Neural Comput & Applic
DOI 10.1007/s00521-016-2375-8

ORIGINAL ARTICLE

Regularization feature selection projection twin support vector machine via exterior penalty

Ping Yi[1] · Aiguo Song[1] · Jianhui Guo[2] · Ruili Wang[3]

Received: 13 April 2015 / Accepted: 21 May 2016

Abstract In the past years, non-parallel plane classifiers that seek projection direction instead of hyperplane for each class have attracted much attention, such as the multi-weight vector projection support vector machine (MVSVM) and the projection twin support vector machine (PTSVM). Instead of solving two generalized eigenvalue problems in MVSVM, PTSVM solves two related SVM-type problems to obtain the two projection directions by solving two smaller quadratic programming problems, similar to twin support vector machine. In order to suppress input space features, we propose a novel non-parallel classifier to automatically select significant features, called regularization feature selection projection twin support vector machine (RFSPTSVM). In contrast to the PTSVM, we first incorporate a regularization term to ensure the optimization problems are convex, and then replace all the terms with L_1-norm ones. By minimizing an exterior penalty function of the linear programming problem and using a fast generalized Newton algorithm, our RFSPTSVM obtains very sparse solutions. For nonlinear case, the method utilizes minimal number of kernel functions. The experimental results on toy datasets, Myeloma dataset, several UCI benchmark datasets, and NDCC generated datasets show the feasibility and effectiveness of the proposed method.

Keywords Projection twin support vector machine · Multi-weight vector projection support vector machine · Feature selection · Exterior penalty

1 Introduction

Support vector machines (SVMs) [1–3] that are useful classification tools for supervised machine learning have already obtained many significant achievements in practical problems such as drug discovery [4], face recognition [5–9], text categorization [10], time series prediction [11], regression estimation [12, 13], gait recognition [14, 15], and target detection [16, 17].

In the spire of proximal SVM (PSVM) [18], Mangasarian and Wild [19] proposed a generalized eigenvalue proximal support vector machine (GEPSVM), which aims at generating two non-parallel hyperplanes such that each hyperplane is closer to its class and is also as far as possible from the other class. Recently, Jayadeva et al. [20] proposed a TWSVM classifier for binary classification, motivated by GEPSVM. TWSVM generates two non-parallel planes so that each plane is closer to one of two classes and is at least unit distance away from the other one. It is implemented by solving two smaller QPPs rather than a single large QPP, which makes the learning speed of TWSVM 4 times faster than that of the classical SVM. TWSVM was enhanced to least squares TWSVM (LSTWSVM), which was developed by Kumar and Gopal [21]. The solution of LSTWSVM can be obtained by solving two systems of linear equations instead of solving QPPs, and it possesses an extremely fast training speed.

✉ Jianhui Guo
guojianhui@njust.edu.cn

Ping Yi
greenapple8189@163.com

[1] School of Instrument Science and Engineering, Southeast University, Nanjing, China

[2] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China

[3] School of Engineering and Advanced Technology, Massey University, Auckland, New Zealand

Published online: 06 June 2016

Unlike TWSVM which improves GEPSVM by using SVM-type formulation, the MVSVM [22] was proposed to enhance the performance of GEPSVM by solving two optimal weight vector projections instead of two planes so that the samples are as close to samples of the same class while being as far as possible from samples of the other class. For an incoming sample, the class label is determined depending on the distance between the projection of sample and projected class' mean, it will be assigned to the closest class mean. The weight vectors of MVSVM can be found by solving a pair of eigenvalue problems corresponding to the maximum eigenvalue.

Then, based on MVSVM and TWSVM, Chen et al. [23] proposed the PTSVM. Instead of solving two generalized eigenvalue problems in MVSVM, PTSVM solves two related SVM-type problems to obtain the two projection directions. It is implemented by solving two smaller QPPs similar to TWSVM. Furthermore, PTSVM can generate multiple projection directions by using the proposed recursive algorithm. The experimental results in [23] show PTSVM has comparable or better performance when compared with four other state-of-the-art multiple-surface classification (MSC) algorithms such as GEPSVM, TWSVM, LSTWSVM (least squares version of TWSVM), and MVSVM. In order to avoid the possible singularity problem in PTSVM, Shao et al. [24] proposed a projection twin support vector machine with regularization terms, called RPTSVM for short, and the nonlinear form was considered. RPTSVM was obtained by reformulating the primal problem to introduce a regularization term in its objective function.

Feature suppression is very important for a classification problem, especially in very highdimensional spaces, such as drug discovery and classification of gene microarrays (often a primary goal in microarray cancer diagnosis was to identify the genes responsible for classification, rather than class prediction [25]). One way to approach the feature selection problem in classification is to combine a separate feature selection step with the SVM methodology. For example, one could use univariate ranking [26] and recursive feature elimination [27] to select a subset of variables and then fit a L_2-norm SVM by using the selected subset variables. However, these procedures depend on the external feature selection methods.

It is well known that L_1-norm SVM can obtain sparse solutions that result in its classifier being easy to store and faster to compute and suppress features [28–31]. Relevant features can be automatically selected by estimating their coefficients, especially when there are noisy features. Gao et al. [32] proposed a feature selection method called L_1-norm least squares twin support vector machines (NELSTSVM) by applying a Tikhonov regularization term into LSTWSVM learning framework and replacing all the L_2-norm terms in the objective structure of LSTWSVM with the L_1-norm terms. Bai et al. [33] proposed a feature selection method based on TWSVM, called FTSVM, in which a feature selection matrix was introduced. Also, based on the LSTWSVM, a feature selection method for non-parallel plane support vector machine classification (called FLSTSVM) [34] was proposed for feature suppression, in which only a L_1-norm Tikhonov regularization term was added to the objective of the LSTWSVM, while the original L_2-norm term was left unchanged. Recently, Guo et al. [35] proposed a new feature suppression algorithm for binary classification, which is FLSPTSVM (feature selection least squares projection twin support vector machine).

In this paper, we propose a regularization feature selection projection twin support vector machine (RFSPTSVM). Based on the PTSVM objective function, we first incorporate a regularization term to ensure the optimization problems are convex. We then replace all the terms with L_1-norm ones except the slack terms. By minimizing an EP function of the LP problem and using a fast generalized Newton algorithm, RFSPTSVM obtains very sparse solutions as it automatically discards irrelevant features by estimating their coefficients by zero. For nonlinear case, the method utilizes minimal number of kernel functions. The experimental results on toy datasets, several UCI benchmark datasets, and NDC Data generator datasets show the feasibility and effectiveness of the proposed method.

This paper is organized as follows. Section 2 briefly introduces the related works including GEPSCM, MVSVM, and PTSVM. Section 3 proposes our RFSPTSVM and its nonlinear version. Experimental results are described in Sect. 4. Finally, the concluding remarks and future work are given in Sect. 5.

2 Related work

Suppose a binary classification problem in the n-dimensional real space R^n, we are given a set of training data points represented by $X = \{(x_j^{(i)})|i = 1, 2, \quad j = 1, 2, \ldots, m_i.\}$, where $x_j^{(i)} \in R^n$, the jth input belongs to class i, and $m = m_1 + m_2, \quad y_j \in \{+1, -1\}$ are corresponding outputs. We further denote the m_1 inputs of Class $+1$ by matrix A in $R^{m_1 \times n}$ and the m_2 inputs of Class -1 by matrix B in $R^{m_2 \times n}$. The L_2-norm of x is denoted by $\|x\|$, and L_1-norm of x is denoted by $\|x\|_1$. In the following, we review two eigenvalue-type fast support vector machine classifiers GEPSVM [19] and MVSVM [22], and SVM-type fast support vector machine classifiers PTSVM [23].

2.1 GEPSVM

The notion of GEPSVM [19] is to generate two distinct non-parallel hyperplanes to approximate two class sets, respectively. Each hyperplane is closest to one class and as far as the other class. The basic notion of two non-parallel hyperplanes in an n-dimensional input space:

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

where superscript "T" denotes transposition and $(w_i, b_i) \in (R^n \times R)\,(i = 1, 2)$. Then, the criteria of GEPSVM yield the following optimization problem:

$$\text{(GEPSVM1)} \quad \min_{w_1,b_1} \frac{\|Aw_1 + e_1b_1\|^2 + \delta \left\| \begin{pmatrix} w_1 \\ b_1 \end{pmatrix} \right\|^2}{\|Bw_1 + e_2b_1\|^2} \tag{2}$$

and

$$\text{(GEPSVM2)} \quad \min_{w_2,b_2} \frac{\|Bw_2 + e_2b_2\|^2 + \delta \left\| \begin{pmatrix} w_2 \\ b_2 \end{pmatrix} \right\|^2}{\|Aw_2 + e_1b_2\|^2} \tag{3}$$

where e_1 and e_2 are column vectors of ones of appropriate dimensions; $\delta > 0$ is the regularization term to improve stability and performance. Let us give the following definitions:

$$G = [A\, e_1]^{\mathrm{T}}[A\, e_1] + \delta \cdot I, \quad H = [B\, e_2]^{\mathrm{T}}[B\, e_2], \quad z_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} \tag{4}$$

and

$$L = [B\, e_2]^{\mathrm{T}}[B\, e_2] + \delta \cdot I, \quad M = [A\, e_1]^{\mathrm{T}}[A\, e_1], \quad z_2 = \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} \tag{5}$$

The optimization problem in (2) and (3) can be reformulated as:

$$\min_{z_1} \frac{z_1^T G z_1}{z_1^T H z_1} \quad \text{and} \quad \min_{z_1} \frac{z_2^T L z_2}{z_2^T M z_2} \tag{6}$$

The above objective functions in (6) are exactly Rayleigh quotient [28], and the global optimal solution can be obtained by solving the following two related generalized eigenvalue problems (GEPs):

$$Gz_1 = \lambda H z_1 \quad \text{and} \quad Lz_2 = \lambda M z_2 \tag{7}$$

The eigenvectors corresponding to smallest eigenvalues are the solution to (2) and (3). For an unknown test data point $x \in R^n$, the class label is determined as:

$$\text{label}(x) = \arg\min_{i=1,2} \left| w_i^{\mathrm{T}} x + b_i \right| \tag{8}$$

