Neural Process Lett (2016) 44:795–811
DOI 10.1007/s11063-016-9495-0

A Novel Twin Support Vector Machine for Binary Classification Problems

Sugen Chen[1,2] · **Xiaojun Wu**[1] · **Renfeng Zhang**[1]

Published online: 14 January 2016

Abstract Based on the recently proposed twin support vector machine and twin bounded support vector machine, in this paper, we propose a novel twin support vector machine (NTSVM) for binary classification problems. The significance of our proposed NTSVM is that the objective function is changed in the spirit of regression, such that hyperplanes separate as much as possible. In addition, the successive overrelaxation technique is used to solve quadratic programming problems to speed up the training process. Experimental results obtained on several artificial and UCI benchmark datasets show the feasibility and effectiveness of the proposed method.

Keywords Pattern recognition · Binary classification · Twin support vector machine · Successive overrelaxation technique (SOR)

1 Introduction

Image classification is one of the most fundamental problems in computer vision and pattern recognition, which is to assign one or more category labels to an image. However, image classification is a complex process that may be affected by many factors. There are several stages in image classification such as preprocessing, sample selection, feature extraction and classifier design and so on. In those stages, feature extraction and classifier design

✉ Xiaojun Wu
xiaojun_wu_jnu@163.com; wu_xiaojun@jiangnan.edu.cn

Sugen Chen
chensugen@126.com

Renfeng Zhang
zhangfang779797646@gmail.com

[1] School of IoT Engineering, Jiangnan University, Wuxi 214122, People's Republic of China

[2] School of Mathematics & Computational Science, Anqing Normal University, Anqing 246133, People's Republic of China

are especially important. In feature extraction, there are a lot of different methods, such as PCA [1], LDA [2], 2DPCA [3], STL [4] and GTDA [5] and so on. Actually, varied features describe different properties of the same image. In order to make full use of various features, multi-view learning [6,7] and multiple feature fusion [8,9] have been widely studied. In classifier design, support vector machine (SVM) [10,11], as a powerful tool for pattern classification and regression, has been widely used to a variety of real-world problems such as pattern recognition [12,13], bioinformatics [14], text categorization [15,16] and financial applications [17]. Moreover, some related works with regard to SVM are also worthy of attention [18–20]. In order to improve the performance of SVM-based relevance feedback in image retrieval, the asymmetric bagging and random subspace for SVM was proposed [18]. In order to address the problems in LR-based (Laplacian regularization) image annotation, Liu and Tao [19] proposed a multiview framework of SVM for image annotation. Furthermore, aiming to solve the class separation problem, Tao et al. [20] discussed subspace learning for SVM. However, SVM is computationally expensive because its solution follows from solving a quadratic programming problem (QPP) with numerous constraints, especially when handing large-scale data. In addition, SVM seeks only a separating plane such that it cannot well cope with the complex XOR problems. Recently, in order to improve the generalization ability and computational speed of SVM, multi-hyperplane support vector machine, which seeks two nonparallel hyperplanes instead of one hyperplane in SVM, has been widely investigated. In 2006, Mangasarian and Wild [21] proposed a generalized eigenvalue proximal support vector machine (GEPSVM), which is originally motivated to solve XOR problem and reduce the computing time of SVM. In this approach, training points of each class are proximal to one of two nonparallel hyperplanes. Next year, Jayadeva, Khemchandani and Chandra [22] proposed another nonparallel hyperplane classifier named twin support vector machine (TWSVM), which aims at seeking two nonparallel hyperplanes such that each hyperplane is close to one of the two classes and is as far as possible from the other simultaneously. Experimental results in [22] and [23] have shown the effectiveness of TWSVM over both standard SVC and GEPSVM, at the same time, TWSVM can also well deal with the XOR problems. Therefore, from then on, the methods of constructing the nonparallel hyperplanes support vector machine have been extensively studied, such as twin bounded support vector machine (TBSVM) [24], TPMSVM [25], RTSVM [26], STSVM [27], PPSVC [28] and NHSVM [29] and so on. Similarly, the methods of finding two projection directions [30–34] and two hyperspheres [35,36] also have been widely investigated, such as MVSVM [30], PTSVM [31] and TSVH [36].

