IJAMML 2:1 (2015) 27-45 March 2015
ISSN: 2394-2258
Available at http://scientificadvances.co.in

APPLICATION OF SMOOTHING TECHNIQUE ON PROJECTIVE TSVM

Xinxin Zhang and Liya Fan

School of Mathematics Sciences, Liaocheng University, 252059, P. R. China

Abstract

This paper is devoted to extend projective twin support vector machine (PTSVM) by smoothing technique and propose a navel classification method with linear and nonlinear versions, named as smoothed projective twin support vector machine (SPTSVM). The advantage of SPTSVM is to solve a pair of unconstraint differentiable optimization problems rather than a pair of dual QPPs. By means of Newton-Armijo method, an effective fast algorithm is suggested for solving SPTSVM. Experiment results compared with SVM, TSVM, PTSVM, and SPTSVM show that the proposed SPTSVM is a fast and effective classification method.

Keywords: projective twin support vector machine, plus function, smoothing approximation function, unconstraint differentiable optimization, Newton-Armijo method.

*Corresponding author.

E-mail address: fanliya63@126.com (Liya Fan).

2010 Mathematics Subject Classification: 68W40, 68Q25.
Submitted by Jianqiang Gao.
Received February 7, 2015

1. Introduction

Support vector machines (SVMs), being computationally powerful tools for classification and regression problems [1-4], have been successfully applied to a wide variety of real-world problems, such as financial forecasting [5], computational biology [6], optimal control [7], image segmentation [8], time series prediction [9] and so on. The main idea of SVM is to find an optimal separating hyperplane by maximizing the margin between two parallel boundary hyperplanes of positive and negative examples.

Different from SVM, multisurface proximal support vector machine via generalized eigenvalues (GEPSVM) [10], makes binary classification by two nonparallel hyperplanes, one for each class. In this approach, data points of each class are clustered around the corresponding hyperplane. A new input will be assigned to a class based on its proximity to one of the two hyperplanes. This formulation leads to two generalized eigenvalue problems. Based on GEPSVM, twin support vector machine (TSVM) proposed by Jayadeva and Chandra [11] constructs a pair of nonparallel hyperplanes by solving two smaller size QPPs rather than a single quadratic programming problem (QPP) such that each one is as close as possible to one class, and as far as possible from the other class. A new input will be assigned to one of the classes depending on its proximity to which hyperplane. Experiments show that TSVM is faster than SVM [11, 12].

Different from TSVM that seeks a hyperplane for each class by using SVM-type formulation, a multi-weight vector projection support vector machine (MVSVM) [13] is proposed by seeking one weight vector, such that the samples of one class are closest to its class meanwhile the samples of different classes are separated as far as possible. The weight vectors of MVSVM can be found by solving a pair of eigenvalue problems. By means of MVSVM and TSVM, projective twin support vector machine (PTSVM) proposed by Chen et al. [14] seeks a projection axis for each class by solving an associated SVM-type QPP such that the projected samples are well separated from those of the other class in its respective subspace.

In order to enhance the performance of PTSVM, inspired by the works in [15], we will introduce smoothing technique into PTSVM in this paper and develop a new classification method, termed as smoothed PTSVM (SPTSVM). This approach will results in solving a pair of unconstraint differentiable optimization problems rather than a pair of dual QPPs. In addition, we will use an effective fast algorithm to solve SPTSVM, which is called Newton-Armijo algorithm. To verifying the effectiveness of SPTSVM, we perform a series of comparative experiments with SVM, TSVM, and PTSVM on 10 datasets taken from UCI database and 6 datasets taken from NDC database.

The rest of the paper is organized as follows. We briefly review TSVM and PTSVM in Section 2 and propose SPTSVM with linear and nonlinear versions in Section 3. A series of comparative experiments are performed in Section 4 and some conclusions are given in Section 5.

2. Related Works

In this section, we review TSVM and PTSVM briefly, for details, see [11, 12, 14]. Let $\{(x_j^{(i)}, y_j^{(i)})\}_{j=1}^{m_i}, i = 1, 2$ be a sample set of data for a binary classification problem, where $i = 1$ denotes the positive class, $i = -1$ the negative class and $x_j^{(i)} \in R^n$ and $y_j^{(i)} \in \{-1, 1\}$ are the input and class label of j-th sample belonging to class i, respectively. Let $A = [x_1^{(1)}, \cdots, x_{m_1}^{(1)}]^T \in R^{m_1 \times n}$ and $B = [x_1^{(2)}, \cdots, x_{m_2}^{(2)}]^T \in R^{m_2 \times n}$ be input matrices and $m = m_1 + m_2$. Let $e_1 \in R^{m_1}$ and $e_2 \in R^{m_2}$ be vectors with elements being ones.