2.2 MVSVM

Similar to GEPSVM, MVSVM [22] is an eigenvalue-type classifier. However, rather than finding the distinct hyperplanes of GEPSVM, MVSVM tries to find a weight vector direction for each class so that each of the two class sets is close to its own class mean and meanwhile the points sharing different labels are separated as far as possible. The optimization criterion and corresponding constraints are defined as follows [22]:

$$\begin{aligned} \text{(MVSVM1)} \ \max_{w_1} & \left(w_1^{\mathrm{T}} \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)} - w_1^{\mathrm{T}} \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)} \right)^2 \\ & - \beta \sum_{i=1}^{m_1} \left(w_1^{\mathrm{T}} x_i^{(1)} - w_1^{\mathrm{T}} \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)} \right)^2 \\ & \text{s.t.} \ \|w_1\|^2 = 1 \end{aligned} \tag{9}$$

and

$$\begin{aligned} \text{(MVSVM2)} \ \max_{w_2} & \left(w_2^{\mathrm{T}} \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)} - w_2^{\mathrm{T}} \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)} \right)^2 \\ & - \beta \sum_{i=1}^{m_2} \left(w_2^{\mathrm{T}} x_i^{(2)} - w_2^{\mathrm{T}} \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)} \right)^2 \\ & \text{s.t.} \ \|w_2\|^2 = 1 \end{aligned} \tag{10}$$

where β is a trade-off regularization constant. The formulation of MVSVM can avoid the singularity problem in GEPSVM by optimizing the difference instead of quotient [36]. To simplify the above formulation, let us give the following definitions:

$$S_1 = \sum_{i=1}^{m_1} \left(x_i^{(1)} - \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)} \right) \left(x_i^{(1)} - \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)} \right)^{\mathrm{T}} \tag{11}$$

$$S_2 = \sum_{i=1}^{m_2} \left(x_i^{(2)} - \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)} \right) \left(x_i^{(2)} - \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)} \right)^{\mathrm{T}} \tag{12}$$

$$S_3 = \left(\frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)} - \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)} \right) \left(\frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)} - \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)} \right)^{\mathrm{T}} \tag{13}$$

Considering the previous definitions, we convert MVSVM1 and MVSVM2 to their equivalent formulations shown as follows:

$$\text{(MVSVM1)} \quad \max_{\|w_1\|=1} w_1^{\mathrm{T}} S_3 w_1 - \beta w_1^{\mathrm{T}} S_1 w_1 \tag{14}$$

and

$$\text{(MVSVM2)} \quad \max_{\|w_2\|=1} w_2^{\mathrm{T}} S_3 w_2 - \beta w_2^{\mathrm{T}} S_2 w_2 \tag{15}$$

The optimal directions w_1 and w_2 are the eigenvectors corresponding to the maximum eigenvalue of matrices $S_3 - \beta S_1$ and $S_3 - \beta S_2$, respectively [22]. For an unknown testing data point $x \in R^n$, its class label is determined as:

$$\text{label}(x) = \arg\min_{i=1,2} \left(\left| w_i^{\mathrm{T}} x - w_i^{\mathrm{T}} \frac{1}{m_i} \sum_{j=1}^{m_i} x_j^{(i)} \right| \right) \tag{16}$$

2.3 PTSVM

The main idea of PTSVM [23] is to seek a projection direction for each class so that the within-class variance of the projected samples of its own class is minimized; meanwhile, the other class projected samples scatter away as far as possible. Then, the formulations of PTSVM are a pair of QPPs:

$$\begin{aligned} &\text{(PTSVM1)} \ \min_{w_1} \frac{1}{2} \sum_{i=1}^{m_1} \left(w_1^{\mathrm{T}} x_i^{(1)} - w_1^{\mathrm{T}} \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)} \right)^2 \\ &+ C_1 \sum_{k=1}^{m_2} \xi_k \ \text{s.t.} \ w_1^{\mathrm{T}} x_k^{(2)} - w_1^{\mathrm{T}} \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)} + \xi_k \geq 1, \\ &\xi_k \geq 0, \quad k = 1, 2, \ldots, m_2 \end{aligned} \tag{17}$$

and

$$\begin{aligned} &\text{(PTSVM2)} \ \min_{w_2} \frac{1}{2} \sum_{i=1}^{m_2} \left(w_2^{\mathrm{T}} x_i^{(2)} - w_2^{\mathrm{T}} \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)} \right)^2 + C_2 \sum_{k=1}^{m_2} \eta_k \\ &\text{s.t.} - \left(w_2^{\mathrm{T}} x_k^{(1)} - w_2^{\mathrm{T}} \frac{1}{m_2} \sum_{j=1}^{m_2} x_j^{(2)} \right) + \eta_k \geq 1, \\ &\eta_k \geq 0, \quad k = 1, 2, \ldots, m_1 \end{aligned} \tag{18}$$

where C_1 and C_2 are both trade-off constants, and ξ_k and η_k are all nonnegative slack variables.

Considering the definitions in Eqs. (10) and (11), PTSVM1 (17) and PTSVM2 (18) can be rewritten as follows:

$$\begin{aligned} &\text{(PTSVM1)} \ \min_{w_1} \frac{1}{2} w_1^{\mathrm{T}} S_1 w_1 + C_1 e_2^{\mathrm{T}} \xi \\ &\text{s.t.} \ B w_1 - \frac{1}{m_1} e_2 e_1^{\mathrm{T}} A w_1 + \xi \geq e_2, \quad \xi \geq 0 \end{aligned} \tag{19}$$

and

$$\begin{aligned} &\text{(PTSVM2)} \ \min_{w_2} \frac{1}{2} w_2^{\mathrm{T}} S_2 w_2 + C_2 e_1^{\mathrm{T}} \eta \\ &\text{s.t.} - \left(A w_2 - \frac{1}{m_2} e_1 e_2^{\mathrm{T}} B w_2 \right) + \eta \geq e_1, \quad \eta \geq 0 \end{aligned} \tag{20}$$

In order to solve the constrained minimization problems in (19) and (20), the Lagrangian function is introduced and the Wolf dual problems of (19) and (20) are as follows:

$$\begin{aligned} &\min_{\alpha} \frac{1}{2} \alpha^{\mathrm{T}} \left(B - \frac{1}{m_1} e_2 e_1^{\mathrm{T}} A \right) S_1^{-1} \left(B^{\mathrm{T}} - \frac{1}{m_1} A^{\mathrm{T}} e_1 e_2^{\mathrm{T}} \right) \alpha - e_2^{\mathrm{T}} \alpha \\ &\text{s.t.} \ 0 \leq \alpha \leq C_1 e_2 \end{aligned} \tag{21}$$

and

$$\begin{aligned} &\min_{\gamma} \frac{1}{2} \gamma^{\mathrm{T}} \left(A - \frac{1}{m_2} e_1 e_2^{\mathrm{T}} B \right) S_2^{-1} \left(A^{\mathrm{T}} - \frac{1}{m_2} B^{\mathrm{T}} e_2 e_1^{\mathrm{T}} \right) \gamma - e_1^{\mathrm{T}} \gamma \\ &\text{s.t.} \ 0 \leq \gamma \leq C_2 e_1 \end{aligned} \tag{22}$$

The projection directions are given by:

$$w_1 = S_1^{-1} \left(B^{\mathrm{T}} - \frac{1}{m_1} A^{\mathrm{T}} e_1 e_2^{\mathrm{T}} \right) \alpha \tag{23}$$

$$w_2 = S_2^{-1} \left(A^{\mathrm{T}} - \frac{1}{m_2} B^{\mathrm{T}} e_2 e_1^{\mathrm{T}} \right) \gamma \tag{24}$$

For testing, the label of an incoming test sample x is determined by using Eq. (16).

3 RFSPTSVM

3.1 Linear RFSPTSVM

In contrast to PTSVM, RFSPTSVM computes the L_1-norm distance between the projection of samples and the projected samples owned class's mean, and then introduces the L_1-norm regularization terms $\|w_1\|_1$ and $\|w_2\|_1$ to the objective functions. Thus, Eqs. (17) and (18) in PTSVM are modified as follows:

$$\begin{aligned} &\text{(RFSPTSVM1)} \ \min_{w_1} \frac{1}{2} \sum_{i=1}^{m_1} \left\| w_1^{\mathrm{T}} x_i^{(1)} - w_1^{\mathrm{T}} \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)} \right\|_1 \\ &+ \varepsilon \|w_1\|_1 + C_1 \sum_{k=1}^{m_2} \xi_k \\ &\text{s.t.} \ w_1^{\mathrm{T}} x_k^{(2)} - w_1^{\mathrm{T}} \frac{1}{m_1} \sum_{j=1}^{m_1} x_j^{(1)} + \xi_k \geq 1, \\ &\xi_k \geq 0, \quad k = 1, \quad 2, \ldots, m_2 \end{aligned} \tag{25}$$

and

$$\begin{aligned}(\text{RFSPTSVM2})\ \min_{w_2}\ & \frac{1}{2}\sum_{i=1}^{m_2}\left\|w_2^T x_i^{(2)} - w_2^T\frac{1}{m_2}\sum_{j=1}^{m_2}x_j^{(2)}\right\|_1 \\ & +\varepsilon\|w_2\|_1+C_2\sum_{k=1}^{m_2}\eta_k \\ & \text{s.t.}\ -(w_2^T x_k^{(1)} - w_2^T\frac{1}{m_2}\sum_{j=1}^{m_2}x_j^{(2)}) + \eta_k \geq 1, \\ & \eta_k \geq 0,\ k=1,2,\ldots,m_1 \end{aligned} \tag{26}$$

Here, ε is a feature suppression parameter.

Let us give the following definitions:

$$G = A - \frac{1}{m_1}e_1 e_1^{\mathrm{T}} A \tag{27}$$

$$H = B - \frac{1}{m_2}e_2 e_2^{T} B \tag{28}$$

Considering the definitions in Eqs. (27) and (28), RFSPTSVM1 (25) and RFSPTSVM2 (26) can be rewritten as follows:

$$\begin{aligned}(\text{RFSPTSVM1})\ \min_{w_1}\ & \frac{1}{2}\|Gw_1\|_1+\varepsilon\|w_1\|_1+C_1 e_2^{\mathrm{T}}\xi \\ & \text{s.t.}\ Bw_1 - \frac{1}{m_1}e_2 e_1^{\mathrm{T}} A w_1 + \xi \geq e_2, \quad \xi \geq 0 \end{aligned} \tag{29}$$

and

$$\begin{aligned}(\text{RFSPTSVM2})\ \min_{w_2}\ & \frac{1}{2}\|Hw_2\|_1+\varepsilon\|w_2\|_1+C_2 e_1^{\mathrm{T}}\eta \\ & \text{s.t.}\ -(Aw_2 - \frac{1}{m_2}e_1 e_2^{\mathrm{T}} B w_2) + \eta \geq e_1, \quad \eta \geq 0 \end{aligned} \tag{30}$$

Due to the L_1-norm term $\|w_i\|_1, \quad i=1,2$, many elements in weight vectors will approximate to zero, and then the remaining nonzero elements in weight vectors will be chosen as selected features. For simplicity of the description, we first discuss Eq. (29) of RFSPTSVM1. We transform Eq. (29) to an explicit LP as in [30, 31] by setting:

$$w_1 = p_1 - q_1, \quad Gw_1 = r_1 - s_1, p_1, q_1, r_1, s_1 > 0 \tag{31}$$

$$M = B - \frac{1}{m_1}e_2 e_1^{\mathrm{T}} A \tag{32}$$

This results in the following LP problem:

$$\begin{aligned} \min & e_1^{\mathrm{T}}(r_1+s_1) + \varepsilon e^{\mathrm{T}}(p_1+q_1) + C_1 e_2^{\mathrm{T}}\xi \\ & \text{s.t.}\ M(p_1-q_1)+\xi \geq e_2, \\ & G(p_1-q_1) = r_1 - s_1 \\ & p_1, q_1, r_1, s_1, \xi \geq 0 \end{aligned} \tag{33}$$

where $e \in R^m$ is a column vector of ones. Equation (33) is a standard LP problem and can be solved with large computational burden. Thus, it has limited application to small-scale classification. Fortunately, it can be converted to an unconstrained minimization problem (UMP) by use of the EP theory in the dual space [31].