However, the above nonparallel hyperplane methods [22–24,29] are almost in seeking two nonparallel hyperplanes $w_1^T x + b_1 = 0$, $w_2^T x + b_2 = 0$ and minimizing $||Aw_1 + e_1 b_1||_2^2$ or $||Bw_2 + e_2 b_2||_2^2$. Inspired by [37], in this paper, we propose a novel twin support vector machine, which aims at seeking two hyperplanes $w_1^T x + b_1 = 1$, $w_2^T x + b_2 = -1$ and minimizing $||Aw_1 + e_1 b_1 - e_1||_2^2$ or $||Bw_2 + e_2 b_2 + e_2||_2^2$ such that the two hyperplanes separate as much as possible. Furthermore, a regularization term is used to overcome overfitting and singularity problem, which is similar to [24]. Thus, the structural risk minimization principle is implemented.

The rest paper is organized as follows. Section 2 gives a brief overview of TWSVM and TBSVM. Section 3 proposes our linear novel twin support vector machine (NTSVM) and extends to nonlinear NTSVM. Experimental results are described in Sect. 4, and finally, conclusions are given in Sect. 5.

2 Brief Reviews of TWSVM and TBSVM

2.1 TWSVM

Consider a binary classification problem of m_1 data points in positive class and m_2 data points in negative class. Suppose that all of the data points in positive class are denoted by a matrix $A \in R^{m_1 \times n}$ and the data points in negative class are denoted by a matrix $B \in R^{m_2 \times n}$. A_i is the i-th row of A which is a row vector in R^n. For linear case, TWSVM [22] determines two nonparallel hyperplanes:

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

where $w_i \in R^n, b_i \in R, i = 1, 2$. The TWSVM aims at seeking a pair of nonparallel hyperplanes for the two classes such that each hyperplane is close to one of the two classes and is at a distance at least one from the other class. The primal problems of TWSVM are

$$\min_{w_1, b_1, \xi_2} \frac{1}{2} \left\| Aw_1 + e_1 b_1 \right\|_2^2 + c_1 e_2^T \xi_2 \quad s.t. \quad -(Bw_1 + e_2 b_1) + \xi_2 \geq e_2, \quad \xi_2 \geq 0 \tag{2}$$

$$\min_{w_2, b_2, \xi_1} \frac{1}{2} \left\| Bw_2 + e_2 b_2 \right\|_2^2 + c_2 e_1^T \xi_1 \quad s.t. \quad (Aw_2 + e_1 b_2) + \xi_1 \geq e_1, \quad \xi_1 \geq 0 \tag{3}$$

where c_1 and c_2 are the positive penalty parameters, ξ_1 and ξ_2 are nonnegative slack variables, e_1 and e_2 are vectors of all ones of appropriate dimensions. By introducing the Lagrangian multipliers, the Wolfe dual of quadratic programming problems (QPPs) (2) and (3) can be represented as follows, respectively.

$$\max_{\alpha} \; e_2^T \alpha - \frac{1}{2} \alpha^T G (H^T H)^{-1} G^T \alpha \quad s.t. \; 0 \leq \alpha \leq c_1 e_2 \tag{4}$$

$$\max_{\gamma} \; e_1^T \gamma - \frac{1}{2} \gamma^T H (G^T G)^{-1} H^T \gamma \quad s.t. \; 0 \leq \gamma \leq c_2 e_1 \tag{5}$$

where $G = [B \; e_2], \; H = [A \; e_1]$ and $\alpha \in R^{m_2}, \gamma \in R^{m_1}$ are Lagrangian multipliers.