2.1. Twin support vector machine

Different from SVM finding the classification hyperplane by maximizing the margin between two parallel boundary hyperplanes, which involves the minimization of a quadratic programming problem (QPP), linear TSVM seeks two nonparallel hyperplanes

$$x^T w_1 + b_1 = 0 \text{ and } x^T w_2 + b_2 = 0, \tag{1}$$

by considering the following two QPPs:

$$\min_{w_1, b_1, \xi} \frac{1}{2}(Aw_1 + e_1 b_1)^T (Aw_1 + e_1 b_1) + \frac{c_1}{2} \xi^T \xi$$

$$s.t. -(Bw_1 + e_2 b_1) + \xi \geq e_2, \quad \xi \geq 0, \tag{2}$$

$$\min_{w_2, b_2, \eta} \frac{1}{2}(Bw_2 + e_2 b_2)^T (Bw_2 + e_2 b_2) + \frac{c_2}{2} \eta^T \eta$$

$$s.t. \ (Aw_2 + e_1 b_2) + \eta \geq e_1, \quad \eta \geq 0, \tag{3}$$

where $c_1, c_2 > 0$ are penalty parameters, $\xi \in R^{m_2}$ and $\eta \in R^{m_1}$ are vectors of slack variable and $w_1, w_2 \in R^n$ and $b_1, b_2 \in R$ are decision variables, such that each hyperplane is the closest to one class and the farthest from another class. A new input is assigned to one class depending on its proximity to the two nonparallel hyperplanes. By solving the Wolfe dual forms of the problems (2) and (3), respectively,

$$\min_{\alpha} \frac{1}{2} \alpha^T G(H^T H)^{-1} G^T \alpha - e_2^T \alpha$$

$$s.t. \ 0 \leq \alpha \leq c_1 e_2, \tag{4}$$

$$\min_{\beta} \frac{1}{2} \beta^T H(G^T G)^{-1} H^T \beta - e_1^T \beta$$

$$s.t. \ 0 \leq \beta \leq c_2 e_1, \tag{5}$$

where $\alpha \in R^{m_2}$ and $\beta \in R^{m_1}$ are vectors of Lagrange multipliers and $H = [A, e_1] \in R^{m_1 \times (n+1)}$, $G = [B, e_2] \in R^{m_2 \times (n+1)}$, we can obtain the optimal Lagrange multipliers vectors α^* and β^* and then deduce that

$$u_1^* = ((w_1^*)^T, b_1^*) = -(H^T H)^{-1} G^T \alpha^*,$$

$$u_2^* = ((w_2^*)^T, b_2^*) = -(G^T G)^{-1} H^T \beta^*. \tag{6}$$

It notes that since $H^T H$ and $G^T G$ are symmetric nonnegative definite matrices, we can regularize them if they are singular, that is, we can replace $H^T H$ and $G^T G$ by $H^T H + \varepsilon I_{n+1}$ and $G^T G + \varepsilon I_{n+1}$, respectively, where $\varepsilon > 0$ is a sufficiently small number and I_{n+1} denotes the $n+1$ order unit matrix. Consequently, the class label of a new input $x \in R^n$ can be assigned by

$$\text{class}(x) = \arg\min_{i=1,2} \frac{\left|(w_i^*)^T x + b_i^*\right|}{\|w_i^*\|}, \tag{7}$$

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

2.2. Projection twin support vector machine

The main idea of PTSVM is to find a projection axis for each class such that within-class variance of the projected samples of its own class is minimized meanwhile the projected samples of the other class scatter away as far as possible. This leads to the following two optimization problems:

$$\min_{w_1, \xi_k} \frac{1}{2} \sum_{i=1}^{m_1} \left(w_1^T x_i^{(1)} - w_1^T \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)}\right)^2 + c_1 \sum_{k=1}^{m_2} \xi_k$$

$$s.t.\ w_1^T x_k^{(2)} - w_1^T \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)} + \xi_k \geq 1; \tag{8}$$

$$\xi_k \geq 0,\ k = 1, 2, \cdots, m_2,$$

$$\min_{w_2, \eta_k} \frac{1}{2} \sum_{i=1}^{m_2} \left(w_2^T x_i^{(2)} - w_2^T \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)}\right)^2 + c_2 \sum_{k=1}^{m_1} \eta_k$$

$$s.t.\ -\left(w_2^T x_k^{(1)} - w_2^T \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)}\right) + \eta_k \geq 1; \tag{9}$$

$$\eta_k \geq 0,\ k = 1, 2, \cdots, m_1,$$

where $c_1, c_2 > 0$ are trade off constants, ξ_k and η_k are nonnegative slack variables. To simplify the above formulation, let

$$S_1 = \sum_{j=1}^{m_1} (x_i^{(1)} - \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)})(x_i^{(1)} - \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)})^T \in R^{n \times n},$$

$$S_2 = \sum_{j=1}^{m_2} (x_i^{(2)} - \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)})(x_i^{(2)} - \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)})^T \in R^{n \times n}.$$