Let us deduce the solution of Eq. (33). Constructing the Lagrangian of Eq. (33):

$$\begin{aligned} L(r_1, s_1, p_1, q_1, \xi) = & e_1^{\mathrm{T}}(r_1+s_1) + \varepsilon e^{\mathrm{T}}(p_1+q_1) \\ & + C_1 e_2^{\mathrm{T}}\xi - \alpha_1^{\mathrm{T}}(M(p_1-q_1)+\xi-e_2) \\ & - \alpha_2^{\mathrm{T}}(G(p_1-q_1)-r_1+s_1) - \alpha_3^{\mathrm{T}} r_1 \\ & - \alpha_4^{\mathrm{T}} s_1 - \alpha_5^{\mathrm{T}} p_1 - \alpha_6^{\mathrm{T}} q_1 - \alpha_7^{\mathrm{T}}\xi \end{aligned} \tag{34}$$

where $\alpha_i, i=1,2,\ldots,7$ are the vectors of Lagrange multipliers. The partial derivatives of Lagrangian function (34) on variables $(r_1, s_1, p_1, q_1, \xi)$ are set to be zeroes, and we can get:

$$\begin{aligned} e_1 + \alpha_2 - \alpha_3 &= 0 \\ e_1 - \alpha_2 - \alpha_4 &= 0 \\ \varepsilon e - M^{\mathrm{T}}\alpha_1 - G^{\mathrm{T}}\alpha_2 - \alpha_5 &= 0 \\ \varepsilon e + M^{\mathrm{T}}\alpha_1 + G^{\mathrm{T}}\alpha_2 - \alpha_6 &= 0 \\ C_1 e_2 - \alpha_1 - \alpha_7 &= 0 \end{aligned} \tag{35}$$

On substituting the equality (35) into the Lagrangian function (34), the dual of primal LP (33) becomes:

$$\begin{aligned} \max\ & e_2^{\mathrm{T}}\alpha_1 \\ \text{s.t.}\ & -\varepsilon e \leq M^{\mathrm{T}}\alpha_1 + G^{\mathrm{T}}\alpha_2 \leq \varepsilon e \\ & -e_1 \leq \alpha_2 \leq e_1, \quad 0 \leq \alpha_1 \leq C_1 e_2 \end{aligned} \tag{36}$$

Setting:

$$u = \begin{bmatrix}\alpha_1 \\ \alpha_2\end{bmatrix} \tag{37}$$

Then, the Eq. (36) can be reformulated as:

$$\begin{aligned} \max\ & E^{\mathrm{T}}u \\ \text{s.t.}\ & -\varepsilon e \leq L^{\mathrm{T}}u \leq \varepsilon e \\ & -S_1 \leq u \leq S_2 \end{aligned} \tag{38}$$

where

$$E = \begin{bmatrix}e_2 \\ O_2\end{bmatrix}, \quad L = \begin{bmatrix}M \\ G\end{bmatrix}, \quad S_1 = \begin{bmatrix}O_1 \\ e_1\end{bmatrix}, \quad S_2 = \begin{bmatrix}C_1 e_2 \\ e_1\end{bmatrix} \tag{39}$$

Here, $O_2 \in R^{m_1}, \quad O_1 \in R^{m_2}$ are column vectors of zeros. According to the procedure as in [31, 32], the EP problem of the optimization problem (38) becomes:

$$\begin{aligned} \min_u\ & -\lambda E^{\mathrm{T}}u + \frac{1}{2}\left\|(L^{\mathrm{T}}u - \varepsilon e)_+\right\|^2 + \frac{1}{2}\left\|(-L^{\mathrm{T}}u - \varepsilon e)_+\right\|^2 \\ & + \frac{1}{2}\left\|(u-S_2)_+\right\|^2 + \frac{1}{2}\left\|(-u-S_1)_+\right\|^2 \end{aligned} \tag{40}$$

where $\lambda > 0$, $\|x\|$ denotes the L_2-norm of x, $x_+ = \max(0, x)$. Equation (40) is the EP problem of the primal Eq. (33).

Proposition If the primal LP (33) is solvable, then the dual EP problem (40) is solvable for all $\lambda > 0$. For any $\mu \in (0, \bar{\lambda}]$ for some $\bar{\lambda} > 0$, the solution λ of Eq. (40) generates an exact solution to primal LP (33) as follows:

$$\begin{aligned} p_1 &= \frac{1}{\lambda}(L^{\mathrm{T}}u - \varepsilon e)_+,\ q_1 = \frac{1}{\lambda}(-L^{\mathrm{T}}u - \varepsilon e)_+ \\ r_1 &= \left(\frac{1}{\lambda}(-u - S_1)_+\right)_{m_1},\ s_1 = \left(\frac{1}{\lambda}(u - S_2)_+\right)_{m_1} \\ \xi &= \left(\frac{1}{\lambda}(u - S_2)_+\right)_{m_2} \end{aligned} \tag{41}$$

where $(X)_{m_1}$ denotes the before m_1 rows of matrix X, and $(X)_{m_2}$ denotes the last m_2 ($m_1 = m - m_2$) rows of matrix X. Then, we can get the projection weight vector of RFSPTSVM1 as:

$$w_1 = \frac{1}{\lambda}(L^{\mathrm{T}}u - \varepsilon e)_+ - \frac{1}{\lambda}(-L^{\mathrm{T}}u - \varepsilon e)_+ \tag{42}$$

The theoretical analysis of this proposition's proof procedure is similar to [31, 32].

Proof Letting:

$$\begin{aligned} v_1 &= \left(L^{\mathrm{T}}u - \varepsilon e\right)_+,\quad v_2 = \left(-L^{\mathrm{T}}u - \varepsilon e\right)_+, \\ v_3 &= (u - S_2)_+,\quad v_4 = (-u - S_1)_+ \end{aligned} \tag{43}$$

Substituting Eq. (43) into Eq. (40), we can get:

$$\begin{aligned} \min_{u,v_1,v_2,v_3,v_4}\ & -\lambda E^{\mathrm{T}}u + \frac{1}{2}\|v_1\|^2 + \frac{1}{2}\|v_2\|^2 + \frac{1}{2}\|v_3\|^2 + \frac{1}{2}\|v_4\|^2 \\ \text{s.t.}\ & -L^{\mathrm{T}}u + \varepsilon e + v_1 \geq 0,\quad L^{\mathrm{T}}u + \varepsilon e + v_2 \geq 0 \\ & -u - S_2 + v_3 \geq 0,\quad u + S_1 + v_4 \geq 0 \end{aligned} \tag{44}$$

Constructing the Lagrange function of Eq. (40):

$$\begin{aligned} L(u, v_1, v_2, v_3, v_4) = & -\lambda E^{\mathrm{T}}u + \frac{1}{2}\|v_1\|^2 + \frac{1}{2}\|v_2\|^2 + \frac{1}{2}\|v_3\|^2 \\ & + \frac{1}{2}\|v_4\|^2 - \alpha_1^{\mathrm{T}}\left(-L^{\mathrm{T}}u + \varepsilon e + v_1\right) \\ & - \alpha_2^{\mathrm{T}}\left(L^{\mathrm{T}}u + \varepsilon e + v_2\right) \\ & - \alpha_3^{\mathrm{T}}(-u - S_2 + v_3) - \alpha_4^{\mathrm{T}}(u + S_1 + v_4) \end{aligned} \tag{45}$$

Getting the K.K.T conditions:

$$\begin{aligned} & -\lambda E + L\alpha_1 - L\alpha_2 + \alpha_3 - \alpha_4 = 0,\ \alpha_1 = v_1, \\ & \alpha_2 = v_2,\ \alpha_3 = v_3,\ \alpha_4 = v_4 \end{aligned} \tag{46}$$

Then, we can get the minimization Wolf dual [37] of Eq. (44):

$$\begin{aligned} \min_{u,v_1,v_2,v_3,v_4}\ & \frac{1}{2}\|v_1\|^2 + \frac{1}{2}\|v_2\|^2 + \frac{1}{2}\|v_3\|^2 + \frac{1}{2}\|v_4\|^2 \\ & + \varepsilon e^{\mathrm{T}}\alpha_1 + \varepsilon e^{\mathrm{T}}\alpha_2 + S_2^{\mathrm{T}}\alpha_3 + S_1^{\mathrm{T}}\alpha_4 \\ \text{s.t.}\ & -\lambda E + L\alpha_1 - L\alpha_2 + \alpha_3 - \alpha_4 = 0, \\ & \alpha_1 = v_1 \geq 0,\ \alpha_2 = v_2 \geq 0, \\ & \alpha_3 = v_3 \geq 0,\ \alpha_4 = v_4 \geq 0 \end{aligned} \tag{47}$$

Setting:

$$\begin{aligned} \gamma_1 &= ((u - S_2)_+)_{m_2},\quad \gamma_2 = ((u - S_2)_+)_{m_1}, \\ \gamma_3 &= ((-u - S_1)_+)_{m_2},\quad \gamma_4 = ((-u - S_1)_+)_{m_1} \end{aligned} \tag{48}$$

Then, we get:

$$\begin{aligned} \alpha_3 = v_3 &= (u - S_2)_+ = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}, \\ \alpha_4 = v_4 &= (-u - S_1)_+ = \begin{bmatrix} \gamma_3 \\ \gamma_4 \end{bmatrix} \end{aligned} \tag{49}$$

Using Eqs. (49) and (39), the inequality constraint of minimization Wolf dual problem (47) can be rewritten as two equalities (both sides of the equation are divided by λ):

$$\begin{aligned} & -e_2 + \frac{1}{\lambda}Mv_1 - \frac{1}{\lambda}Mv_2 + \frac{1}{\lambda}\gamma_1 = \frac{1}{\lambda}\gamma_3 \geq 0 \\ & \frac{1}{\lambda}Gv_1 - \frac{1}{\lambda}Gv_2 + \frac{1}{\lambda}\gamma_2 - \frac{1}{\lambda}\gamma_4 = 0 \end{aligned} \tag{50}$$