The nonparallel hyperplanes (1) can be obtained from the solutions α and γ to the optimization problems (4) and (5) by

$$v_1 = -(H^T H)^{-1} G^T \alpha, \quad \text{where} \quad v_1 = \left[w_1^T \; b_1 \right]^T, \tag{6}$$

$$v_2 = (G^T G)^{-1} H^T \gamma, \quad \text{where} \quad v_2 = \left[w_2^T \; b_2 \right]^T, \tag{7}$$

A new data point $x \in R^n$ is then assigned to the positive class W_1 or negative class W_2, depending on which of the hyperplanes in (1) it lies closer to, i.e.

$$x \in W_i, \quad i = \arg \min_{i=1,2} \left| w_i^T x + b_i \right| \tag{8}$$

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

2.2 TBSVM

Following the basic idea of SVC and TWSVM, TBSVM [24] also seeks a pair of nonparallel hyperplanes such that each hyperplane is proximal to the data points of one class and far from the data points of another class. The main different between TBSVM and TWSVM is that the structural risk minimization principle is implemented by adding the regularization term in TBSVM. The primal problems of TBSVM are

$$\begin{aligned} &\min_{w_1,b_1,\xi_2} \ \frac{1}{2}c_3\left(||w_1||^2+b_1^2\right)+\frac{1}{2}\left|\left|Aw_1+e_1b_1\right|\right|_2^2+c_1e_2^T\xi_2 \\ &s.t. \ -(Bw_1+e_2b_1)+\xi_2\geq e_2, \quad \xi_2\geq 0 \end{aligned} \tag{9}$$

$$\begin{aligned} &\min_{w_2,b_2,\xi_1} \ \frac{1}{2}c_4\left(||w_2||^2+b_2^2\right)+\frac{1}{2}\left|\left|Bw_2+e_2b_2\right|\right|_2^2+c_2e_1^T\xi_1 \\ &s.t. \ (Aw_2+e_1b_2)+\xi_1\geq e_1, \quad \xi_1\geq 0 \end{aligned} \tag{10}$$

where c_1 and c_2 are the positive penalty parameters, c_3 and c_4 are the positive trade-off parameters, ξ_1 and ξ_2 are nonnegative slack variables, e_1 and e_2 are vectors of all ones of appropriate dimensions. By introducing the Lagrangian multipliers, the Wolfe dual of QPPs (9) and (10) can be represented as follows.

$$\max_{\alpha} \ e_2^T\alpha-\frac{1}{2}\alpha^TG\left(H^TH+c_3I\right)^{-1}G^T\alpha \quad s.t. \ 0\leq\alpha\leq c_1e_2 \tag{11}$$

$$\max_{\gamma} \ e_1^T\gamma-\frac{1}{2}\gamma^TH\left(G^TG+c_4I\right)^{-1}H^T\gamma \quad s.t. \ 0\leq\gamma\leq c_2e_1 \tag{12}$$

where $G=[B \ \ e_2]$, $H=[A \ \ e_1]$, I is an identity matrix and$\alpha\in R^{m_2}, \gamma\in R^{m_1}$ are Lagrangian multipliers.

The nonparallel hyperplanes (1) can be obtained from the solutions α and γ to the optimization problems (11) and (12) by

$$v_1=-\left(H^TH+c_3I\right)^{-1}G^T\alpha, \quad \text{where} \quad v_1=\left[w_1^T \ b_1\right]^T \tag{13}$$

$$v_2=\left(G^TG+c_4I\right)^{-1}H^T\gamma, \quad \text{where} \quad v_2=\left[w_2^T \ b_2\right]^T \tag{14}$$

Once the solutions $[w_1^T \ b_1]$ and $[w_2^T \ b_2]$ to (11) and (12) are obtained from (13) and (14), a new data point $x\in R^n$ is assigned to the positive class W_1 or negative class W_2, depending on which of the hyperplanes in (1) it lies closer to, i.e.

$$x\in W_i, \quad i=\arg\min_{i=1,2}\frac{|w_i^Tx+b_i|}{||w_i||} \tag{15}$$

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

3 Novel Twin Support Vector Machines

3.1 Linear NTSVM

Now, let us construct our linear NTSVM. Following the idea of PSVM [37] and TBSVM [24], we propose our method, which seeks two nonparallel hyperplanes:

$$w_1^Tx+b_1=1 \quad \text{and} \quad w_2^Tx+b_2=-1 \tag{16}$$

by considering the following primal problems.

