upsilon_1 = e_1, \upsilon_2 = e_2$, and obtain the solutions w_1^1 and w_2^1 of (34) and (38) with the proper penalty parameters c_i ($i = 1, 2, 3, 4$), respectively.

Step 2. Calculate the slack variables ξ_2^1 and η_1^1 from w_1^1 and w_2^1 in (27) and (28), then obtain υ_1^1 and υ_2^1 from (29) and (30), where $J_1 = |\xi_2^1|^{mean}$ and $J_2 = |\eta_1^1|^{mean}$.

Step 3. Find the solutions w_1^* and w_2^* of (34) and (38) with υ_1^1 and υ_2^1.

Step 4. Construct the decision as

$$x \in W_i, i = \arg\min_{k=1,2} \left| w_k^T(x - \frac{1}{m_k}\sum_{j=1}^{m_k} x_j^{(k)}) \right|, \tag{39}$$

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

B. NONLINEAR WLPTSVM

For nonlinear classification problem, first of all, we define $C^T = [A\, B]^T$ and choose an appropriate kernel function K, and then the primal problems of nonlinear version of the weighted linear loss projection twin support vector machine are expressed as follows.

$$\begin{aligned}\min_{w_1}\ &\frac{c_3}{2}w_1^2 + (\frac{1}{2}\xi_1^T\xi_1 + c_1\upsilon_2^T\xi_2)\\ s.t.\ &K(A, C^T)w_1 - \frac{1}{m_1}e_1e_1^TK(A, C^T)w_1 = \xi_1,\\ &e_2 - (K(B, C^T)w_1 - \frac{1}{m_1}e_2e_1^TK(A, C^T)w_1) = \xi_2,\end{aligned} \tag{40}$$

$$\begin{aligned}\min_{w_2}\ &\frac{c_4}{2}w_2^2 + (\frac{1}{2}\eta_2^T\eta_2 + c_2\upsilon_1^T\eta_1)\\ s.t.\ &K(B, C^T)w_2 - \frac{1}{m_2}e_2e_2^TK(B, C^T)w_2 = \eta_2,\\ &K(A, C^T)w_2 - \frac{1}{m_2}e_1e_2^TK(B, C^T)w_2 + e_1 = \eta_1,\end{aligned} \tag{41}$$

where $\upsilon_2 = (\upsilon_{21}, \upsilon_{22}, \cdots, \upsilon_{2m_2})^T$ and $\upsilon_1 = (\upsilon_{11}, \upsilon_{12}, \cdots, \upsilon_{1m_1})^T$ are determined by (29) and (30).

Similar to the linear case, assuming that J_1 and J_2 are determined, we can obtain the solutions to the problems (40) and (41) as follows.

$$(KerS_1 + c_3I_1)w_1 = c_1 \cdot M_1^T\upsilon_2, \tag{42}$$

$$(KerS_2 + c_4I_2)w_2 = -c_2 \cdot M_2^T\upsilon_1, \tag{43}$$

where I_1 and I_2 are identity matrices, and the matrices $KerS_1$, $KerS_2$, M_1 and M_2 are defined by

$$\begin{aligned}KerS_1 = &(K(A, C^T) - \frac{1}{m_1}e_1e_1^TK(A, C^T))^T \cdot\\ &(K(A, C^T) - \frac{1}{m_1}e_1e_1^TK(A, C^T)),\end{aligned} \tag{44}$$

$$\begin{aligned}KerS_2 = &(K(B, C^T) - \frac{1}{m_2}e_2e_2^TK(B, C^T))^T \cdot\\ &(K(B, C^T) - \frac{1}{m_2}e_2e_2^TK(B, C^T)),\end{aligned} \tag{45}$$

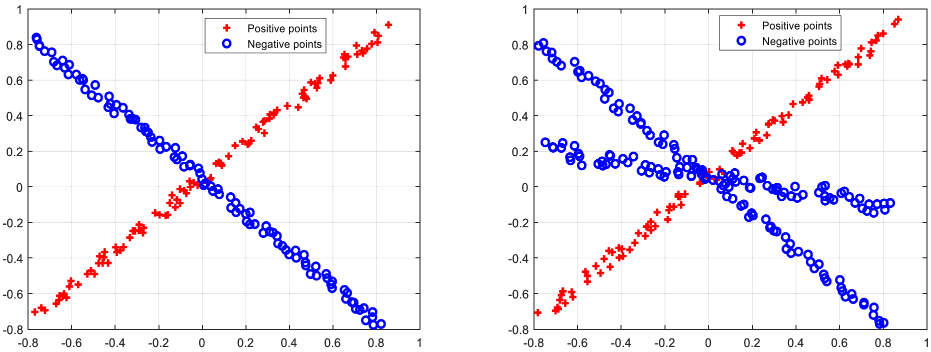
$$M_1 = K(B, C^T) - \frac{1}{m_1}e_2e_1^TK(A, C^T), \tag{46}$$

$$M_2 = K(A, C^T) - \frac{1}{m_2}e_1e_2^TK(B, C^T), \tag{47}$$

Similar to linear WLPTSVM, a weight-setting method with two steps for nonlinear WLPTSVM is constructed. The detailed algorithm is given in Algorithm 2 as follows.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

In order to evaluate our proposed WLPTSVM, we evaluate its classification accuracy and computational efficiency on synthetic datasets, UCI datasets [39] and David Musicant's NDC Data Generator datasets [40]. In our experiments, we focus on the comparison between the proposed WLPTSVM and several state-of-the-art algorithms, including



(a) (b)

FIGURE 1. Two synthetic datasets. (a) XOR. (b) Complex XOR.

TABLE 1. Classification accuracy on synthetic datasets.

Dataset	TWSVM	WLTSVM	PTSVM	RPTSVM	LSPTSVM	WLPTSVM
	Acc±std	Acc±Std	Acc±Std	Acc±Std	Acc±Std	Acc±Std
XOR	98. 92±0. 97	99. 25±0. 47	99. 36±0. 56	99. 33±0. 53	99. 38±0. 44	**99. 42±0. 69**
Complex XOR	90.32±2. 28	96. 73±1. 33	97. 76±2. 46	97.24±3. 81	97. 76±2. 12	**98. 72±0. 68**

Algorithm 2 Nonlinear WLPTSVM

Step 1. Given the training input matrices A and B. Set $\upsilon_1 = e_1, \upsilon_2 = e_2$, and obtain the solutions w_1^1 and w_2^1 of (42) and (43) with the proper penalty parameters c_i ($i = 1, 2, 3, 4$) and kernel function K, respectively.