Then, the problems (8) and (9) can be rewritten as follows, respectively,

$$\min_{w_1, \xi} \frac{1}{2} w_1^T S_1 w_1 + c_1 e_2^T \xi$$

$$s.t.\ Bw_1 - \frac{1}{m_1} e_2 e_1^T A w_1 + \xi \geq e_2, \quad \xi \geq 0, \tag{10}$$

$$\min_{w_2, \eta} \frac{1}{2} w_2^T S_2 w_2 + c_2 e_1^T \eta$$

$$s.t. -(Aw_2 - \frac{1}{m_2} e_1 e_2^T B w_2) + \eta \geq e_1, \quad \eta \geq 0. \tag{11}$$

By solving the Wolfe dual problems of the problems (10) and (11), respectively,

$$\min_{\alpha} \frac{1}{2} \alpha^T (B - \frac{1}{m_1} e_2 e_1^T A) S_1^{-1} (B^T - \frac{1}{m_1} A^T e_1 e_2^T) \alpha - e_2^T \alpha$$

$$s.t.\ 0 \leq \alpha \leq c_1 e_2,$$

$$\min_{\beta} \frac{1}{2} \beta^T (A - \frac{1}{m_2} e_1 e_2^T B) S_2^{-1} (A^T - \frac{1}{m_2} B^T e_2 e_1^T) \beta - e_1^T \beta$$

$$s.t.\ 0 \leq \beta \leq c_2 e_1.$$

We can obtain the optimal Lagrange multipliers vectors α^* and β^* and then deduce that

$$w_1^* = S_1^{-1}(B^T - \frac{1}{m_1} A^T e_1 e_2^T)\alpha^*,$$

$$w_2^* = S_2^{-1}(A^T - \frac{1}{m_2} B^T e_2 e_1^T)\beta^*.$$

It notes that since S_1 and S_2 are symmetric nonnegative definite matrices, we can regularize them if they are singular, that is, we can replace S_1 and S_2 by $S_1 + \varepsilon I_{n+1}$ and $S_2 + \varepsilon I_{n+1}$, respectively, where $\varepsilon > 0$ is a sufficiently small number and I_{n+1} denotes the $n + 1$ order unit matrix. Consequently, the class label of a new input $x \in R^n$ can be assigned by

$$\text{class}(x) = \arg\min_{i=1,2}\left|(w_i^*)^T x - (w_i^*)^T \frac{1}{m_1}\sum\nolimits_{j=1}^{m_i} x_j^{(i)}\right|.$$

3. Smoothed Projection Twin Support Machine

In order to enhance the performance of PTSVM, inspired by the works in [15], we will introduce smoothing technique into PTSVM in this section and develop a new classification method with linear and nonlinear versions, termed as smoothing projection twin support machine (SPTSVM). This approach will results in solving a pair of primal unconstraint differentiable optimization problems rather than a pair of dual QPPs. At the same time, we will propose an effective fast algorithm for SPTSVM by using Newton-Armijo method.

3.1. Linear SPTSVM

By introducing the plus function

$$(x)_+ = \max\{x, 0\}, \quad \forall x \in R,$$

$$(x)_+ = ((x_1)_+, \cdots, (x_n)_+)^T, \quad \forall x \in R^n,$$

the constraints of the primal problems (8) and (9) of PTSVM can be rewritten as, respectively,

$$\xi_k = (1 - w_1^T x_k^{(2)} + w_1^T \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)})_+, \quad k = 1, 2, \cdots, m_2,$$

$$\eta_k = (1 + w_2^T x_k^{(1)} - w_2^T \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)})_+, \quad k = 1, 2, \cdots, m_1,$$

$$\xi_+ = ((\xi_1)_+, \cdots, (\xi_{m_2})_+))^T = (e_2 + e_2\widetilde{A}w_1 - Bw_1)_+ \in R^{m_2},$$

$$\eta_+ = ((\eta_1)_+, \cdots, (\eta_{m_1})_+))^T = (e_1 - e_1\widetilde{B}w_2 + Aw_2)_+ \in R^{m_1},$$

where $\widetilde{A} = \frac{1}{m_1} \sum_{j=1}^{m_1} (x_j^{(1)})^T$ and $\widetilde{B} = \frac{1}{m_2} \sum_{j=1}^{m_2} (x_j^{(2)})^T$.