The Wolf dual problem (47) can be reformulated as follows:

$$\begin{aligned} \min_{u,v_1,v_2,v_3,v_4}\ & \frac{1}{2}\|v_1\|^2 + \frac{1}{2}\|v_2\|^2 + \frac{1}{2}\gamma_1^2 + \frac{1}{2}\gamma_2^2 + \frac{1}{2}\gamma_3^2 + \frac{1}{2}\gamma_4^2 \\ & + \varepsilon e^{\mathrm{T}}v_1 + \varepsilon e^{\mathrm{T}}v_2 + C_1 e_2^{\mathrm{T}}\gamma_1 + e_1^{\mathrm{T}}\gamma_2 + e_1^{\mathrm{T}}\gamma_4 \\ \text{s.t.}\ & \frac{1}{\lambda}Mv_1 - \frac{1}{\lambda}Mv_2 + \frac{1}{\lambda}\gamma_1 \geq e_2,\quad \frac{1}{\lambda}Gv_1 - \frac{1}{\lambda}Gv_2 = \frac{1}{\lambda}\gamma_4 - \frac{1}{\lambda}\gamma_2 \\ & v_1, v_2, \gamma_1, \gamma_2, \gamma_3, \gamma_4 \geq 0 \end{aligned} \tag{51}$$

Letting:

$$\begin{aligned} & p_1 = \frac{1}{\lambda}v_1,\quad q_1 = \frac{1}{\lambda}v_2,\quad \xi = \frac{1}{\lambda}\gamma_1,\quad r_1 = \frac{1}{\lambda}\gamma_4, \\ & s_1 = \frac{1}{\lambda}\gamma_2 \end{aligned} \tag{52}$$

Using Eq. (52) and the first equality in Eq. (50), the Wolf dual problem (47) can be shown as:

$$\begin{aligned} \min_{p_1,q_1,\xi,r_1,s_1}\ & \varepsilon e^{\mathrm{T}}(p_1 + q_1) + e_1^{\mathrm{T}}(r_1 + s_1) + C_1 e_2^{\mathrm{T}}\xi \\ & + \frac{\lambda}{2}\left(\|p_1\|^2 + \|q_1\|^2 + \|r_1\|^2 + \|s_1\|^2 + \|\xi\|^2 + \|M(p_1 - q_1) + \xi - e_2\|^2\right) \\ \text{s.t.}\ & M(p_1 - q_1) + \xi \geq e_2,\quad G(p_1 - q_1) = r_1 - s_1 \\ & p_1, q_1, \xi, r_1, s_1 \geq 0 \end{aligned} \tag{53}$$

As the LP (33) is feasible, the convex quadratic program (53) is feasible. It can be solved for any $\lambda > 0$ as its objective function is bounded from below due to it being a strongly convex quadratic function. It is clear that Eq. (53) has an additional perturbation term

$$(\lambda/2)(\|p_1\|^2+\|q_1\|^2+\|r_1\|^2+\|s_1\|^2+\|\xi\|^2 + \|M(p_1-q_1)+\xi-e_2\|^2)$$

when compared with Eq. (33). By the perturbation theory of linear programs [38], the solutions of Eq. (53) on variables p_1, q_1, ξ, r_1, s_1 are the solutions of the primal problem (33). Naturally, we can get the solutions of the original LP (29).

In an entirely similar way, the formulation (30) (RFSPTSVM2) can be shown as:

$$\begin{aligned} &\min e_2^{\mathrm{T}}(r_2+s_2)+\varepsilon e^{\mathrm{T}}(p_2+q_2)+C_2 e_1^{\mathrm{T}}\eta \\ &\text{s.t.} -Q(p_2-q_2)+\eta \geq e_1, \\ &\quad H(p_2-q_2)=r_2-s_2 \\ &\quad p_2,q_2,r_2,s_2,\eta \geq 0 \end{aligned} \tag{54}$$

where

$$w_2 = p_2 - q_2, \quad Hw_2 = r_2 - s_2, \quad p_2, q_2, r_2, s_2 > 0 \tag{55}$$

$$Q = A - \frac{1}{m_2} e_1 e_2^{\mathrm{T}} B \tag{56}$$

The Wolf dual of Eq. (54) can be represented as:

$$\begin{aligned} &\max \quad F^{\mathrm{T}}v \\ &\text{s.t.} -\varepsilon e \leq K^{\mathrm{T}}v \leq \varepsilon e \\ &\quad -R_1 \leq v \leq R_2 \end{aligned} \tag{57}$$

where

$$v = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}, \quad F = \begin{bmatrix} e_1 \\ O_1 \end{bmatrix}, \quad K = \begin{bmatrix} Q \\ -H \end{bmatrix}, \quad R_1 = \begin{bmatrix} O_2 \\ e_2 \end{bmatrix}, \quad R_2 = \begin{bmatrix} C_2 e_1 \\ e_2 \end{bmatrix} O_2 \in R^{m_1}, \quad O_1 \in R^{m_2} \tag{58}$$

The EP of dual problem (54) becomes:

$$\min_v -\lambda F^{\mathrm{T}}v + \frac{1}{2}\left\|(K^{\mathrm{T}}v-\varepsilon e)_+\right\|^2 + \frac{1}{2}\left\|(-K^{\mathrm{T}}v-\varepsilon e)_+\right\|^2 + \frac{1}{2}\left\|(v-R_2)_+\right\|^2 + \frac{1}{2}\left\|(-v-R_1)_+\right\|^2 \tag{59}$$

According to the proposition, we can get the solution of Eq. (54) in the same way:

$$\begin{aligned} &p_2 = \frac{1}{\lambda}\left(-K^{\mathrm{T}}v-\varepsilon e\right)_+, \quad q_2 = \frac{1}{\lambda}\left(K^{\mathrm{T}}v-\varepsilon e\right)_+ \\ &r_2 = \left(\frac{1}{\lambda}(-v-R_1)_+\right)_{m_2}, \quad s_2 = \left(\frac{1}{\lambda}(v-R_2)_+\right)_{m_2} \\ &\eta = \left(\frac{1}{\lambda}(v-R_2)_+\right)_{m_1} \end{aligned} \tag{60}$$

The projection weight vector of RFSPTSVM2 can be obtained as:

$$w_2 = \frac{1}{\lambda}(-K^{\mathrm{T}}v-\varepsilon e)_+ - \frac{1}{\lambda}(K^{\mathrm{T}}v-\varepsilon e)_+ \tag{61}$$

So far, the projection weight vectors w_1 and w_2 are the functions on variables u and v, respectively. If the variables u and v are known, the two projection weight vectors can be determined. According to [31], the unconstrained minimization problems (40) and (59) can be solved by the generalized Newton algorithm. The procedure is stated as follows similar to [34, 35]:

Algorithm 1: Generalized Newton Algorithm for unconstrained minimization problem (40)

(1). Let $f(u) = -\lambda E^T u + \frac{1}{2}\left\|(L^T u - \varepsilon e)_+\right\|^2 + \frac{1}{2}\left\|(-L^T u - \varepsilon e)_+\right\|^2 + \frac{1}{2}\left\|(u-S_2)_+\right\|^2 + \frac{1}{2}\left\|(-u-S_1)_+\right\|^2$;

The gradient as: $\nabla f(u) = -\lambda E + L(L^T u - \varepsilon e)_+ - L(-L^T u - \varepsilon e)_+ + (u-S_2)_+ - (-u-S_1)_+$;

The Hessian as: $\partial^2 f(u) = L(diag((L^T u - \varepsilon e)_* + (-L^T u - \varepsilon e)_+)L^T + diag((u-S_2)_* + (-u-S_1)_*)$ .Here $(z)_*$ denotes the gradient of $(z)_+$,if $z_i > 0, (z_*)_i = 1$; if $z_i < 0, (z_*)_i = 0$; if $z_i = 0, (z_*)_i \in [0,1]$;

(2). Set the parameter values λ , tolerance tol , and $i\max$ (typically: $\lambda \in [10^{-6}, 10^{-4}]$, $tol = 10^{-3}$, $i\max = 50$, while ε and C_1 are set by a tuning procedure);

(3). Start with any $u^0 \in R^m$, For $i = 1,2,\cdots$;

(4). $u^{i+1} = u^i - \mu_i(\partial^2 f(u^i) + \delta I)^{-1}\nabla f(u^i) = u^i + \mu_i d^i$, here the Armijo step size $\mu_i = \max\{1, \frac{1}{2}, \frac{1}{4}, \cdots\}$, such that $f(u^i) - f(u^i + \mu_i d^i) \geq -\frac{\mu_i}{4}\nabla f(u^i)^T d^i$ and $d^i = -(\partial^2 f(u^i) + \delta I)^{-1}\nabla f(u^i)$ is the modified Newton direction;

(5). Stop if $\left\|u^{i+1} - u^i\right\| \leq tol$ or $i = i\max$ .Else set $i = i+1$ and go to (4);

(6). Get the projection direction w_1 using equation (42) with $u = u^i$.

The computational complexity of $(\partial^2 f(u^i)+\delta I)^{-1}$ is $O(m^3)$. This makes it unacceptable to apply this algorithm to large-scale classification data ($m \geq n$). Also, it needs $O(m^2)$ storage space for $(\partial^2 f(u^i)$. The Sherman–Morrison–Woodbury (SMW) equality is utilized to reduce the computation and storage cost. Setting:

$$\begin{aligned} U^2 &= \mathrm{diag}((L^{\mathrm{T}}u-\varepsilon e)_* + (-L^{\mathrm{T}}u-\varepsilon e)_*), \\ V &= \mathrm{diag}((u-S_2)_* + (-u-S_1)_*) + \delta I, \quad P = UV \end{aligned} \tag{62}$$

Thus:

$$\begin{aligned} (\partial^2 f(u^i)+\delta I)^{-1} &= (LUU^{\mathrm{T}}L^{\mathrm{T}}+V)^{-1} \\ &= V^{-1}(I-P(I+P^{\mathrm{T}}U^{-1}P)^{-1}P^{\mathrm{T}}V^{-1}) \end{aligned} \tag{63}$$

where U and V are diagonal matrixes, $(I+P^{\mathrm{T}}U^{-1}P)^{-1} \in R^{n\times n}$. Thus, the computational complexity of Eq. (63) is $O(n^2)$.

The generalized Newton algorithm for an unconstrained minimization problem (59) is entirely similar. Thus, we can get the projection direction w_2 using Eq. (61).