$$\begin{aligned}
&\min_{w_1,b_1,\xi_2} \ \frac{1}{2}c_3\left(||w_1||_2^2 + b_1^2\right) + \frac{1}{2}\left|\left|Aw_1 + e_1b_1 - e_1\right|\right|_2^2 + c_1e_2^T\xi_2 \\
&s.t. \ \ -(Bw_1 + e_2b_1) + \xi_2 \geq e_2, \quad \xi_2 \geq 0 \qquad (17) \\
&\min_{w_2,b_2,\xi_1} \ \frac{1}{2}c_4\left(||w_2||_2^2 + b_2^2\right) + \frac{1}{2}\left|\left|Bw_2 + e_2b_2 + e_2\right|\right|_2^2 + c_2e_1^T\xi_1 \\
&s.t. \ \ (Aw_2 + e_1b_2) + \xi_1 \geq e_1, \quad \xi_1 \geq 0 \qquad (18)
\end{aligned}$$

where c_1 and c_2 are the positive penalty parameters, c_3 and c_4 are the positive trade-off parameters, ξ_1 and ξ_2 are nonnegative slack variables, e_1 and e_2 are vectors of all ones of appropriate dimensions.

Then, we discuss the difference between the primal problems of TBSVM and our NTSVM, by comparing optimization problems (9) and (17). Obviously, minimizing $||Aw_1+e_1b_1-e_1||_2^2$ in (17) instead of minimizing $||Aw_1 + e_1b_1||_2^2$ in (9) means our NTSVM aims to obtain $[w_1^T \ b_1]$ such that the positive data points are proximal to the hyperplane$w_1^T x + b_1 = 1$ and the negative data points are far away from it at the same time. Based on this change, our optimization problem will be similar to regression problem, i.e., we generate a function $f_1(x) = w_1^T x + b_1$ to fit the positive class label +1. Meanwhile, we find another function $f_2(x) = w_2^T x + b_2$ to fit the negative class label −1. In addition, similar to TBSVM, our NTSVM also adds a regularization term to the objective function in (17). Thus, the structural risk minimization principle is considered.

In order to get the solutions to problems (17) and (18), we need to derive their dual problems. For (17), by introducing the Lagrangian multipliers, the Lagrangian function is given by

$$\begin{aligned}
L(w_1, b_1, \xi_2, \alpha, \beta) = {} & \frac{1}{2}c_3\left(||w_1||_2^2 + b_1^2\right) + \frac{1}{2}\left|\left|Aw_1 + e_1b_1 - e_1\right|\right|_2^2 + c_1e_2^T\xi_2 \\
& -\alpha^T\left[-(Bw_1 + e_2b_1) + \xi_2 - e_2\right] - \beta^T\xi_2 \qquad (19)
\end{aligned}$$

where $\alpha \in R^{m_2}$ and $\beta \in R^{m_2}$ are the vectors of Lagrangian multipliers. By using Karush-Kuhn-Tucker (KKT) conditions, we can get

$$\nabla_{w_1} = c_3w_1 + A^T(Aw_1 + e_1b_1 - e_1) + B^T\alpha = 0 \qquad (20)$$

$$\nabla_{b_1} = c_3b_1 + e^T(Aw_1 + e_1b_1 - e_1) + e_2^T\alpha = 0 \qquad (21)$$

$$\nabla_{\xi_2} = c_1e_2 - \alpha - \beta = 0 \qquad (22)$$

$$-(Bw_1 + e_2b_1) + \xi_2 \geq e_2, \quad \xi_2 \geq 0 \qquad (23)$$

$$\alpha^T(Bw_1 + e_2b_1 - \xi_2 + e_2) = 0, \quad \beta^T\xi_2 = 0 \qquad (24)$$

$$\alpha \geq 0, \quad \beta \geq 0 \qquad (25)$$

Since $\beta \geq 0$, from (22) and (25), we can get

$$0 \leq \alpha \leq c_1e_2 \qquad (26)$$

Let $G = [B \ e_2], \ H = [A \ e_1], v_1 = [w_1^T \ b_1]^T$, (20) and (21) imply that