Step 2. Calculate the slack variables ξ_2^1 and η_1^1 from w_1^1 and w_2^1 in (40) and (41), then obtain υ_1^1 and υ_2^1 from (29) and (30), where $J_1 = |\xi_2^1|^{mean}$ and $J_2 = |\eta_1^1|^{mean}$.

Step 3. Find the solutions w_1^* and w_2^* of (42) and (43) with υ_1^1 and υ_2^1.

Step 4. Construct the decision as

$$x \in W_i,\ i = \arg\min_{k=1,2} \left| w_k^T \cdot \left[K(x, C^T) - \frac{1}{m_k} \sum_{j=1}^{m_k} K(x_j^{(k)}, C^T) \right] \right|, \tag{48}$$

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

TWSVM [10], PTSVM [11], RPTSVM [24], LSPTSVM [25] and WLTSVM [36]. All above algorithms are implemented in MATLAB R2018a on a personal computer (PC) with an Intel (R) Core (TM) i7-7700CPU(3.60GHz×8) and 32 GB random-access memory (RAM). The "Accuracy", which is used to evaluate the performance of the algorithms, defined as Accuracy = (TP + TN) /(TP + FP + TN + FN), where TP, TN, FP and FN are the number of true positives, true negatives, false positives and false negatives, respectively. The QPPs in TWSVM, PTSVM and RPTSVM are solved by the optimization toolbox QP in MATLAB, while the systems of linear equations in LSPTSVM, WLTSVM and our WLPTSVM are solved by Hestenes-Stiefel conjugate gradient (CG) algorithm [37]. In addition, the positive penalty parameters c_i and kernel wide parameter σ of Gaussian kernel function $K(x, y) = e^{-\|x-y\|^2/2\sigma^2}$ in all algorithms are selected form the set $\{2^i | i = -8, -7, \cdots, 7, 8\}$ by using the standard 10-fold cross-validation methodology.

TABLE 2. The characteristics of benchmark datasets.

Datasets	#Samples	#Features	Datasets	#Samples	#Features
Australian	690	14	Musk	476	166
Bupa-Liver	345	6	PimaIndian	768	8
House-Votes	435	16	Sonar	208	60
Heart-c	303	13	Spect	267	44
Heart-Statlog	270	13	Wpbc	198	34
Ionosphere	351	34			

A. SYNTHETIC DATASETS

In this subsection, two synthetic datasets, including XOR and complex XOR datasets have been used to demonstrate that the proposed WLPTSVM can well solve linearly inseparable problems. In experiments, XOR dataset contains 200 samples (100 positive and 100 negative) and complex XOR dataset contains 260 samples (100 positive and 160 negative). Figure 1 illustrates XOR and complex XOR datasets. Specifically, for XOR and complex XOR datasets, we have investigated the performance of linear TWSVM, WLTSVM, PTSVM, RPTSVM, LSPTSVM and our WLPTSVM. We randomly select 40% for training sets and 60% for testing sets, each experiment repeat 10 times and the average results are listed in Table 1. From Table 1, we can observe that our proposed WLPTSVM obtains the best performance on XOR and complex XOR datasets.

TABLE 3. Test results of linear TWSVM, WLTSVM, PTSVM, RPTSVM, LSPTSVM and WLPTSVM.

Datasets	TWSVM	WLTSVM	PTSVM	RPTSVM	LSPTSVM	WLPTSVM
	Acc±Std (%) Time (s)	Acc±Std (%) Time (s)	Acc±Std (%) Time (s)	Acc±Std (%) Time (s)	Acc±Std (%) Time (s)	Acc±Std (%) Time (s)
Australian	87.13±0.30	86.12±0.18	86.83±0.43	86.93±0.24	86.30±0.12	**87.61±0.31**
	0.0190	0.0003	0.0195	0.0181	0.0002	0.0004
Bupa-Liver	**68.23±0.44**	66.41±1.20	67.68±0.90	67.51±0.82	66.08±0.67	67.97±0.65
	0.0056	0.0002	0.0062	0.0063	0.0001	0.0002
House-Votes	95.59±0.40	95.12±0.22	95.76±0.35	95.70±0.34	95.74±0.51	**95.82±0.24**
	0.0096	0.0003	0.0088	0.0093	0.0002	0.0003
Heart-c	84.92±0.41	85.09±0.50	85.08±0.48	84.66±0.42	84.53±0.42	**85.18±0.55**
	0.0056	0.0002	0.0051	0.0050	0.0002	0.0002
Heart-Statlog	85.07±0.35	85.52±0.44	85.44±0.43	84.96±0.19	84.85±0.21	**86.04±0.35**
	0.0046	0.0002	0.0044	0.0043	0.0001	0.0002
Ionosphere	**92.99±0.44**	91.43±0.34	92.30±0.33	90.85±0.70	90.48±0.66	92.59±0.49
	0.0064	0.0004	0.0060	0.0065	0.0002	0.0004
Musk	84.35±1.08	84.87±0.47	84.12±0.99	**85.83±0.84**	85.38±0.69	84.94±0.66
	0.0128	0.0022	0.0131	0.0121	0.0020	0.0020
PimaIndian	**77.22±0.52**	76.82±0.35	76.48±0.29	76.67±0.27	76.32±0.20	76.52±0.16
	0.0253	0.0031	0.0236	0.0237	0.0016	0.0030
Sonar	79.76±0.87	80.26±0.64	78.07±0.96	80.40±0.73	80.69±0.86	**81.68±1.05**
	0.0039	0.0004	0.0037	0.0035	0.0004	0.0004
Spect	80.45±0.32	80.52±0.04	80.28±0.38	80.33±0.39	80.18±0.40	**80.66±0.34**
	0.0059	0.0004	0.0059	0.0056	0.0003	0.0004
Wpbc	**81.55±0.70**	80.71±0.68	80.16±0.81	80.35±0.77	79.74±0.86	80.49±0.81
	0.0042	0.0003	0.0038	0.0038	0.0002	0.0003
Average	83.39±0.53	82.99±0.46	82.93±0.58	83.11±0.52	82.75±0.51	**83.59±0.51**
	0.0094	0.0007	0.0091	0.0089	0.0005	0.0005
Average rank	2.9091	3.5455	4.0909	3.7273	4.9091	1.8182