In order to avoid the singularity of the matrices S_1 and S_2 involved in PTSVM, we consider adding the generalization terms $\|w_1\|^2$ and $\|w_2\|^2$ in problems (8) and (9), respectively. In addition, for obtaining differentiable optimization problems, we replace one penalty for slack variables in problems (8) and (9) by two penalty. Consequently, we get two improved unconstraint optimization problems

$$\min_{w_1} \frac{1}{2} \left\| Aw_1 - e_1\widetilde{A}w_1 \right\|^2 + \frac{c_1}{2} \left\| (e_2 + e_2\widetilde{A}w_1 - Bw_1)_+ \right\|^2 + \frac{c_3}{2} \|w_1\|^2, \tag{12}$$

$$\min_{w_2} \frac{1}{2} \left\| Bw_2 - e_2\widetilde{B}w_2 \right\|^2 + \frac{c_2}{2} \left\| (e_1 - e_1\widetilde{B}w_2 + Aw_2)_+ \right\|^2 + \frac{c_4}{2} \|w_2\|^2. \tag{13}$$

We know that the plus function $(x)_+$ for $x \in R$ is not differentiable. So, we introduce a smoothing approximation function $\rho(x, \zeta)$ for $(x)_+$, where ζ is a smoothing parameter, and then the problems (12) and (13) can be further improved the following two unconstraint differentiable optimization problems:

$$\min_{w_1} f_1(w_1) = \frac{1}{2} \left\| Aw_1 - e_1\widetilde{A}w_1 \right\|^2 + \frac{c_1}{2} \left\| \rho(e_2 + e_2\widetilde{A}w_1 - Bw_1, \zeta) \right\|^2 + \frac{c_3}{2} \|w_1\|^2, \tag{14}$$

$$\min_{w_2} f_2(w_2) = \frac{1}{2}\left\|Bw_2 - e_2\widetilde{B}w_2\right\|^2 + \frac{c_2}{2}\left\|\rho(e_1 + Aw_2 - e_1\widetilde{B}w_2, \zeta)\right\|^2 + \frac{c_4}{2}\|w_2\|^2. \tag{15}$$

Recently, we find from literature [20-24] that there are some smoothing approximation functions for $(x)_+$. In this paper, we only use the smoothing approximation function $\rho(x, \zeta) = x + \frac{1}{\zeta}\ln(1 + e^{-\zeta x})$ presented in [20]. We can easily prove that the least square TSVM is a special case of smoothed TSVM with $\rho(x, \zeta) = x$.

Next, we propose a fast algorithm to solve the problems (14) and (15) based on Newton-Armijo algorithm. Newton-Armijo algorithm is one of the most popular iterative algorithms for solving unconstraint smooth optimization problems and has been shown to be quadratically convergent (see [15]). In the end, we firstly calculate the gradient vectors $\nabla f_i(w_i)$ and Hessian matrices $\nabla^2 f_i(w_i)$ of the objective functions of the problems (14) and (15)

$$\nabla f_1(w_1) = (A - e_1\widetilde{A})^T(A - e_1\widetilde{A})w_1 + c_1\sum_{i=1}^{m_2}\frac{\rho(z_{1i}, \zeta)}{1 + e^{-\alpha z_{1i}}}\cdot(\widetilde{A}^T - x_i^{(2)}) + c_3 w_1,$$

$$\nabla^2 f_1(w_1) = (A^T - \widetilde{A}^T e_1^T)(A - e_1\widetilde{A}) + c_1\sum_{i=1}^{m_2}\frac{1 + \zeta\rho(z_{1i}, \zeta)e^{-\alpha z_{1i}}}{(1 + e^{-\alpha z_i})^2}\cdot$$

$$(\widetilde{A}^T - x_i^{(2)})(\widetilde{A}^T - x_i^{(2)})^T + c_3 I,$$

$$\nabla f_2(w_2) = (B^T - \widetilde{B}^T e_2^T)(Bw_2 - e_2\widetilde{B}w_2) + c_2\sum_{i=1}^{m_1}\frac{\rho(z_{2i}, \zeta)}{1 + e^{-\alpha z_{2i}}}\cdot(x_i^{(1)} - \widetilde{B}^T) + c_4 w_2,$$