$$\left(H^TH + c_3I\right)v_1 - H^Te_1 + G^T\alpha = 0 \qquad (27)$$

Thus, we can get the augmented vector

$$v_1 = \left(H^T H + c_3 I\right)^{-1} \left[H^T e_1 - G^T \alpha\right] \tag{28}$$

where I is an identity matrix. Then putting (28) into (19) and using (20–22), omitting the constant term, we obtain the Wolfe dual problem of (17) as follows

$$\begin{aligned} &\max_{\alpha} \left[e_2^T + e_1^T H\left(H^T H + c_3 I\right)^{-1} G^T\right]\alpha - \frac{1}{2}\alpha^T G\left(H^T H + c_3 I\right)^{-1} G^T \alpha \\ &\quad s.t.\ \ 0 \le \alpha \le c_1 e_2 \end{aligned} \tag{29}$$

In the same way, we can obtain the Wolfe dual problem of (18) as follows

$$\begin{aligned} &\max_{\gamma} \left[e_1^T + e_2^T G\left(G^T G + c_4 I\right)^{-1} H^T\right]\gamma - \frac{1}{2}\gamma^T H\left(G^T G + c_4 I\right)^{-1} H^T \gamma \\ &\quad s.t.\ \ 0 \le \gamma \le c_2 e_1 \end{aligned} \tag{30}$$

where $\alpha \in R^{m_2}$, $\gamma \in R^{m_1}$ are Lagrangian multipliers.

The nonparallel hyperplanes (16) can be obtained from the solutions α and γ to the optimization problems (29) and (30) by

$$v_1 = \left(H^T H + c_3 I\right)^{-1} \left[H^T e_1 - G^T \alpha\right], \quad \text{where} \quad v_1 = \left[w_1^T\ b_1\right]^T \tag{31}$$

$$v_2 = \left(G^T G + c_4 I\right)^{-1} \left[- G^T e_2 + H^T \gamma\right], \quad \text{where} \quad v_2 = \left[w_2^T\ b_2\right]^T \tag{32}$$

Once v_1 and v_2 are obtained from (31) and (32), the two nonparallel hyperplanes (16) are known. A new data point $x \in R^n$ is then assigned to the positive class W_1 or negative class W_2, depending on which of the hyperplanes in (16) it lies closer to, i.e.

$$x \in W_i, \quad i = \arg\min_{i=1,2} \left\{ \left|w_1^T x + b_1 - 1\right|, \left|w_2^T x + b_2 + 1\right| \right\} \tag{33}$$

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

3.2 Nonlinear NTSVM

In this subsection, we show that our NTSVM can be extended to nonlinear case. Here, we consider the following kernel-generated hyperplanes

$$K(x^T, C^T)w_1 + b_1 = 1 \quad \text{and} \quad K(x^T, C^T)w_2 + b_2 = -1 \tag{34}$$

where $C = [A\ B] \in R^{(m_1+m_2)\times n}$ and K is an appropriate kernel. Similar to linear case, the nonlinear optimization problems can be expressed as

$$\begin{aligned} &\min_{w_1, b_1, \eta_2} \ \frac{1}{2}c_3\left(||w_1||_2^2 + b_1^2\right) + \frac{1}{2}\left|\left|K(A, C^T)w_1 + e_1 b_1 - e_1\right|\right|_2^2 + c_1 e_2^T \eta_2 \\ &\quad s.t.\ \ -\left(K(B, C^T)w_1 + e_2 b_1\right) + \eta_2 \ge e_2, \quad \eta_2 \ge 0 \end{aligned} \tag{35}$$

$$\min_{w_2, b_2, \eta_1} \ \frac{1}{2}c_4\left(||w_2||_2^2 + b_2^2\right) + \frac{1}{2}\left|\left|K(B, C^T)w_2 + e_2 b_2 + e_2\right|\right|_2^2 + c_2 e_1^T \eta_1$$

$$s.t. \quad \left(K(A, C^T)w_2 + e_1 b_2\right) + \eta_1 \geq e_1, \quad \eta_1 \geq 0 \tag{36}$$