B. UCI DATASETS

In order to further compare our WLPTSVM with TWSVM, WLTSVM, PTSVM, RPTSVM and LSPTSVMM, we select 11 datasets from UCI machine learning repository [39]. Specifically, they are Australian, Bupa-Liver, House-Votes, Heart-c, Heart-Statlog, Ionosphere, Musk, PimaIndian, Sonar, Spect and Wpbc, respectively. The characteristics of these datasets are shown in Table 2.

Note that, we use the standard 10-fold cross-validation method to evaluate the performance of six algorithms. That means the dataset is divided randomly into ten subsets, one of those sets is reserved as a test set, and the others are regarded as a training set. This process is repeated ten times, and then the average of ten testing results is used as the performance measure. Specifically, the experimental results of their linear and nonlinear versions are given in Table 3 and Table 4, respectively. The best accuracy for each dataset is shown in bold font and the shortest CPU time is shown by underline for each dataset. In Table 3, we can find that the accuracy of our linear WLPTSVM is better than that of TWSVM, WLTSVM, PTSVM, RPTSVM and LSPTSVM on most of the datasets. Take the Heart-Statlog dataset for example, the accuracy of our WLPTSVM is 86.04%, while TWSVM is 85.07%, WLTSVM is 85.52%, PTSVM is 85.44%, RPTSVM is 84.96% and LSPTSVM is 84.85%, respectively. In addition, experimental results for nonlinear TWSVM, WLTSVM, PTSVM, RPTSVM, LSPTSVM and our WLPTSVM on above 11 UCI datasets are listed in Table 4. It is easy to find that the results in Table 4 are similar to those in Table 3. Especially for Sonar dataset, the accuracy of our nonlinear WLPTSVM obtains 90.43%, which is 1.44% higher than TWSVM, 0.98% higher than WLTSVM, 1.58% higher than PTSVM, 0.63% higher than RPTSVM and 2.06% higher than LSPTSVM, respectively. The average accuracy and CPU time for each algorithm are also reported in the penultimate row of Tables 3 and Table 4, which confirm that the proposed WLPTSVM also obtains the comparable classification accuracy with lower computational time.

Moreover, in order to make a statistic comparison on the effectiveness with the compared algorithms, Friedman test [41] is carried out. For this test, the average ranks of the compared algorithms on the selected datasets are listed in the last row of Table 3 and Table 4. Specifically, we consider $k(=6)$ number of compared algorithms and $n\ (=11)$ number of datasets. Let r_i^j be the rank of the j-th algorithms on the i-th datasets and assume all algorithms are equivalent under null hypothesis. Thus, the average rank of the j-th algorithm is calculated as

$$R_j = \frac{1}{n}\sum_{i=1}^{n} r_i^j, \tag{49}$$

The Friedman statistic is defined as

$$\chi_F^2 = \frac{12n}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right], \tag{50}$$

In fact, he Friedman statistic is distributed according to χ_F^2 with $(k-1)$ degrees of freedom, when n and k are reasonable large. Further, Iman and Davenport [42] showed that Friedman's χ_F^2 presents a pessimistic behavior.

TABLE 4. Test results of nonlinear TWSVM, WLTSVM, PTSVM, RPTSVM, LSPTSVM and WLPTSVM.

Datasets	TWSVM Acc+Std (%) Time (s)	WLTSVM Acc+Std (%) Time (s)	PTSVM Acc+Std (%) Time (s)	RPTSVM Acc+Std (%) Time (s)	LSPTSVM Acc+Std (%) Time (s)	WLPTSVM Acc+Std (%) Time (s)
Australian	87.39±0.25 0.0373	87.25±0.37 0.0246	87.22±0.29 0.0484	87.54±0.23 0.0462	87.58±0.44 0.0258	**87.83±0.35** 0.0251
Bupa-Liver	**72.66±0.46** 0.0098	71.86±0.89 0.0050	72.13±0.74 0.0129	72.27±0.81 0.0114	72.46±0.83 0.0058	72.17±0.93 0.0059
House-Votes	95.63±0.21 0.0150	95.54±0.43 0.0084	95.70±0.21 0.0201	95.72±0.24 0.0184	95.86±0.28 0.0089	**96.09±0.34** 0.0082
Heart-c	84.75±0.47 0.0081	84.90±0.63 0.0037	85.09±0.38 0.0105	84.76±0.40 0.0096	84.52±0.41 0.0044	**85.11±0.45** 0.0038
Heart-Statlog	84.52±0.55 0.0070	84.84±0.30 0.0030	84.44±0.46 0.0089	84.59±0.19 0.0078	84.44±0.26 0.0037	**85.14±0.34** 0.0040
Ionosphere	94.30±0.38 0.0112	94.42±0.45 0.0057	94.26±0.42 0.0143	**95.65±0.24** 0.0135	95.13±0.20 0.0062	95.17±0.19 0.0052
Musk	94.27±0.47 0.0192	94.18±0.58 0.0120	94.75±0.45 0.0257	**94.81±0.35** 0.0254	94.46±0.37 0.0132	94.37±0.63 0.0147
PimaIndian	**77.94±0.46** 0.0470	77.34±0.31 0.0332	76.94±0.52 0.0639	77.24±0.42 0.0589	77.19±0.32 0.0368	77.49±0.39 0.0349
Sonar	88.99±0.87 0.0051	89.45±0.81 0.0022	88.85±0.84 0.0065	89.80±0.75 0.0061	88.37±0.99 0.0026	**90.43±0.79** 0.0021
Spect	81.67±0.56 0.0082	81.64±0.39 0.0033	81.44±0.59 0.0102	81.49±0.76 0.0088	81.75±0.72 0.0038	**82.11±0.53** 0.0030
Wpbc	82.32±1.36 0.0052	81.32±0.28 0.0021	**82.78±1.64** 0.0067	80.82±0.90 0.0062	80.29±1.01 0.0022	81.39±0.85 0.0017
Average	85.86±0.55 0.0157	85.70±0.49 0.0094	85.78±0.59 0.0207	85.88±0.48 0.0193	85.67±0.53 0.0103	**86.12±0.53** 0.0099
Average rank	3.6364	4.1818	4.5455	3.1818	3.6364	1.9091