$$\nabla^2 f_2(w_2) = (B^T - \widetilde{B}^T e_2^T)(B - e_2\widetilde{B}) + c_2\sum_{i=1}^{m_1}\frac{1 + \zeta\rho(z_{2i}, \zeta)e^{-\alpha z_{2i}}}{(1 + e^{-\alpha z_i})^2}\cdot$$

$$(x_i^{(1)} - \widetilde{B}^T)(x_i^{(1)} - \widetilde{B}^T)^T + c_4 I,$$

where $z_{1i} = 1 + \widetilde{A}w_1 - (x_i^{(2)})^T w_1$, $z_{2i} = 1 - \widetilde{B}w_2 - (x_i^{(1)})^T w_2$ and I is the identity matrix of appropriate dimension.

Then we calculate the search direction d_t by Newton method (called Newton direction) and search stepsize λ_t by Armijo method (called Armijo stepsize) for t-th step iteration. The specific procedure is as follows, in which we only solve the problem (14), with the similar way we can solve the problem (15).

Algorithm 1. The Newton-Armijo algorithm for linear SPTSVM.

Step 1. Initialization. For given parameter values c_1, c_3 and the maximum number of iterations T, let $t = 0$, $\varepsilon > 0$ be small enough and take arbitrarily nonzero vector $w_1^t \in R^n$.

Step 2. Calculate Newton direction d_t by solving the system of linear equations $\nabla^2 f_1(w_1^t)d_t = -\nabla f_1(w_1^t)$.

Step 3. Calculate Armijo stepsize λ_t by inexact linear searching, that is, choose $\lambda_t = \max\left\{1, \frac{1}{2}, \frac{1}{4}, \cdots\right\}$ satisfying

$$f_1(w_1^t) - f_1(w_1^t + \lambda_t d_t) \geq -\frac{\lambda_t}{4}\nabla f_1(w_1^t)^T d_t.$$

Step 4. Update w_1^t. Calculate the next iterative point by formula $w_1^{t+1} = w_1^t + \lambda_t d_t$.

Step 5. If $\left\|w_1^{t+1} - w_1^t\right\| < \varepsilon$ or the maximum number of iterations T is achieved, stop iteration and take $w_1^* = w_1^{t+1}$; otherwise, put $t \leftarrow t+1$ and return to Step 2.

Step 6. The class label of a new input $x \in R^n$ is assigned by

$$\text{class}(x) = \arg\min_{i=1,2}\left|(w_i^*)^T x - (w_i^*)^T \frac{1}{m_i}\sum_{j=1}^{m_i} x_j^{(i)}\right|.$$

3.2. Nonlinear SPTSVM

In this subsection, we consider the nonlinear version of SPTSVM by means of the kernel skill (for details, see [25]). Let $k : R^n \times R^n \to R$ be a kernel function with the reproducing kernel Hilbert space (RKHS) H and the nonlinear feature mapping $\phi : R^n \to H$. Let $w_j = \sum_{i=1}^{m} \phi(x_i) u_{ji}$ and $u_j = (u_{j1}, \cdots, u_{jm})^T \in R^m$, $j = 1, 2$. Nonlinear SPTSVM seeks a projection axis for each class by considering the following two optimization problems:

$$\min_{u_1, \xi} \frac{1}{2} \left\| K(X_1, X) u_1 - e_1 \widetilde{A}_{\text{ker}} u_1 \right\|^2 + \frac{c_1}{2} \xi^T \xi + \frac{c_3}{2} \|u_1\|^2$$

$$s.t.\ K(X_2, X) u_1 - e_2 \widetilde{A}_{\text{ker}} u_1 + \xi \geq e_2, \quad \xi \geq 0, \tag{16}$$

$$\min_{u_2, \eta} \frac{1}{2} \left\| K(X_2, X) u_2 - e_2 \widetilde{B}_{\text{ker}} u_2 \right\|^2 + \frac{c_2}{2} \eta^T \eta + \frac{c_4}{2} \|u_2\|^2$$

$$s.t. -(K(X_1, X) u_2 - e_1 \widetilde{B}_{\text{ker}} u_2) + \eta \geq e_1, \quad \eta \geq 0, \tag{17}$$

where $X_1 = [x_1^{(1)}, \cdots, x_{m_1}^{(1)}]^T \in R^{m_1 \times n}$, $X_2 = [x_1^{(2)}, \cdots, x_{m_2}^{(2)}]^T \in R^{m_2 \times n}$, $X = [x_1, \cdots, x_m]^T \in R^{m \times n}$,