The class of an unknown data point $x \in R^n$ is determined depending on the distance between the projection of x and the projected class means. The class label of unknown data point is expressed as:

$$\mathrm{label}(x) = \arg\min_{i=1,2}\left(\left|w_i^{\mathrm{T}}x - w_i^{\mathrm{T}}\frac{1}{m_i}\sum_{j=1}^{m_i}x_j^{(i)}\right|\right) \tag{64}$$

3.2 Nonlinear RFSPTSVM

In this subsection, we turn now to nonlinear kernel classifiers. In the nonlinear kernel case, the objective functions of the optimization problem for RFSPTSVM can be reformulated as:

$$\begin{aligned} (\mathrm{KRFSPTSVM1}) \quad & \min_{u_1}\frac{1}{2}\left\|(K(A,C^{\mathrm{T}})-\frac{1}{m_1}e_1e_1^{\mathrm{T}}K(A,C^{\mathrm{T}}))u_1\right\|_1 \\ & +\varepsilon\|u_1\|_1+C_1e_2^{\mathrm{T}}\xi \\ & \quad \text{s.t.} \quad K(B,C^{\mathrm{T}})u_1-\frac{1}{m_1}e_2e_1^{\mathrm{T}}K(A,C^{\mathrm{T}})u_1 \\ & +\xi \geq e_2, \quad \xi \geq 0 \end{aligned} \tag{65}$$

$$\begin{aligned} (\mathrm{KRFSPTSVM2}) \quad & \min_{u_2}\frac{1}{2}\left\|(K(B,C^{\mathrm{T}})-\frac{1}{m_2}e_2e_2^{\mathrm{T}}K(B,C^{\mathrm{T}}))u_2\right\|_1 \\ & +\varepsilon\|u_2\|_1+C_2e_1^{\mathrm{T}}\eta \\ & \quad \text{s.t.} -(K(A,C^{\mathrm{T}})u_2-\frac{1}{m_2}e_1e_2^{\mathrm{T}}K(B,C^{\mathrm{T}})u_2) \\ & +\eta \geq e_1, \eta \geq 0 \end{aligned} \tag{66}$$

where $C = \begin{bmatrix} A \\ B \end{bmatrix}$, and K is an appropriately chosen kernel.

Setting:

$$g = K(A,C^{\mathrm{T}}) - \frac{1}{m_1}e_1e_1^{\mathrm{T}}K(A,C^{\mathrm{T}}) \tag{67}$$

$$h = K(B,C^{\mathrm{T}}) - \frac{1}{m_1}e_2e_1^{\mathrm{T}}K(A,C^{\mathrm{T}}) \tag{68}$$

$$a = K(B,C^{\mathrm{T}}) - \frac{1}{m_2}e_2e_2^{\mathrm{T}}K(B,C^{\mathrm{T}}) \tag{69}$$

$$b = K(A,C^{\mathrm{T}}) - \frac{1}{m_2}e_1e_2^{\mathrm{T}}K(B,C^{\mathrm{T}}) \tag{70}$$

In the same way as in the linear case, the EP problem of the optimization problem (65) becomes:

$$\begin{aligned} \min_u \; & -\lambda E^{\mathrm{T}}u+\frac{1}{2}\left\|(L^{\mathrm{T}}u-\varepsilon e)_+\right\|^2+\frac{1}{2}\left\|(-L^{\mathrm{T}}u-\varepsilon e)_+\right\|^2 \\ & +\frac{1}{2}\left\|(u-S_2)_+\right\|^2+\frac{1}{2}\left\|(-u-S_1)_+\right\|^2 \end{aligned} \tag{71}$$

where

$$E = \begin{bmatrix} e_2 \\ O_2 \end{bmatrix}, \quad L = \begin{bmatrix} h \\ g \end{bmatrix}, \quad S_1 = \begin{bmatrix} O_1 \\ e_1 \end{bmatrix}, \quad S_2 = \begin{bmatrix} C_1e_2 \\ e_1 \end{bmatrix} \tag{72}$$

Here, $O_2 \in R^{m_1}$, $O_1 \in R^{m_2}$ are column vectors of zeros.

The EP problem of the optimization problem (66) becomes:

$$\begin{aligned} \min_v \; & -\lambda F^{\mathrm{T}}v+\frac{1}{2}\left\|(K^{\mathrm{T}}v-\varepsilon e)_+\right\|^2+\frac{1}{2}\left\|(-K^{\mathrm{T}}v-\varepsilon e)_+\right\|^2 \\ & +\frac{1}{2}\left\|(v-R_2)_+\right\|^2+\frac{1}{2}\left\|(-v-R_1)_+\right\|^2 \end{aligned} \tag{73}$$

where

$$\begin{aligned} & F = \begin{bmatrix} e_1 \\ O_1 \end{bmatrix}, \quad K = \begin{bmatrix} b \\ -a \end{bmatrix}, \quad R_1 = \begin{bmatrix} O_2 \\ e_2 \end{bmatrix}, \\ & R_2 = \begin{bmatrix} C_2e_1 \\ e_2 \end{bmatrix}, \quad O_2 \in R^{m_1}, \quad O_1 \in R^{m_2} \end{aligned} \tag{74}$$

In the same way as in the linear case, the u and v of Eqs. (71) and (73) can be solved by the generalized Newton method. When u and v are known,

$$w_1 = \frac{1}{\lambda}(L^{\mathrm{T}}u-\varepsilon e)_+ - \frac{1}{\lambda}(-L^{\mathrm{T}}u-\varepsilon e)_+ \tag{75}$$

$$w_2 = \frac{1}{\lambda}(-K^{\mathrm{T}}v-\varepsilon e)_+ - \frac{1}{\lambda}(K^{\mathrm{T}}v-\varepsilon e)_+ \tag{76}$$

The label of an incoming data point x is determined in a manner similar to the linear case, which can be expressed as:

$$\text{label}(x) = \arg\min_{i=1,2}\left(\left|K(x, C^{\mathrm{T}})u_i - \frac{1}{m_i}e_i K(X_i, C^{\mathrm{T}})u_i\right|\right) \tag{77}$$

where $X_1 = A$ and $X_2 = B$.

4 Experimental results

The testing was carried out on a PC with an Intel(R) Core(TM) i5 M480 processor (2.67 GHz) and 2 GB of memory running Windows 7 utilizing MATLAB 7.10.0 R2012a. In the experiments, for each SVM algorithm for nonlinear kernels, a Gaussian kernel $K(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2)$ was used. In order to demonstrate the performance of our proposed algorithm, we test our algorithm using publicly available datasets: the UCI Repository [39], MNIST datasets [42], one classical artificial dataset "cross plane" [19], and the high-dimensional multiple myeloma dataset [40], as well as the datasets generated by Thompson's NDCC Data Generator [41].

We compare the proposed algorithms with three state-of-the-art non-parallel multiple-surface classification methods, including GEPSVM [19], TWSVM [20], and PTSVM [23], and two fast-feature selection methods for original standard SVM (i.e., NLPSVM [30] and LPNewton [31]). The dual QPPs occurring in TWSVM and PTSVM are solved by the Mosek optimization toolbox http://www.mosek.com. The classification accuracy of each algorithm is the standard tenfold cross-validation technique for all the datasets. Also, we can get the means and standard deviations of the results.

Furthermore, tuning on 10 % of the training set was used to search the unknown parameters from the set $\{2^i | i = -7, -6, \ldots, 6, 7\}$ except the parameters δ which tuned from the set $\{10^i | i = -3 - 2, \ldots, 2, 3\}$. For NLPSVM, LPNewton, and our RFSPTSVM, the components of $w_i, i = 1, 2$ less than $1e{-}8$ in absolute are discarded. The remaining features are used to classify the unknown samples.

RFSPTSVM's feature suppression performance is reported by the formula $(F_1 + F_2)/2$, where F_1 and F_2 (F_1, $F_2 \leq n$, where n is the dimensions of the input datasets) are the number of features remaining in RFSPTSVM1 and RFSPTSVM2 for the linear case, respectively. For the nonlinear case, F_1 and F_2 are the number of kernel functions remaining in KRFSPTSVM1 and KRFSPTSVM2, respectively.

4.1 Toy example

A simple "cross planes" example is a generalization of the XOR example. We use the artificial datasets "cross planes" to visually illustrate the effectiveness of the proposed method on XOR datasets. Figure 1 illustrates their distributions and classification results. The average results of GEPSVM, TWSVM, PTSVM, NLPSVM, LPNewton, and our RFSPTSVM on the "cross plane" are reported in Table 1. From Table 1, one can observe that RFSPTSVM obtained the best classification correctness of 99.47 %, which is the same as the GEPSVM. This indicates RFSPTSVM is capable of dealing with XOR problems. NLPSVM and LPNewton are feature selection methods based on the original standard SVM, which do not have a good performance as the XOR example is a

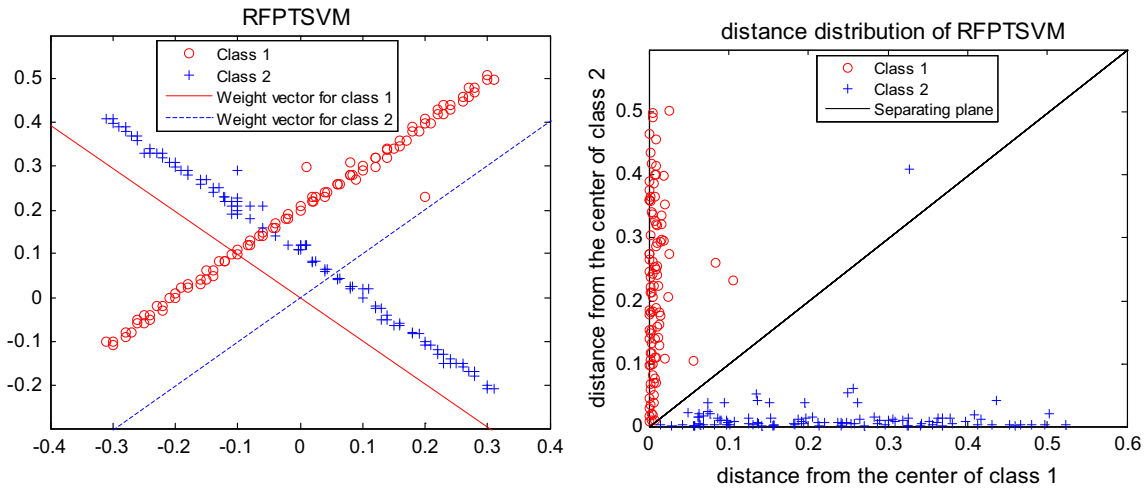

Fig. 1 "Cross plane" dataset learned by RFSPTSVM

Table 1 Classification accuracy (%) on cross plane

Datasets	GEPSVM Train (%) Test ± std (%)	TWSVM Train (%) Test ± std (%)	PTSVM Train (%) Test ± std (%)	NLPSVM Train (%) Test ± std (%)	LPNewton Train (%) Test ± std (%)	RFSPTSVM Train (%) Test ± std (%)
Cross plane 196 × 2	**99.49**	98.07	99.09	64.06	67.18	**99.49**
	99.47 ± 1.57	97.97 ± 3.34	96.92 ± 3.36	60.16 ± 7.19	62.21 ± 9.64	**99.47 ± 1.57**

Bold values are of the highest class accuracy

Table 2 Comparison on multiple myeloma dataset with linear kernel

Datasets	GEPSVM Test Acc ± std (%) Features ± std Train Tim (s)	TWSVM Test Acc ± std (%) Features ± std Train Tim (s)	PTSVM Test Acc ± std (%) Features ± std Train Tim (s)	NLPSVM Test Acc ± std (%) Features ± std Train Tim (s)	LPNewton Test Acc ± std (%) Features ± std Train Tim (s)	RFSPTSVM Test Acc ± std (%) Features ± std Train Tim (s)
Myeloma	*	*	*	**97.27 ± 5.82**	94.18 ± 11.94	96.27 ± 4.57
105 × 28,032	*	*	*	4.50 ± 1.12	6.60 ± 9.90	9.50 ± 1.95
	*	*	*	5.5521	18.4799	11.1101

* The experiments were stopped due to that algorithms run out of memory. Time is for onefold in seconds. Means and standard deviations of test correctness and selected features are over tenfold

Bold value is the highest class accuracy

difficult test case for original standard linear SVM classifiers.