Thus, they derived a better statistic as follow

$$F_F = \frac{(n-1)\chi_F^2}{n(k-1)-\chi_F^2}, \tag{51}$$

which is distributed according to the F-distribution with $(k-1)$ and $(k-1)(n-1)$ degrees of freedom.

For linear case, in Table 3, our WLPTSVM ranks the first with an average score of 1.8182. To illustrate the measured average ranks are significantly different from the mean rank by the null hypothesis, according to (50) and (51), we obtain

$$\chi_F^2 = \frac{12\times 11}{6\times 7}[(2.9091^2 + 3.5455^2 + 4.0909^2 + 3.7273^2 + 4.9091^2 + 1.8182^2) - \frac{6\times 7^2}{4}] = 17.4956$$

$$F_F = \frac{10\times 17.4956}{11\times 5 - 17.4956} = 4.6649$$

Moreover, for 6 linear algorithms and 11 datasets, F_F is distributed according to the F-distribution with $(6-1)=5$ and $(6-1)\times(11-1) = 50$ degrees of freedom. Thus, we find that the critical value of $F(5,50)$ is 2.400 for the level of significant $\alpha = 0.05$ and it is less than the value of $F_F = 4.6649$, which indicates the null hypothesis is rejected. It means that the compared algorithms are significantly different on selected datasets. Similarly, in Table 4, we can find that the nonlinear WLPTSVM ranks the first with an average score of 1.9091. According to (50) and (51), we obtain

$$\chi_F^2 = \frac{12\times 11}{6\times 7}[(3.6364^2 + 4.1818^2 + 4.5455^2 + 3.1818^2 + 3.6364^2 + 1.9091^2) - \frac{6\times 7^2}{4}] = 15.2879$$

$$F_F = \frac{10\times 15.2879}{11\times 5 - 15.2879} = 3.8497$$

TABLE 5. The characteristics of benchmark datasets.

Datasets	Training data	Testing data	Features
NDC-500	500	50	32
NDC-1000	1000	100	32
NDC-2000	2000	200	32
NDC-3000	3000	300	32
NDC-5000	5000	500	32
NDC-8000	8000	800	32
NDC-10000	10000	1000	32
NDC-20000	20000	2000	32

Thus, for 6 nonlinear algorithms and 11 selected datasets, the critical value of $F(5,50)$ is equal to 2.400 for the level of significant $\alpha = 0.05$ and it is also less than the value of $F_F = 3.8497$. Then, the null hypothesis is rejected and the compared nonlinear algorithms are significantly different.

C. NDC DATASETS

In this subsection, we conduct some experiments on large scale classification datasets and the David Musicants NDC Data Generator [40] is used to evaluate the computation time for various algorithms with respect to number of data points. Table 5 lists a description of NDC datasets, each dataset is divided into a training set and testing set. For experiments on NDC datasets, we fixed parameters of all algorithms to be the same (i.e. $c_i = 1, \sigma = 1$). The training accuracy, testing accuracy and training time are reported in Tables 6 and Table 7, respectively.

To be specific, Table 6 shows the comparison results for the linear TWSVM, WLTSVM, PTSVM, RPTSVM, LSPTSVM and our WLPTSVM on NDC datasets. In Table 6,

TABLE 6. Comparison on NDC datasets for linear classifiers.

Datasets	TWSVM Train/Test (%) Time (s)	WLTSVM Train/Test (%) Time (s)	PTSVM Train/Test (%) Time (s)	RPTSVM Train/Test (%) Time (s)	LSPTSVM Train/Test (%) Time (s)	WLPTSVM Train/Test (%) Time (s)
NDC-500	96.00/88.00 0.0478	95.20/88.00 0.0016	95.80/86.00 0.0167	95.60/82.00 0.0194	95.60/82.00 0.0018	96.00/88.00 0.0014
NDC-1000	95.30/92.00 0.0696	95.00/92.00 0.0020	95.40/93.00 0.0589	95.20/95.00 0.0585	95.10/94.00 0.0023	95.50/95.00 0.0015
NDC-2000	94.90/95.00 0.2996	95.05/93.50 0.0023	95.00/95.00 0.2783	94.90/95.00 0.2626	94.75/94.50 0.0027	95.60/93.00 0.0020
NDC-3000	95.17/94.33 0.8631	95.27/93.67 0.0030	94.83/95.33 0.7965	94.93/94.67 0.8174	94.63/94.67 0.0032	95.37/94.00 0.0022
NDC-5000	94.74/94.40 3.2282	95.02/93.20 0.0031	94.84/94.40 3.1229	94.72/94.40 2.8963	94.64/94.60 0.0034	95.00/94.60 0.0028
NDC-8000	94.94/92.37 10.2427	95.21/92.38 0.0033	94.85/92.00 10.9781	94.85/91.75 10.3830	94.71/91.00 0.0036	94.75/92.18 0.0032
NDC-10000	95.22/94.40 16.8586	95.38/93.80 0.0041	95.12/94.40 18.2475	95.13/94.50 18.1892	94.96/94.50 0.0046	95.25/94.10 0.0036
NDC-20000	94.77/94.50 127.3396	95.18/93.15 0.0092	94.75/94.25 131.2221	94.73/94.30 125.2716	94.59/93.45 0.0094	95.23/94.00 0.0109

TABLE 7. Comparison on NDC datasets for nonlinear classifiers.