$$K(X_1, X) = \begin{bmatrix} k(x_1^{(1)}, x_1) & \cdots & k(x_1^{(1)}, x_m) \\ \vdots & \ddots & \vdots \\ k(x_{m_1}^{(1)}, x_1) & \cdots & k(x_{m_1}^{(1)}, x_m) \end{bmatrix} \in R^{m_1 \times m},$$

$$K(X_2, X) = \begin{bmatrix} k(x_1^{(2)}, x_1) & \cdots & k(x_1^{(2)}, x_m) \\ \vdots & \ddots & \vdots \\ k(x_{m_2}^{(2)}, x_1) & \cdots & k(x_{m_2}^{(2)}, x_m) \end{bmatrix} \in R^{m_2 \times m},$$

and $\widetilde{A}_{\text{ker}} = \dfrac{e_1^T K(X_1, X)}{m_1}$, $\widetilde{B}_{\text{ker}} = \dfrac{e_2^T K(X_2, X)}{m_2}$. Similar to the linear version, by introducing the smoothing approximation function $\rho(x, \zeta)$ for plus function and avoiding the singularity of the matrices, we can improve the problems (16) and (17) into the following forms:

$$\min_{u_1} \frac{1}{2}\left\|K(X_1, X)u_1 - e_1\widetilde{A}_{\text{ker}}u_1\right\|^2 + \frac{c_1}{2}\left\|\rho(e_2 - K(X_2, X)u_1 + e_2\widetilde{A}_{\text{ker}}u_1, \zeta)\right\|^2 + \frac{c_3}{2}\|u_1\|^2, \tag{18}$$

$$\min_{u_2} \frac{1}{2}\left\|K(X_2, X)u_2 - e_2\widetilde{B}_{\text{ker}}u_2\right\|^2 + \frac{c_2}{2}\left\|\rho(e_1 + K(X_1, X)u_2 - e_1\widetilde{B}_{\text{ker}}u_2, \zeta)\right\|^2 + \frac{c_4}{2}\|u_2\|^2. \tag{19}$$

For the problems (18) and (19), we also can propose a fast algorithm by means of Newton-Armijo method. In order to reduce the length of the paper, we will not repeat the algorithm description. After solving the optimal solutions u_1^* and u_2^* of the problems (18) and (19), the class label of a new input $x \in R^n$ can be assigned by

$$\text{class}(x) = \arg\min_{i=1,2}\left|K(x, X)u_i^* \frac{1}{m_i} e_i^T K(X_i, X)u_i^*\right|,$$

where $K(x, X) = [k(x_1, x), \cdots, k(x_m, x)]$.

4. Experiments

In this section, in order to verify the effectiveness of linear and nonlinear SPTSVM, we perform a series of comparative experiments on classification accuracy and running time with SVM, TSVM, and PTSVM by using 10 datasets taken from UCI database [26] and 6 datasets taken from NDC database [27], which are listed in Table 1.

Table 1. Description of NDC datasets

Dataset	Training data	Test data	Features
NDC-200	200	40	32
NDC-500	500	100	32
NDC-700	700	140	32
NDC-1000	1000	200	32
NDC-2000	2000	400	32
NDC-3000	3000	600	32

In order to save experimental time and without loss of generality, experiments with linear versions are implemented on all 16 datasets and nonlinear versions on only 4 datasets of UCI database in Matlab (7.11.0) R2010b environment on a PC with an Intel P4 processor (2.30GHz) with 4GB RAM. Take $\varepsilon = 10^{-3}$, $T = 50$ and the five-fold cross-validation method is used in all experiments. The classification accuracy is defined by

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN},$$

where TP, TN, FP, and FN denote the numbers of true positive, true negative, false positive, and false negative, respectively. For Iris, Vehicle, Waveform, and Balance four datasets with 3 classes, we choose, respectively, the later two classes for experiments.

We know that the performance of classifiers seriously depends on the choice of parameters, in order to facilitate the comparison, all the parameters involved in classification methods are selected as the same value for each dataset, which are obtained optimally from $\{2^{-8}, \cdots, 2^{8}\}$ by grid search. Specifically, in the linear case, take $c = c_1 = c_2 = c_3 = c_4 = 1$ for 10 datasets of UCI database and take $c = c_1 = c_2 = c_3 = c_4 = 2$ for 6 datasets of NDC database. In the nonlinear case, they are listed in Table 4.

The experiment results on 16 datasets with four linear classifiers are listed in Tables 2-3, respectively, and on 4 datasets with three nonlinear classifiers is listed in Tables 4. In nonlinear classifiers, the Gaussian RBF kernel function $k(x, y) = \exp(-\|x - y\|^2/\sigma^2), \forall x, y \in R^n$ is used and the selected kernel parameter σ is listed in Table 4.