4.2 Multiple Myeloma dataset

Feature selection is a significant advantage of our RFSPTSVM algorithm. Therefore, the performance of our RFSPTSVM algorithm on the multiple myeloma dataset is first compared with GEPSVM, TWSVM, PTSVM, NLPSVM, and LPNewton. Multiple myeloma is cancer of the plasma cell. The data consist of 105 samples, and each sample is in $R^{28,032}$. The detailed information about the dataset can be found in Ref. [40].

We fixed the feature suppression parameter and penalty parameters of all algorithms to be 1 (i.e., $\varepsilon = 1, \quad C = 1, \quad C_1 = 1, \quad C_2 = 1$) for saving the time of tuning parameters due to high dimensions. Table 2 shows that our proposed algorithm achieves good classification accuracy and a remarkable result of suppressing features especially for datasets in a very high-dimensional input space. The feature suppression performance of RFSPTSVM is comparable with NLPSVM and LPNewton.

4.3 UCI datasets and MNIST datasets

We implemented the six algorithms GEPSVM, TWSVM, PTSVM, NLPSVM, LPNewton, and our RFSPTSVM on multiple selected benchmark UCI datasets [39] and MNIST datasets [42] to further investigate the performance of RFSPTSVM. We applied paired t tests comparison of RFSPTSVM versus GEPSVM, TWSVM, PTSVM, NLPSVM, and LPNewton. The significance level of paired t tests was set to 0.05. In other words, there is greater difference between two classifiers' accuracy value distributions when p value below 0.05.

Table 3 shows the results comparing the linear classifiers. The numbers of test cases in which p values are below 0.05 by comparison RFSPTSVM versus GEPSVM is 7 out of 14 cases, versus TWSVM is 3 out of 14 cases, versus PTSVM and LPNewton is 1 out of 14 cases, versus NLPSVM is 0 out of 14 cases, respectively. The results indicated that RFSPTSVM's generalization capability is comparable to that of TWSVM and PTSVM, but better than that of GEPSVM. Specifically, RFSPTSVM gains better accuracy on 5 of 14 datasets such as Sonar, Votes, Heart-statlog, Tic-tac-toe, and MNIST 1 versus 2. Furthermore, RFSPTSVM obtains effective feature suppressing ability which is similar with NLPSVM and LPNewton in the linear cases.

Table 4 reports the comparison for the nonlinear case with Gaussian kernel. The ratios in Table 4 denote the selected kernel functions rates of RFSPTSVM, LPNewton, and NLPSVM. Table 4 reveals that RFSPTSVM has a comparable generalization performance to both TWSVM and PTSVM, but with fewer kernel functions. Specifically, RFSPTSVM gains comparable or better accuracy on 4 of 12 datasets, such as Ionosphere, Votes, Spect, and Tic-tac-toe, while in the nonlinear case, our method can perform feature suppression in a high-dimensional space. With a

Table 3 Tenfold testing classification accuracy comparison of linear classifiers

Datasets	GEPSVM Acc ± std (%) *p* value Time (s)	TWSVM Acc ± std (%) *p* value Time (s)	PTSVM Acc ± std (%) *p* value Time (s)	NLPSVM Acc ± std (%), Feat. *p* value Time (s)	LPNewton Acc ± std (%), Feat. *p* value Time (s)	RFSPTSVM Acc ± std (%), Feat. *p* value Time (s)
UCI Ionosphere 351 × 34	76.93 ± 6.54	**92.87 ± 4.47**	88.60 ± 4.96	86.02 ± 6.07, 11.4	87.47 ± 4.44, 12.4	86.03. ± 4.34, 22.9
	0.0064	0.0003	0.1351	0.996	0.2400	
	0.0109	0.134	0.131	0.083	0.0798	0.0634
UCI BUPA Liver 345 × 6	59.13 ± 5.49	**69.51 ± 7.48**	65.77 ± 7.38	69.31 ± 7.43, 6.0	68.71 ± 5.69, 5.0	67.01 ± 7.27, 4.9
	0.0156	0.4987	0.7319	0.3458	0.3816	
	0.0068	0.1238	0.1154	0.0557	0.0540	0.0749
UCI Cleveland 296 × 13	73.97 ± 5.31	81.00 ± 7.20	**82.78 ± 6.73**	80.39 ± 4.76, 9.9	78.99 ± 5.88, 12.9	81.39 ± 8.86, 10.2
	0.0168	0.861	0.7382	0.6243	0.3504	
	0.0071	0.1269	0.1054	0.0566	0.0442	0.1008
UCI Housing 506 × 13	73.14 ± 5.42	85.38 ± 4.84	85.36 ± 5.25	86.56 ± 4.19, 10.8	**86.75 ± 4.82**, 12.2	85.37 ± 3.58, 11.2
	0.0006	0.9967	0.9964	0.4042	0.4667	
	0.0074	0.2025	0.1827	0.0652	0.0654	0.0939
UCI Pima 768 × 8	74.74 ± 4.05	**77.87 ± 3.93**	76.04 ± 3.26	77.48 ± 3.88, 7.9	74.23 ± 3.35, 2.2	75.39 ± 6.22, 7.9
	0.7105	0.0581	0.7754	0.1638	0.5738	
	0.0069	0.4203	0.3984	0.0711	0.0303	0.0993
UCI Sonar 208 × 60	74.12 ± 8.13	70.24 ± 8.58	75.57 ± 12.46	74.62 ± 9.61, 17.9	74.55 ± 13.5, 13.2	**76.43 ± 6.95**, 13.4
	0.4697	0.0456	0.8569	0.5710	0.6965	
	0.0150	0.1011	0.0951	0.0902	0.0368	0.1247
UCI Votes 435 × 16	94.01 ± 2.38	94.24 ± 3.45	94.73 ± 3.93	94.02 ± 1.55, 4.5	94.25 ± 1.86, 2.7	**94.94 ± 2.24**, 11.0
	0.3420	0.5145	0.8406	0.3097	0.4953	
	0.0070	0.1871	0.1675	0.0368	0.0304	0.0564
UCI Monk3 554 × 6	80.14 ± 3.76	87.19 ± 4.97	**87.34 ± 4.56**	80.13 ± 2.956, 2.9	80.13 ± 2.95, 2.1	81.36 ± 4.31, 3.0
	0.5775	0.0071	0.0169	0.4264	0.4264	
	0.0081	0.2122	0.1892	0.0553	0.0371	0.0640
UCI Spect 267 × 44	74.17 ± 8.70	75.26 ± 8.49	78.63 ± 7.91	79.81 ± 5.46, 8.6	**80.20 ± 6.73**, 3.0	79.44 ± 8.09, 15.5
	0.3839	0.3539	0.9523	0.4945	0.4238	
	0.0122	0.1374	0.1336	0.0352	0.0339	0.0471
UCI Heart-statlog 270 × 13	72.59 ± 8.14	82.96 ± 7.62	84.41 ± 5.09	83.70 ± 7.80, 11.3	69.26 ± 10.74, 2.7	**84.44 ± 5.44,** 12.75
	0.0095	0.5338	0.8402	0.7640	0.0040	
	0.0069	0.1182	0.1063	0.0557	0.0291	0.0501
UCI Tic-tac-toe 958 × 9	65.87 ± 3.17	65.45 ± 4.45	65.24 ± 8.00	65.76 ± 4.66, 4.3	67.43 ± 3.79, 8.0	**67.96 ± 6.68,** 4.8
	0.4084	0.4029	0.4792	0.4775	0.8606	
	0.0065	0.6014	0.6011	0.0731	0.0286	0.0732
UCI Wpbc 194 × 33	75.42 ± 12.84	70.58 ± 10.92	72.08 ± 9.19	**79.53 ± 11.39, 6.4**	76.34 ± 16.11, 8.5	75.34 ± 10.28, 11.0
	0.9721	0.1557	0.5068	0.1039	0.7902	
	0.0074	0.1095	0.0934	0.0647	0.0665	0.0594
MNIST 1 versus 2 820 × 784	90.39 ± 12.81	99.39 ± 0.81	99.39 ± 0.81	98.29 ± 1.11, 9.00	99.15 ± 16.11, 18.5	**99.63 ± 0.56**, 55.30
	0.0085	0.3434	0.3434	0.1032	0.3092	
	12.9881	0.9881	1.1590	5.8695	6.1157	10.35
MNIST 5 versus 6 790 × 784	85.32 ± 22.27	91.65 ± 2.60	91.01 ± 2.79	93.54 ± 2.56, 28.90	**96.08 ± 1.91**, 46.9	92.27 ± 10.23, 68.89
	0.035	0.9254	0.9475	0.5446	0.2337	
	14.1933	0.9095	1.0952	11.8694	12.1683	12.56

Time is for onefold in seconds. Means and standard deviations of test correctness and selected features are over tenfold

Bold values are of the highest class accuracy

Table 4 Tenfold testing classification accuracy of nonlinear classifiers with Gaussian kernel