Datasets	TWSVM Train/Test (%) Time (s)	WLTSVM Train/Test (%) Time (s)	PTSVM Train/Test (%) Time (s)	RPTSVM Train/Test (%) Time (s)	LSPTSVM Train/Test (%) Time (s)	WLPTSVM Train/Test (%) Time (s)
NDC-500	100/98.00 0.0337	97.80/94.00 0.0222	100/96.00 0.0478	97.80/94.00 0.0423	97.80/94.00 0.0236	98.00/96.00 0.0234
NDC-1000	100/99.00 0.1634	98.70/97.00 0.1142	100/99.00 0.2487	98.60/99.00 0.1782	98.60/99.00 0.1048	98.60/99.00 0.1029
NDC-2000	100/100 0.9054	98.95/98.50 0.7953	100/100 0.8950	99.05/98.50 0.8851	98.90/99.00 0.6869	99.05/98.50 0.7767
NDC-3000	100/100 2.2853	98.27/98.00 1.9094	100/100 3.1836	99.63/97.67 2.6098	99.50/98.00 1.7855	99.10/99.00 1.8104
NDC-5000	100/100 7.3329	98.54/98.80 7.0786	100/100 9.5435	99.84/99.60 8.7329	99.78/99.20 7.0439	98.60/98.20 6.8638
NDC-8000	100/100 25.8527	99.08/98.50 24.4834	100/100 32.2047	99.81/99.75 29.5591	99.81/99.75 24.5214	98.59/99.75 23.9211
NDC-10000	100/100 47.2758	99.39/99.50 45.2086	100/100 64.3014	99.91/99.80 59.7671	99.88/99.90 45.8844	100/99.80 45.5321
NDC-20000	100/100 383.5708	99.64/99.60 340.5155	100/100 437.8307	99.95/99.85 401.7089	99.93/99.90 344.7382	100/99.90 337.9132

TABLE 8. The computational complexity of six classifiers.

Algorithms	Methods	Complexity of linear case	Complexity of nonlinear case
TWSVM	QP	$O(2*(n+1)^3)+O(m_1^3)+O(m_2^3)$	$O(2*(m+1)^3)+O(m_1^3)+O(m_2^3)$
WLTSVM	CG	$O(4*(n+1)^2 r)$	$O(4*(m+1)^2 r)$
PTSVM	QP	$O(2*n^3)+O(m_1^3)+O(m_2^3)$	$O(2*m^3)+O(m_1^3)+O(m_2^3)$
RPTSVM	QP	$O(2*n^3)+O(m_1^3)+O(m_2^3)$	$O(2*m^3)+O(m_1^3)+O(m_2^3)$
LSPTSVM	CG	$O(2*n^2 r)$	$O(2*m^2 r)$
WLPTSVM	CG	$O(4*n^2 r)$	$O(4*m^2 r)$

we can see that WLPTSVM obtains the comparable accuracies and performs faster than other algorithms on most datasets. For the nonlinear case, Table 7 shows the comparison results of all the algorithms conducted on NDC datasets with Gaussian kernel. The results on these datasets show that WLTSVM, LSPTSVM and our WLPTSVM are much faster than TWSVM, PTSVM and RPTSVM. The reason might be that the QPPs in TWSVM, PTSVM and RPTSVM are solved by the optimization toolbox QP in MATLAB, while the systems of linear equations in WLTSVM, LSPTSVM

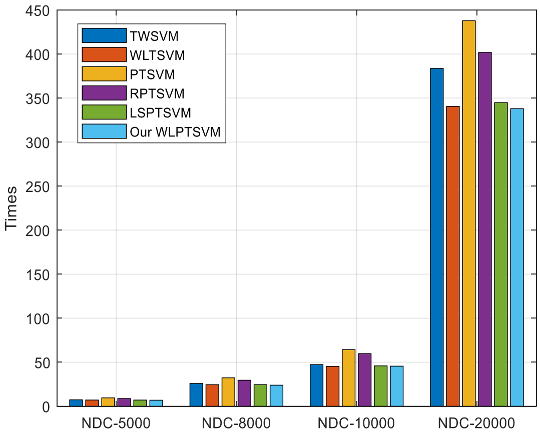


FIGURE 2. The training times of large scale NDC datasets.

and our WLPTSVM are solved by CG. For the linear case, taking NDC-20000 dataset for example, the training time of TWSVM is 127.3396 second, PTSVM is 131.2221 second and RPTSVM is 125.2716 second, while WLTSVM is 0.0092 second, LSPTSVM is 0.0094 second and our proposed WLPTSVM is 0.0109 second, respectively. For the nonlinear case, the training times of all algorithms on four large scale NDC datasets, e.g. NDC-5000, NDC-8000, NDC-10000 and NDC-20000, are shown in Figure 2. So, the results of Table 6, Table 7 and Figure 2 can indicate the efficiency of our WLPTSVM when dealing with large scale problems.

D. DISCUSSIONS

In this subsection, we will give some discussions about our WLPTSVM. First, according to optimization problems (27)-(28) for the linear case and (40)-(41) for the nonlinear case, there are many parameters, e.g. the penalty parameters c_1, c_2, c_3, c_4 and kernel wide parameter σ for nonlinear case. However, these parameters may significantly impact the performance of WLPTSVM. In order to investigate the influence of these parameters to the proposed method, we discuss their effect to the classification performance to our WLPTSVM on 2 UCI datasets, e.g. Australian, House-Votes. For simplicity, the parameters are set $c_1 = c_2$ and $c_3 = c_4$ for linear case and set $c_1 = c_2 = c_3 = c_4$ for nonlinear case. Figure 3 and Figure 4 show the influence of the parameters on accuracy with linear and nonlinear cases on above selected datasets, respectively.