From Table 2, we can see that the classification accuracy of SPTSVM is better than that of SVM, TSVM, and PTSVM except to Iris, Pima and Balance three datasets. From Table 3, we can see that the classification accuracies of TSVM, PTSVM, and SPTSVM are almost same. From Table 4, we can see that the classification accuracy of SPTSVM is much better than that of PTSVM and better than that of TSVM except to Breast dataset. In addition, we can see from Tables 2-4 that the running time of SPTSVM is much faster than that of SVM, TSVM, and PTSVM.

According to the above analysis, we can conclude that the proposed SPTSVM in this paper is a fast and effective classification method.

Table 2. Comparison results on UCI datasets with linear classifiers

Dataset	SVM
	Accuracy(%)
	Time(s)
		TSVM
		Accuracy(%)
		Time(s)
			PTSVM
			Accuracy(%)
			Time(s)
				SPTSVM
				Accuracy(%)
				Time(s)

Iris (150 × 4)	46.00	92.00	**97.00**	93.00
	2.1165	0.6764	0.4015	0.2544
Vehicle (150 × 18)	43.00	80.00	87.00	**96.00**
	0.9774	0.9261	0.9374	0.9073
Vote (435 × 16)	45.00	94.00	96.00	**96.00**
	22.3835	0.8148	1.2286	0.8878
Waveform (150 × 21)	48.00	85.00	94.00	**94.00**
	0.9975	1.7310	0.9770	0.7798
Heart (303 × 13)	40.00	80.42	71.50	**82.50**
	21.8097	1.4095	1.5077	0.9195
Breast (277 × 9)	29.09	70.91	67.50	**72.37**
	11.0981	1.0266	1.1766	0.5484
Liver (345 × 6)	47.59	50.69	51.00	**59.66**
	37.4085	2.9929	1.9903	0.9475
WBC (600 × 9)	47.73	94.45	96.47	**97.14**
	134.7745	2.1118	2.0755	1.0822
Pima (768 × 8)	43.53	77.52	**77.78**	76.08
	301.1075	5.5552	4.7964	1.5656
Balance (625 × 4)	32.63	**93.51**	80.70	91.23
	205.1922	2.2641	2.5184	1.9759

Table 3. Comparison results on NDC datasets with linear classifiers

Dataset	SVM
	Accuracy(%)
	Time(s)
		TSVM
		Accuracy(%)
		Time(s)
			PTSVM
			Accuracy(%)
			Time(s)
				SPTSVM
				Accuracy(%)
				Time(s)

NDC–200	49.74 6.8972	**94.87** 0.6764	96.41 0.4015	94.36 0.2544
NDC–500	50.91 174.5929	93.13 2.2207	93.53 4.0082	**93.94** 1.8806
NDC–700	49.86 494.6866	**96.85** 4.0462	96.00 5.0680	95.86 2.0753
NDC–1000	51.46 1748.1991	96.38 9.4119	96.38 12.0677	**96.48** 3.7086
NDC–2000	50.77 4646.8998	96.75 34.3231	96.80 32.3449	**97.05** 9.4606
NDC–3000	– –	97.50 75.4724	97.06 72.0731	**97.56** 12.8803

Table 4. Comparison results with nonlinear classifiers

Dataset	Parameters
	σ
	c_1, c_2, c_3, c_4
		TSVM
		Accuracy(%)
		Time(s)
			PTSVM
			Accuracy(%)
			Time(s)
				SPTSVM
				Accuracy(%)
				Time(s)

Breast (277 × 9)	0.2 (100,0.1,10,0.1)	**74.9091** 16.0340	63.00 12.6062	72.00 5.7369
Heart (303 × 13)	10 (1,1,1,1)	64.17 8.5809	64.17 12.4591	**79.17** 8.2543
Liver (345 × 6)	0.3 (1,1,1,1)	61.72 12.9630	54.14 16.8381	**67.59** 8.0878
Pima (768 × 8)	0.3 (1,1,1,1)	71.37 75.3000	67.97 103.9722	**76.86** 150.172320

5. Conclusion

In this paper, by means of a smoothing approximation function of plus function, we extend PTSVM to SPTSVM with linear and nonlinear versions and propose an effective fast algorithm for solving SPTSVM by using Newton-Armijo method. In order to verify the effectiveness of SPTSVM, we perform a series of comparative experiments on classification accuracy and running time with SVM, TSVM, and PTSVM on 10 datasets in UCI database and 6 datasets in NDC database. Experiment results show that the proposed SPTSVM is a fast and effective classification method. We know that smoothing technology is a powerful methodology for fast solving various SVM-type classifiers. But the main challenge is to select proper smoothing approximation functions for different classifier modellings. Along this line of thought, we will study two aspects of problems. One is the improvement of modelling and another is the selection of smoothing approximation functions.