Datasets	GEPSVM Acc ± std (%) No. of Ker., p value Time (s)	TWSVM Acc ± std (%) No. of Ker., p value Time (s)	PTSVM Acc ± std (%) No. of Ker., p value Time (s)	NLPSVM Acc ± std (%) No. of Ker., Ratio p value Time (s)	LPNewton Acc ± std (%), No. of Ker., Ratio p value Time (s)	RFSPTSVM Acc ± std (%), No. of Ker., Ratio p value Time (s)
UCI Ionosphere 351 × 34	87.76 ± 11.44	87.45 ± 5.75	85.75 ± 6.64	92.58 ± 5.60	93.71 ± 4.57	**95.15 ± 2.87**
	316	316	316	14.10, 4.46 %	55.90, 17.69 %	56.15, 17.78 %
	0.0026	0.003	0.0327	0.0676	0.1310	
	0.7504	0.2215	0.3306	0.9886	1.3540	1.4011
UCI BUPA Liver 345 × 6	69.24 ± 3.93	**74.19 ± 7.35**	64.01 ± 9.34	66.09 ± 7.97	68.49 ± 10.86	66.36 ± 7.50
	310	310	310	23.00, 7.40 %	167.80, 54.02 %	49.95, 16.09 %
	0.2432	0.0116	0.4683	0.9302	0.5918	
	0.6774	0.1903	0.2857	1.1812	0.6472	2.5521
UCI Pima 768 × 8	73.19 ± 4.84	**76.70 ± 2.95**	74.48 ± 4.58	72.78 ± 5.77	68.46 ± 10.01	65.25 ± 3.85
	691	691	691	25.20, 3.65 %	55.50, 8.03 %	137.85, 19.94 %
	0.0001	0.00002	0.0155	0.0127	0.3943	
	8.7971	0.9352	1.2644	8.7959	5.4087	3.5166
UCI Sonar 208 × 60	68.21 ± 7.06	82.62 ± 5.61	**86.02 ± 4.10**	79.86 ± 7.24	71.14 ± 15.01	79.90 ± 12.30
	188	188	188	10.50, 5.61 %	35.60, 19.03 %	32.80, 17.52 %
	0.0332	0.5618	0.8947	0.9878	0.1406	
	0.2074	0.1170	0.2164	0.3281	0.2540	0.7840
UCI Votes 435 × 16	81.84 ± 11.03	93.09 ± 2.95	93.09 ± 2.54	92.39 ± 3.14	93.55 ± 2.87	**94.95 ± 2.67**
	392	392	392	15.70, 4.01 %	24.30, 6.21 %	49.70, 12.70 %
	0.0106	0.0521	0.4421	0.0743	0.1100	
	1.5743	0.2972	0.896	2.9440	2.8797	2.9219
UCI Monk3 554 × 6	88.49 ± 4.29	**98.02 ± 1.70**	95.85 ± 2.12	96.22 ± 2.19	95.49 ± 3.77	96.94 ± 2.14
	499	499	499	23.60, 4.73 %	23.00, 4.61 %	75.75, 15.19 %
	0.0003	0.0236	0.00001	0.1051	0.2065	
	1.6957	0.4700	0.6513	5.5091	5.5526	6.2585
UCI Spect 267 × 44	78.69 ± 6.73	77.92 ± 6.69	78.23 ± 3.53	78.30 ± 4.20	**79.44 ± 5.94**	**79.44 ± 8.09**
	240	240	240	5.20, 2.17 %	16.40, 6.82 %	24.75, 10.30 %
	0.8509	0.4509	0.00006	0.5970	0.9947	
	0.4007	0.1644	0.2575	0.4583	0.5170	0.2553
UCI Tic-tac-toe 958 × 9	76.59 ± 10.67	98.85 ± 0.98	**98.33 ± 1.06**	95.00 ± 9.83	95.93 ± 5.35	**98.33 ± 0.95**
	862	862	862	31.80, 3.69 %	62.40, 7.24 %	127.40, 14.78 %
	0.0002	0.0957	0.3280	0.3434	0.2026	
	31.7196	1.8900	2.3551	16.3194	18.9064	21.3597
UCI Germ 1000 × 24	62.10 ± 15.33	**74.80 ± 5.81**	73.30 ± 4.47	70.00 ± 4.42	70.80 ± 4.91	65.60 ± 12.25
	900	900	900	24.30, 2.70 %	33.50, 3.72 %	146.80, 16.31 %
	0.6206	0.0752	0.0001	0.2748	0.2392	
	32.6543	1.9954	2.6218	14.5970	14.2430	10.1959
UCI Monk2 601 × 6	71.87 ± 9.89	**83.39 ± 11.69**	71.69 ± 11.28	66.40 ± 5.56	73.71 ± 10.48	75.73 ± 12.79
	540	540	540	12.50, 2.31 %	89.20, 16.49 %	84.00, 15.53 %
	0.3679	0.1378	0.0001	0.0263	0.6976	
	5.2843	0.5726	0.7995	2.1690	3.6931	2.6026
MNIST 1 versus 2 820 × 784	89.46 ± 3.27	97.44 ± 1.67	**99.76 ± 0.48**	99.51 ± 0.59	99.15 ± 0.78	96.41 ± 3.41
	738	738	738	12.30, 1.67 %	5.90, 0.80 %	30.40, 4.21 %
	0.000614	0.0140	0.1248	0.0263	0.1248	
	20.5025	2.1475	9.2412	9.9743	6.8964	11.2557
MNIST 5 versus 6 790 × 784	95.32 ± 2.99	99.11 ± 1.27	**99.49 ± 0.62**	96.08 ± 6.26	91.01 ± 14.51	98.35 ± 1.50
	711	711	711	64.60, 9.09 %	13.2, 1.86 %	32.80, 4.61 %
	0.0013	0.7803	0.2903	0.2619	0.1191	
	19.0959	1.8650	8.5165	11.8532	9.6744	17.0655

Time is for onefold in seconds. Means and standard deviations of test correctness and numbers of kernel functions are over tenfold

Bold values are of the highest class accuracy

Table 5 Comparison on large NDCC datasets with linear kernel

Datasets	GEPSVM Train (%) Test (%) Features Train time (s)	TWSVM Train (%) Test (%) Features Train time (s)	PTSVM Train (%) Test (%) Features Train time (s)	NLPSVM Train (%) Test (%) Features Train time (s)	LPNewton Train (%) Test (%) Features Train time (s)	RFSPTSVM Train (%) Test (%) Features Train time (s)
NDCC-1k 1000 × 7	93.89 ± 2.01	96.40 ± 0.13	95.29 ± 0.43	96.40 ± 0.13	96.40 ± 0.13	96.40 ± 0.13
	93.10 ± 2.39	96.40 ± 1.20	94.40 ± 2.76	96.40 ± 1.20	96.40 ± 1.20	96.40 ± 1.20
	7 ± 0.0	7 ± 0.0	7 ± 0.0	0.90 ± 0.94	2.20 ± 0.75	3.90 ± 0.62
	0.0011	1.9148	1.9531	0.0092	0.0061	0.0520
NDCC-3k 3000 × 7	86.69 ± 0.52	96.30 ± 0.09	94.93 ± 0.14	96.30 ± 0.09	96.30 ± 0.09	96.30 ± 0.09
	86.67 ± 1.54	96.30 ± 0.85	95.20 ± 1.20	96.30 ± 0.85	96.30 ± 0.85	96.30 ± 0.85
	7 ± 0.0	7 ± 0.0	7 ± 0.0	1.30 ± 1.90	4.30 ± 0.90	3.55 ± 1.01
	0.0015	42.0572	46.6926	0.0937	0.0965	0.2294
NDCC-5k 5000 × 7	82.26 ± 0.96	96.34 ± 0.10	95.52 ± 0.15	96.34 ± 0.10	96.34 ± 0.10	96.34 ± 0.10
	82.06 ± 1.34	96.34 ± 0.89	95.20 ± 1.41	96.34 ± 0.89	96.34 ± 0.89	96.34 ± 0.89
	7 ± 0.0	7 ± 0.0	7 ± 0.0	2.30 ± 1.85	4.10 ± 0.30	2.95 ± 0.91
	0.0014	186.4249	224.1549	0.0474	0.0379	0.4913
NDCC-50k 50,000 × 7	79.97 ± 1.47	*	*	96.24 ± 0.04	96.24 ± 0.04	96.23 ± 0.04
	80.01 ± 1.68	*	*	96.24 ± 0.34	96.24 ± 0.34	96.23 ± 0.35
	7 ± 0.0	*	*	2.80 ± 2.23	5.80 ± 0.60	4.40 ± 0.80
	0.0102	*	*	1.6799	1.0814	3.8713
NDCC-100k 100,000 × 7	79.84 ± 1.46	*	*	96.22 ± 0.02	96.22 ± 0.02	96.21 ± 0.02
	79.85 ± 1.45	*	*	96.22 ± 0.16	96.22 ± 0.16	96.21 ± 0.16
	7 ± 0.0	*	*	3.30 ± 1.68	5.10 ± 0.3	4.05 ± 0.69
	0.0157	*	*	3.9774	2.3859	8.4703

* Experiments were stopped due to too much computational time. Time is for onefold in seconds. Means and standard deviations of correctness and selected features are over tenfold

similar phenomenon to the linear case, RFSPTSVM can strongly suppress the kernel functions.

4.4 NDCC datasets

We also carried out experiments on large normally distributed clusters on cubes (NDCC) datasets which were generated by Thompson's NDCC data generator [41] to further analyze how the computing costs of all these algorithms scale with respect to the number of samples. Tables 4 and 5 give a comparison of the accuracy and computing time for the linear and nonlinear algorithms, respectively. For experiments with the NDCC datasets, we fixed the feature suppression parameter and penalty parameters of all algorithms to be 1 (i.e., $\varepsilon = 1$, $C = 1$, $C_1 = 1$, $C_2 = 1$). For nonlinear cases, we set the Gaussian kernel parameters with $\gamma = 2^4$ for all experiments with nonlinear kernel.

From Table 5, we see that all the algorithms obtain similar classification accuracy except GEPSVM has lower correctness. RFSPTSVM takes a considerably less computational time than TWSVM and PTSVM on all datasets. Furthermore, RFSPTSVM does not require any special optimizers, whereas TWSVM and PTSVM have been implemented with fast interior point solvers of Mosek optimization toolbox for MATLAB. For large NDCC-100k dataset, our RFSPTSVM computing cost is only 8.4703 s. The computing complexity of RFSPTSVM is about $O(2r(m^2 + n^3))(m \gg n)$ where r is the number of iterations in the RFSPTSVM solver. Because RFSPTSVM obtains two hyperplanes in which LPNewton and NLPSVM are ones, thus its computing cost is about twice to LPNewton and NLPSVM. In the feature suppression aspect, RFSPTSVM gives comparable performance LPNewton and NLPSVM. In the nonlinear case, the dimensions of kernel functions are very high. Table 6 shows that RFSPTSVM selected very few kernel functions. On the dataset NDCC-3k, GEPSVM, TWSVM, and PTSVM required 2700 kernel functions in which RFSPTSVM just needed 59.25.