Second, in our experiments, we have compared the performance of our WLPTSVM and other five algorithms. As we know, TWSVM, PTSVM and RPTSVM need to solve two quadratic programming problems (QPPs), while LSPTSVM, WLTSVM and WLPTSVM only need to solve two systems of linear equations. Specifically, the main computational times of TWSVM, PTSVM and RPTSVM are consumed in solving two inverse matrices and two QPPs, while the main computational time is consumed in solving two systems of linear equations for LSPTSVM and two systems of linear equations for WLTSVM and WLPTSVM twice. The QPPs are solved by the optimization toolbox QP in MATLAB, while the systems of linear equations are solved by CG. We analyzed the computational complexity of our WLPTSVM as follows. According to algorithm 1 and algorithm 2, the systems of linear equations (34), (38) or (42), (43) need to be solved in our WLPTSVM. It is not hard to find that the matrices of (34) and (38) are of dimension $n \times n$, while the matrices of (42) and (43) are of dimension $m \times m$, where $m = m_1 + m_2$ is the total number of training samples, m_1 and m_2 are the number of positive and negative training samples. For large values of m or n, these matrices cannot be stored. Thus, similar to WLTSVM, CG algorithm is used to solve our WLPTSVM. As we know, the computational complexity of the direct method to solve systems of linear equations is $O(n^3)$. Fortunately, by using the CG algorithm, the computational complexity of our linear WLPTSVM is $O(2 * n^2 r)$, where n is the dimension of samples and r is the number of iterations. However, it should be noted that the solution of linear WLPTSVM requires inversion of matrix of size $n \times n$ twice and the solution of nonlinear WLPTSVM requires inversion of matrix of size $m \times m$ twice. Thus, if the

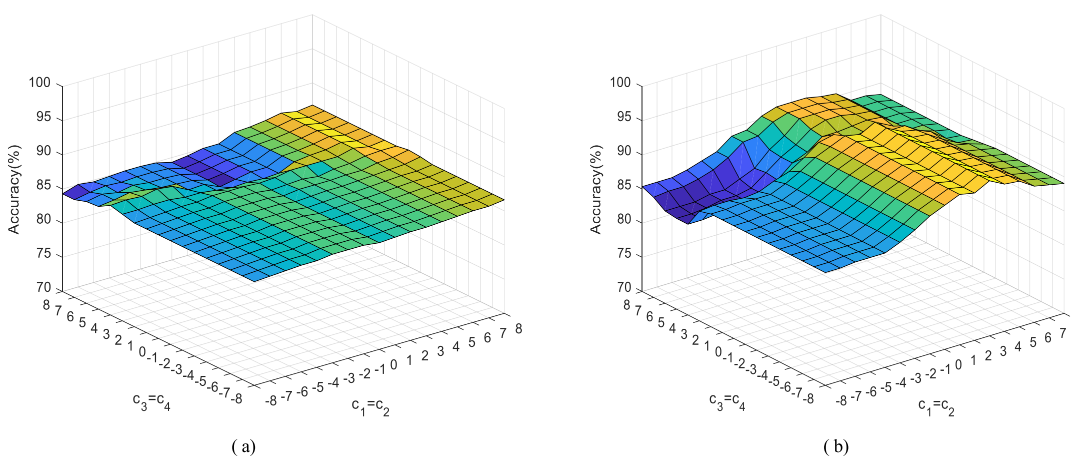


FIGURE 3. The influence of the parameters for linear WLPTSVM on Australian and House-votes datasets. (a) Australian. (b) House-votes.

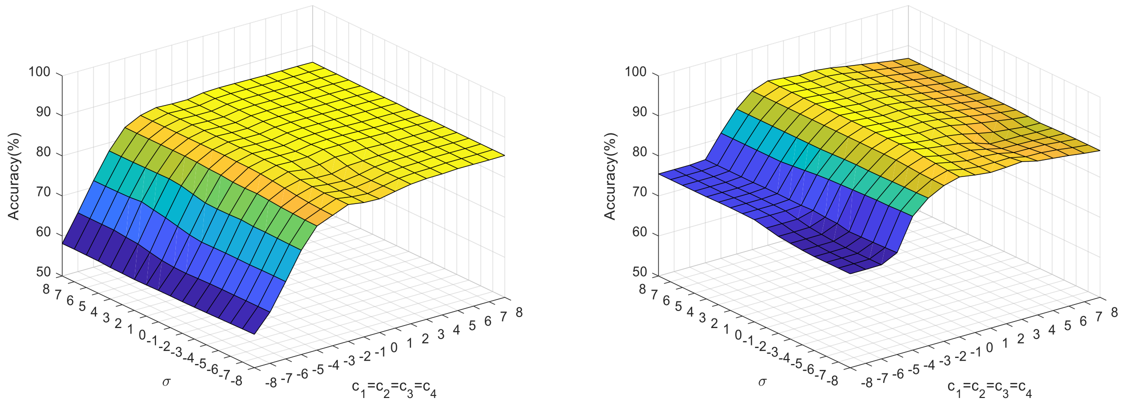


FIGURE 4. The influence of the parameters for nonlinear WLPTSVM on Australian and House-votes datasets. (a) Australian. (b) House-votes.

number of samples becomes very large, the reduced kernel technique [43] may be utilized to reduce the dimensionality for our nonlinear WLPTSVM. Similarly, we can analyze the computational complexity of other five algorithms. In a word, the detailed comparisons are reported in Table 8. From Table 8, we can find that LSPTSVM has the lowest computational complexity and the computational complexity of WLTSVM and WLPTSVM is almost the same, but lower than TWSVM, PTSVM and RPTSVM.

Third, in this paper, we only have proposed the algorithm for binary classification. However, multi-class classification problems are also common in real-world applications. In fact, our proposed WLPTSVM can be easily extended to multi-class classification problem by the one-versus-one, one-versus-rest strategies. Take K-class classification for example, for one-versus-one strategy, it needs to consider the samples of two classes for each binary classifier and establish $K(K-1)/2$ binary classifiers. For one-versus-rest strategy, it needs to construct K binary classifiers and the samples of one class are trained with the rest samples from the other classes for each binary classifier. In general, how to effectively extend binary classifier to multi-class classifier is also an interesting issue, which may be our future work.

At last, according to [11], [44], [45], we can find that recursive procedure to seek more than one directions for each class maybe can boost the performance. Therefore, how to extend our WLPTSVM to recursive case is also an interesting problem and we will address it in the future.