References

[1] C. Cortes and V. N. Vapnik, Support vector networks, Machine Learning 20 (1995), 273-297.

[2] O. L. Mangasarian, Nonlinear Programming, Society for Industrial and Applied Mathematics, 1994.

[3] C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining Knowledge Discovery 2(2) (1998), 121-167.

[4] E. Osuna, R. Freund and F. Girosi, Training support vector machines: An application to face detection, IEEE (1997), 130-136.

[5] H. Ince and T. B. Trafalis, Support Vector Machine for Regression and Applications to Financial Forecasting, International Joint Conference on Neural Networks, 2002.

[6] W. S. Noble, Support vector machine applications in computational biology, Kernel Methods in Computational Biology (2004), 77-92.

[7] X. Tang, L. Zhuang, J. Cai and C. Li, Multi-fault classification based on support vector machine trained by chaos particle swarm optimization, Knowledge Based Systems 23(5) (2010), 486-490.

[8] Y. X. Li, Y. H. Shao, L. Jing and N. Y. Deng, An efficient support vector machine approach for identifying proteins-nitrosylation sites, Protein and Peptide Letters 18(6) (2011), 573-587.

[9] K. W. Lau and Q. M. Wu, Local prediction of non-linear time series using support vector regression, Pattern Recognition 41(5) (2008), 1539-1547.

[10] O. L. Mangasarian and E. W. Wild, Multisurface proximal support vector classification via generalized eigenvalues, IEEE Transactions on Pattern Analysis and Machine Intelligence 28(1) (2006), 69-74.

[11] R. Khemchandani Jayadeva and S. Chandra, Twin support vector machines for pattern classification, IEEE Trans. Pattern Anal. Mach. Intell. 29(5) (2007), 905-910.

[12] Y. H. Shao, C. H. Zhang, X. B. Wang and N. Y. Deng, Improvements on twin support vector machines, IEEE Transactions on Neural Networks 22(6) (2011), 962-968.

[13] Q. Ye, C. Zhao, N. Ye and Y. Chen, Multi-weight vector projection support vector machines, Pattern Recognition Letters 31(13) (2010), 2006-2011.

[14] X. Chen, J. Yang, Q. Ye and J. Liang, Recursive projection twin support vector machine via within-class variance minimization, Pattern Recognition 44 (2011), 2643-2655.

[15] Y. J. Lee and O. L. Mangasarian, A smooth support vector machine for classification, Computational Optimization and Applications 20(1) (2001), 5-22.

[16] M. A. Kumar and M. Gopal, Least squares twin support vector machines for pattern classification, Expert Systems with Applications 36(4) (2009), 7535-7543.

[17] M. Arun Kumar and M. Gopal, Application of smoothing technique on twin support vector machine, Pattern Recognition Letters 29 (2008), 1842-1848.

[18] X. Chen, J. Yang, J. Liang and Q. Ye, Smooth twin support vector regression, Neural Computation 21(3) (2012), 505-513.

[19] Z. Wang, Y. Shao and T. Wu, A GA-based model selection for smooth twin parametric-margin support vector machine, Pattern Recognition 46 (2013), 2267-2277.

[20] O. L. Mangasarian, Mathematical programming in neural networks, ORSA Journal on Computing 5(4) (1993), 349-360.

[21] Y. B. Yuan, J. Yan and C. X. Xu, Polynomial smooth support vector machine, Chinese Journal of Computers 28(1) (2005), 9-17.

[22] J. F. Q. Wu, Smooth support vector machine based on piecewise function, Science Direct 20(5) (2013), 122-128.

[23] M. C. Pinar and S. A. Zenios, On smoothing exact penalty functions for convex constrained optimization, SIAM J. Optim. 4 (1994), 486-511.

[24] I. Zhang, A smoothing-out technique for min-max optimization, Math. Program 19 (1980), 61-77.

[25] Y. J. Lee and O. L. Mangasarian, Reduced Support Vector Machines, International Conference on Data Mining, Chicago, 2001.

[26] C. L. Blake and C. J. Merz, UCI Repository for Machine Learning Databases, 1998.

http://www.ics.uci.edu/mlearn/MLRepository.html

[27] D. R. Musicant, NDC: Normally Distributed Clustered Datasets, 1998.

http:// www.cs.wisc.edu/musicant/data/ndc

■