Table 6 Comparison on large NDCC datasets with Gaussian kernel

Datasets	GEPSVM Train (%) Test (%) No. of Ker. Train Time (s)	TWSVM Train (%) Test (%) No. of Ker. Train Time (s)	PTSVM Train (%) Test (%) No. of Ker. Train Time (s)	NLPSVM Train (%) Test (%) No. of Ker., Ratio Train Time (s)	LPNewton Train (%) Test (%) No. of Ker., Ratio Train Time (s)	RFSPTSVM Train (%) Test (%) No. of Ker., Ratio Train Time (s)
NDCC-500 500 × 7	94.18 ± 0.98	100.00 ± 0.0	100.00 ± 0.0	96.27 ± 0.22	96.02 ± 0.21	98.24 ± 0.35
	91.20 ± 3.25	95.00 ± 1.61	86.00 ± 5.29	95.60 ± 1.50	95.40 ± 1.56	95.00 ± 2.19
	450 ± 0.0	450 ± 0.0	450 ± 0.0	3.70 ± 1.00, 0.82 %	3.20 ± 1.17, 0.71 %	15.15 ± 1.30, 3.37 %
	3.2189	0.5764	0.6429	2.3502	2.7645	4.7743
NDCC-700 700 × 7	93.75 ± 0.91	100.00 ± 0.0	100.00 ± 0.0	97.46 ± 0.33	97.14 ± 0.24	97.83 ± 0.31
	89.86 ± 3.35	96.29 ± 1.59	93.00 ± 4.31	96.29 ± 1.71	96.14 ± 1.81	94.43 ± 3.86
	630 ± 0.0	630 ± 0.0	630 ± 0.0	10.40 ± 1.91, 1.65 %	9.80 ± 1.78, 1.56 %	19.80 ± 1.75, 3.14 %
	10.8794	1.4203	1.5561	10.7448	10.8018	13.0596
NDCC-1k 1000 × 7	94.00 ± 2.04	100.00 ± 0.0	100.00 ± 0.0	97.88 ± 0.14	97.74 ± 0.21	98.44 ± 0.12
	87.00 ± 4.96	96.20 ± 1.40	89.90 ± 3.11	96.40 ± 1.20	96.40 ± 1.20	95.40 ± 3.14
	900 ± 0.0	900 ± 0.0	900 ± 0.0	17.80 ± 2.27, 1.98 %	15.20 ± 1.83, 1.69 %	31.10 ± 1.34, 3.46 %
	34.4507	3.7789	4.1212	28.4261	20.0663	33.1474
NDCC-2k 2000 × 7	95.96 ± 0.92	100.00 ± 0.0	100.00 ± 0.0	97.58 ± 0.18	97.39 ± 0.12	97.88 ± 0.14
	93.40 ± 2.68	96.45 ± 0.91	88.20 ± 2.90	96.45 ± 0.61	96.45 ± 0.65	92.00 ± 6.04
	1800 ± 0.0	1800 ± 0.0	1800 ± 0.0	33.00 ± 3.74, 1.83 %	33.40 ± 1.96, 1.86 %	47.70 ± 2.50, 2.65 %
	273.2425	26.6014	28.4173	190.1433	189.8440	371.4334
NDCC-3k 3000 × 7	90.88 ± 4.73	100.00 ± 0.0	100.00 ± 0.0	97.57 ± 0.14	97.46 % ± 0.11	97.68 ± 0.19
	88.50 ± 3.86	95.43 ± 0.76	89.93 ± 2.95	96.93 ± 0.99	96.93 ± 0.99	95.63 ± 1.96
	2700 ± 0.0	2700 ± 0.0	2700 ± 0.0	38.60 ± 2.06, 1.43 %	43.70 ± 2.15, 1.62 %	59.25 ± 2.35, 2.19 %
	880.9547	196.9897	197.8982	497.9220	490.5906	968.5315

5 Conclusion

In this paper, we have improved PTSVM and proposed an effective and efficient algorithm called RFSPTSVM. PTSVM tries to solve two dual QPPs, while RFSPTSVM obtains two projection directions by minimizing an EP function of the LP problem and utilizing a fast generalized Newton algorithm to solve it. Our method outperforms other multi-plane learning algorithms (e.g., GEPSVM, TWSVM, and PTSVM) in terms of feature suppression. Also, it has comparable generalization performance to TWSVM and PTSVM both in the linear and nonlinear cases. Experimental results on Myeloma dataset, UCI datasets, MNIST datasets, and NDCC datasets demonstrate that RFSPTSVM obtains remarkable results of suppressing features especially for datasets in a very high-dimensional input space, which is similar to the fast-feature selection methods NLPSVM and LPNewton. Furthermore, experimental results on the synthetic datasets reveal that RFSPTSVM is capable of dealing with XOR examples, while XOR problem is a difficult test case for NLPSVM and LPNewton. RFSPTSVM takes considerably less computational cost than TWSVM and PTSVM on all datasets in the linear case. For example, our RFSPTSVM can easily classify the NDCC-100k dataset in 8.4703 s. RFSPTSVM is a suitable method of solving classification problems in very high-dimensional spaces. Our further work will try to apply and extend our method to some real-world multi-category classification problems.

Acknowledgments The authors would like to thank the anonymous reviewers for their constructive comments and suggestions. This work was partially supported by the Natural Science Foundation of Jiangsu Province of China (Grant No. BK20140794), the China Postdoctoral Science Foundation (Grant No. 2014M551599), and the Fundamental Research Funds for the Central Universities (Grant No. 30916011326).

References

1. Burges C (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Discov 2:121–167
2. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines: and other kernel-based learning methods. Cambridge University Press, Cambridge
3. Cortes C, Vapnik VN (1995) Support vector networks. Mach Learn 20:273–297
4. Demiriz A, Bennett KP, Breneman CM, Embrechts MJ (2001) Support vector machine regression in chemometrics. In: Computing science and statistics, proceedings of the 33rd symposium on the interface. American Statistical Association for the Interface Foundation of North America, Washington, DC
5. Osuna E, Freund R, Girosi F (1997) Training support vector machines: an application to face detection. In: Proceedings of the 1997 IEEE Computer Society conference on computer vision pattern recognition, pp 130–136
6. Jia G, Martinez AM (2009) Support vector machines in face recognition with occlusions. In: Proceedings of the 2009 IEEE conference on comput vision and pattern recognition, Miami, Florida, pp 136–141
7. Hotta Kazuhiro (2008) Robust face recognition under partial occlusion based on support vector machine with local Gaussian summation kernel. Image Vis Comput 26(11):1490–1498
8. Wang Zhenyu, Yang Wankou, Ben Xianye (2015) Low-resolution degradation face recognition over long distance based on CCA. Neural Comput Appl 26(7):1645–1652
9. Yang Wankou, Wang Zhenyu, Sun Changyin (2015) A collaborative representation based projections method for feature extraction. Pattern Recogn 48(1):20–27
10. Joachims T (1998) Text categorization with support vector machines: learning with many relevant features. Machine Learning ECML-98:137–142
11. Cao L (2003) Support vector machines experts for time series forecasting. Neurocomputing 51:321–339
12. Chen X, Yang J, Liang J, Ye Q (2012) Recursive robust least squares support vector regression based on maximum correntropy criterion. Neurocomputing 97:63–73
13. Zhao Y, Zhao J, Zhao M (2013) Twin least squares support vector regression. Neurocomputing 118:225–236
14. Ben Xianye, Zhang Peng et al (2015) Gait recognition and micro-expression recognition based on maximum margin projection with tensor representation. Neural Comput Appl. doi:10.1007/s00521-015-2031-8
15. Ben Xianye, Meng Weixiao et al (2013) Kernel coupled distance metric learning for gait recognition and face recognition. Neurocomputing 120:577–589
16. Du B, Zhang L (2015) Target detection based on a dynamic subspace. Pattern Recogn 47(1):344–358
17. Du B, Zhang L (2014) A discriminative metric learning based anomaly detection method. IEEE Trans Geosci Remote Sens 52(11):6844–6857
18. Fung G, Mangasarian OL (2001) Proximal support vector machine classifiers. In: Provost F, Srikant R (eds) Proceedings of the knowledge discovery and data mining, pp 77–86
19. Mangasarian O, Wild E (2006) Multisurface proximal support vector classification via generalize eigenvalues. IEEE Trans Pattern Anal Mach Intell 28(1):69–74
20. Jayadeva R, Khemchandani S (2007) Chandra, twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29:905–910
21. Arun Kumar M, Gopal M (2009) Least squares twin support vector machines for pattern classification. Expert Syst Appl 36:7535–7543
22. Ye Q, Zhao C, Ye N, Chen Y (2010) Multi-weight vector projection support vector machines. Pattern Recogn Lett 31(13):2006–2011
23. Chen X, Yang J, Ye Q, Liang J (2011) Recursive projection twin support vector machine via within-class variance minimization. Pattern Recogn. doi:10.1016/j.patcog.2011.03.001
24. Shao Yuan-Hai, Wang Zhen, Chen Wei-Jie, Deng Nai-Yang (2013) A regularization for the projection twin support vector machine. Knowl Based Syst 37:203–210
25. Zhu J, Rosset S, Hastie T, Tibshirani R (2004) 1-norm support vector machines. In: Thrun S, Saul LK, Scholkopf BH (eds) Advances in neural information processing systems16–NIPS2003. MIT Press, Cambridge
26. Golub T, Slonim D, Tamayo P, Huard C, Gaasenbeek M, Mesirov J, Coller H, Loh M, Downing J, Caligiuri M (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286:531–536
27. Guyon I, Weston J, Barhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46:389–422
28. Zhou WD, Zhang L, Jiao LC (2002) Linear programming support vector machines. Pattern Recogn 35(12):2927–2936
29. Zou H (2007) An improved 1-norm SVM for simultaneous classification and variable selection. In: Proceedings of the eleventh international conference on artificial intelligence and statistics
30. Fung G, Mangasarian OL (2004) A feature selection Newton method for support vector machine classification. Comput Optim Appl 28(2):185–202
31. Mangasarian OL (2006) Exact 1-norm support vector machines via unconstrained convex differentiable minimization. J Mach Learn Res 7:1517–1530
32. Gao Shangbing, Ye Q, Ye N (2011) 1-norm least squares twin support vector machines. Neurocomputing 74:3590–3597
33. Bai L, Wang Z, Shao YH et al (2014) A novel feature selection method for twin support vector machine. Knowl Based Syst 59:1–8
34. Ye Q, Zhao C, Ye N, Zheng H, Chen X (2012) A feature selection method for nonparallel plane support vector machine classification. Optim Methods Softw 27(3):431–443
35. Guo J et al (2014) Feature selection for least squares projection twin support vector machine. NeuroComputing 144:174–183
36. Tao Y, Yang J (2010) Quotient vs. difference: comparison between the two discriminant criteria. Neurocomputing 73:1808–1817
37. Mangasarian OL (1994) Nonlinear programming. SIAM, Philadelphia
38. Mangasarian OL, Meyer RR (1979) Nonlinear perturbation of linear programs. SIAM J Control Optim 17(6):745–752
39. Blake C, Merz C (1998) UCI repository of machine learning databases, Department of Information and Computer Sciences, University of California, Irvine. http://www.ics.uci.edu/~mlearn/MLRepository.html
40. Page D, Zhan F, Cussens J, Waddell M, Hardin J, Barlogie B, Shaughnessy J Jr (2002) Comparative data mining for microarrays: a case study based on Multiple Myeloma. Technical Report 1453, Computer Sciences Department, University of Wisconsin
41. Thompson ME (2006) NDCC: normally distributed clustered datasets on cubues. www.cs.wisc.edu/dmi/svm/ndcc
42. LeCun Y, Cortes C (2010) MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/