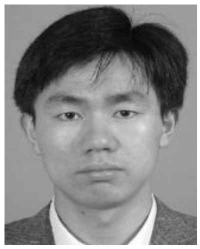
V. CONCLUSIONS

In this paper, we have proposed a novel weighted linear loss projection twin support vector machine (WLPTSVM) for binary classification problems. Instead of solving dual QPPs in PTSVM, our WLPTSVM finds two projection directions by solving systems of linear equations, allowing it to classify large datasets efficiently. Experimental results on synthetic and several benchmark datasets illustrate that our proposed WLPTSVM obtains comparable classification accuracy to that of PTSVM, but with reduced computational cost. It should be pointed out that there are many parameters in our WLPTSVM, so parameter selection is a practical problem and needs to be investigated in the future. In addition, the extension of our WLPTSVM to multi-class classification [46]–[48], multi-label classification [49] and feature selection problems [50], [51] are also interesting. Furthermore, how to use our WLPTSVM to deal with the large-scale classification problems in real world is also under our consideration.

REFERENCES

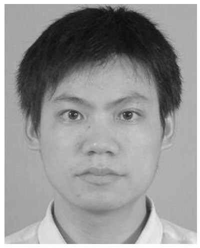
[1] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
[2] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer, 1995.
[3] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 1997, pp. 130–136.
[4] W. S. Noble, "Kernel methods in computational biology," in *Support Vector Machine Applications in Computational Biology*. Cambridge, MA, USA: MIT, 2004, pp. 71–92.
[5] 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 Trans. Knowl. Data Eng.*, vol. 20, no. 9, pp. 1264–1272, Sep. 2008.
[6] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods: Support Vector Learning*. Cambridge, MA, USA: MIT Press, 1999, pp. 185–208.
[7] C. C. Chang and C. J. Lin, "Libsvm: A Library for support vector machines," Dept. Comput. Sci. Inf. Eng., Nat. Taiwan Univ., Taipei, Taiwan, 2001. [Online]. Available: https://www.csie.ntu.edu.tw/~cjlin/libsvm/
[8] G. M. Fung and O. L. Mangasarian, "Multicategory proximal support vector machine classifiers," *Mach. Learn.*, vol. 59, nos. 1–2, pp. 77–97, May 2005.
[9] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, Jun. 1999.
[10] Jayadeva, R. Khemchandai, and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 905–910, May 2007.
[11] X. Chen, J. Yang, Q. Ye, and J. Liang, "Recursive projection twin support vector machine via within-class variance minimization," *Pattern Recognit.*, vol. 44, nos. 10–11, pp. 2643–2655, Oct. 2011.
[12] O. Mangasarian and E. Wild, "Multisurface proximal support vector machine classification via generalized eigenvalues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 69–74, Jan. 2006.
[13] M. A. Kumar and M. Gopal, "Least squares twin support vector machines for pattern classification," *Expert Syst. Appl.*, vol. 36, no. 4, pp. 7535–7543, May 2009.

[14] Y. H. Shao, C. H. Zhang, X. B. Wang, and N. Y. Deng, ''Improvements on twin support vector machines,'' *IEEE Trans. Neural Netw.*, vol. 22, no. 6, pp. 962–968, May 2011.
[15] X. Peng, ''TPMSVM: A novel twin parametric-margin support vector machine for pattern recognition,'' *Pattern Recognit.*, vol. 44, nos. 10–11, pp. 2678–2692, Oct. 2011.
[16] Z. Qi, Y. Tian, and Y. Shi, ''Robust twin support vector machine for pattern classification,'' *Pattern Recognit.*, vol. 46, no. 1, pp. 305–316, Jan. 2013.
[17] Y. Tian, Z. Qi, X. Ju, Y. Shi, and X. Liu, ''Nonparallel support vector machines for pattern classification,'' *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1067–1079, Jul. 2014.
[18] X. Ma, Q. L. Ye, and H. Yan, ''L2P-norm distance twin support vector machine,'' *IEEE Access*, vol. 5, pp. 23473–23483, Oct. 2017.
[19] R. Khemchandani, P. Saigal, and S. Chandra, ''Angle-based twin support vector machine,'' *Ann. Oper. Res.*, vol. 269, nos. 1–2, pp. 387–417, Oct. 2018.
[20] S.-G. Chen and X.-J. Wu, ''A new fuzzy twin support vector machine for pattern classification,'' *Int. J. Mach. Learn.*, vol. 9, no. 9, pp. 1553–1564, Sep. 2018.
[21] S. Ding, X. Hua, and J. Yu, ''An overview on nonparallel hyperplane support vector machine algorithms,'' *Neural Comput. Appl.*, vol. 25, no. 5, pp. 975–982, Oct. 2014.
[22] S. Ding, N. Zhang, X. Zhang, and F. Wu, ''Twin support vector machine: Theory, algorithm and applications,'' *Neural Comput. Appl.*, vol. 28, no. 11, pp. 3119–3130, Nov. 2017.
[23] Q. Ye, C. Zhao, N. Ye, and Y. Chen, ''Multi-weight vector projection support vector machines,'' *Pattern Recognit. Lett.*, vol. 31, no. 13, pp. 2006–2011, Oct. 2010.
[24] Y.-H. Shao, Z. Wang, W.-J. Chen, and N.-Y. Deng, ''A regularization for the projection twin support vector machine,'' *Knowl.-Based Syst.*, vol. 37, pp. 203–210, Jan. 2013.
[25] Y. Shao, N. Deng, and Z. Yang, ''Least squares recursive projection twin support vector machine for classification,'' *Pattern Recognit.*, vol. 45, no. 6, pp. 2299–2307, Jun. 2012.
[26] S. Ding and X. Hua, ''Recursive least squares projection twin support vector machines for nonlinear classification,'' *Neurocomputing*, vol. 130, pp. 3–9, Apr. 2014.
[27] X. Xie, ''Improvement on projection twin support vector machine,'' *Neural Comput. Appl.*, vol. 30, no. 2, pp. 371–387, Jul. 2018.
[28] X. Hua and S. Ding, ''Weighted least squares projection twin support vector machines with local information,'' *Neurocomputing*, vol. 160, pp. 228–237, Jul. 2015.
[29] Q. Hou, L. Liu, L. Zhen, and L. Jing, ''A novel projection nonparallel support vector machine for pattern classification,'' *Eng. Appl. Artif. Intell.*, vol. 75, pp. 64–75, Oct. 2018.
[30] S. Chen, X. Wu, and H. Yin, ''A novel projection twin support vector machine for binary classification,'' *Soft Comput.*, vol. 23, no. 2, pp. 655–668, Jan. 2019.
[31] X. J. Peng and D. Chen, ''PTSVRs: Regression models via projection twin support vector machine,'' *Inf. Sci.*, vol. 435, pp. 1–14, Apr. 2018.
[32] Z. Gu, Z. Zhang, J. Sun, and B. Li, ''Robust image recognition by L1-norm twin-projection support vector machine,'' *Neurocomputing*, vol. 223, pp. 1–11, Feb. 2017.
[33] W.-J. Chen, C.-N. Li, Y.-H. Shao, J. Zhang, and N.-Y. Deng, ''Robust L_1-norm multi-weight vector projection support vector machine with efficient algorithm,'' *Neurocomputing*, vol. 315, pp. 345–361, Nov. 2018.
[34] H. Zhao, Q. Ye, M. Naiem, and L. Fu, ''Robust $L_{2,1}$-norm distance enhanced multi-weight vector projection support vector machine,'' *IEEE Access*, vol. 7, pp. 3275–3286, Jan. 2019.
[35] S. Mehrkanoon, X. L. Huang, and J. A. K. Suykens, ''Non-parallel support vector classifiers with different loss functions,'' *Neurocomputing*, vol. 143, pp. 294–301, Nov. 2014.
[36] Y. Shao, W. Chen, Z. Wang, and N. Deng, ''Weighted linear loss twin support vector machine for large-scale classification,'' *Knowl.-Based Syst.*, vol. 73, pp. 276–288, Jan. 2015.
[37] Y. Saad, *Iterative Methods for Sparse Linear Systems*, vol. 82. Philadelphia, PA, USA: SIAM, 2003.
[38] J. A. K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle, ''Weighted least squares support vector machines: Robustness and sparse approximation,'' *Neurocomputing*, vol. 48, nos. 1–4, pp. 85–105, Oct. 2002.
[39] C. L. Blake and C. J. Merz, ''UCI repository of machine learning databases,'' Dept. Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, 1998. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html
[40] D. R. Musicant, ''NDC: Normally distributed clustered datasets,'' Dept. Comput. Sci., Univ. Wisconsin, Madison, Madison, WI, USA, 1998. [Online]. Available: http://research.cs.wisc.edu/dmi/svm/ndc/
[41] J. Demšar, ''Statistical comparisons of classifiers over multiple data sets,'' *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.
[42] R. L. Iman and J. M. Davenport, ''Approximations of the critical region of the fbietkan statistic,'' *Commun. Statist.-Theory Methods*, vol. 9, no. 6, pp. 571–595, Apr. 1980.
[43] Y. J. Lee and O. L. Mangasarian, ''RSVM: Reduced support vector machines,'' in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2001, pp. 1–17.
[44] C. Xiang, X. A. Fan, and T. H. Lee, ''Face recognition using recursive Fisher linear discriminant,'' *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2097–2105, Aug. 2006.
[45] Q. Tao, D. Chu, and J. Wang, ''Recursive support vector machines for dimensionality reduction,'' *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 189–193, Jan. 2008.
[46] Z.-M. Yang, H.-J. Wu, C.-N. Li, and Y.-H. Shao, ''Least squares recursive projection twin support vector machine for multi-class classification,'' *Int. J. Mach. Learn. Cybern.*, vol. 7, no. 3, pp. 411–426, Jun. 2016.
[47] S.-G. Chen and X.-J. Wu, ''Multiple birth least squares support vector machine for multi-class classification,'' *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 6, pp. 1731–1742, Dec. 2017.
[48] S. Ding, X. Zhang, Y. An, and X. Yu, ''Weighted linear loss multiple birth support vector machine based on information granulation for multi-class classification,'' *Pattern Recognit.*, vol. 67, pp. 32–46, Jul. 2017.
[49] Z. Hanifelou, P. Adibi, S. A. Monadjemi, and H. Karshenas, ''KNN-based multi-label twin support vector machine with priority of labels,'' *Neurocomputing*, vol. 322, pp. 177–186, Dec. 2018.
[50] P. Yi, A. Song, J. Guo, and R. Wang, ''Regularization feature selection projection twin support vector machine via exterior penalty,'' *Neural Comput. Appl.*, vol. 28, no. 1, pp. 683–697, Dec. 2017.
[51] R. Yan, Q. L. Ye, L. Y. Zhang, N. Ye, and X. B. Shu, ''A feature selection method for projection twin support vector machine,'' *Neural Process. Lett.*, vol. 47, no. 1, pp. 21–38, Feb. 2018.



SUGEN CHEN received the B.S. degree in mathematics and applied mathematics from Anqing Normal University, Anqing, Anhui, in 2004, the M.S. degree in computational mathematics from the Hefei University of Technology, Hefei, Anhui, in 2009, and the Ph.D. degree in control science and engineering from Jiangnan University, Wuxi, Jiangsu, in 2016. Since 2015, he has been an Associate Professor with the School of Mathematics and Computational Science, Anqing Normal University. His research interests include pattern recognition and intelligent systems, and machine learning.

JUNFENG CAO received the B.S. degree in mathematical education from Jiangsu Normal University, Xuzhou, China, in 2002, the M.S. degree in fundamental mathematics from Nanjing Normal University, Nanjing, China, in 2008, and the Ph.D. degree in control science and engineering from Jiangnan University, Wuxi, China, in 2017. His research interests include image segmentation and edge detection.



ZHONG HUANG received the B.S. degree in computer science and technology from Anqing Normal University, Anqing, Anhui, China, in 2005, and the M.S. degree in computer software and theory and the Ph.D. degree in computer application technology from the Hefei University of Technology, Hefei, Anhui, China, in 2008 and 2016, respectively. Since 2018, he has been an Associate Professor with the School of Physics and Electrical Engineering, Anqing Normal University. His research interests include humanoid robots, affective computing, and computer vision.