By using the Lagrangian method and KKT conditions, we can obtain Wolfe dual problems as follows

$$\begin{aligned} &\max_{\alpha} \left[e_2^T + e_1^T \tilde{H}\left(\tilde{H}^T \tilde{H} + c_3 I\right)^{-1} \tilde{G}^T\right]\alpha - \frac{1}{2}\alpha^T \tilde{G}\left(\tilde{H}^T \tilde{H} + c_3 I\right)^{-1} \tilde{G}^T \alpha \\ &s.t. \quad 0 \leq \alpha \leq c_1 e_2 \end{aligned} \tag{37}$$

$$\begin{aligned} &\max_{\gamma} \left[e_1^T + e_2^T \tilde{G}\left(\tilde{G}^T \tilde{G} + c_4 I\right)^{-1} \tilde{H}^T\right]\gamma - \frac{1}{2}\gamma^T \tilde{H}\left(\tilde{G}^T \tilde{G} + c_4 I\right)^{-1} \tilde{H}^T \gamma \\ &s.t. \quad 0 \leq \gamma \leq c_2 e_1 \end{aligned} \tag{38}$$

where $\tilde{G} = [K(B, C^T) \ e_2]$, $\tilde{H} = [K(A, C^T) \ e_1]$. From (35)-(38), and the augmented vectors $v_1 = [w_1^T \ b_1]^T$ and $v_2 = [w_2^T \ b_2]^T$ can be obtained by

$$v_1 = \left(\tilde{H}^T \tilde{H} + c_3 I\right)^{-1}\left[\tilde{H}^T e_1 - \tilde{G}^T \alpha\right] \tag{39}$$

$$v_2 = \left(\tilde{G}^T \tilde{G} + c_4 I\right)^{-1}\left[-\tilde{G}^T e_2 + \tilde{H}^T \gamma\right] \tag{40}$$

Once the augmented vector v_1 and v_2 are obtained from (39) and (40), the two nonparallel hyperplanes (34) are known. A new data point $x \in R^n$ is then assigned to the positive class W_1 or negative class W_2, depending on which of the hyperplanes in (34) it lies closer to, i.e.

$$x \in W_i, \quad i = \arg\min_{i=1,2} \left\{ \left|w_1^T K(x^T, C^T) + b_1 - 1\right|, \left|w_2^T K(x^T, C^T) + b_2 + 1\right| \right\} \tag{41}$$

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

3.3 Implementation

In this subsection, we discuss the implementation of our proposed NTSVM. In our NTSVM, the dual problem can be rewritten as the following unified form

$$\begin{aligned} &\min_{\alpha} \quad \frac{1}{2}\alpha^T Q\alpha - f^T \alpha \\ &s.t. \quad 0 \leq \alpha \leq ce \end{aligned} \tag{42}$$

where Q is a positive definite matrix and f is a vector. For example, if we choose $Q = G(H^T H + c_3 I)^{-1} G^T$ and $f = [e_2^T + e_1^T H(H^T H + c_3 I)^{-1} G^T]^T$, the problem (42) becomes the problem (29). Similarly, the problem (42) becomes problem (37), when we choose $Q = \tilde{G}(\tilde{H}^T \tilde{H} + c_3 I)^{-1} \tilde{G}^T$ and $f = [e_2^T + e_1^T \tilde{H}(\tilde{H}^T \tilde{H} + c_3 I)^{-1} \tilde{G}^T]^T$.

Algorithm 1. SOR Algorithm
Step 1. Select the parameter $t \in (0, 2)$ and the initial vector α^0, set $k = 0$;
Step 2. Compute α^{k+1} by $\alpha^{k+1} = (\alpha^k - t \cdot D^{-1}(Q\alpha^k - f + L(\alpha^{k+1} - \alpha^k)))_\#$; where Q and f are set according to (29), (30), (37) and (38). $(u)_\#$ denotes the 2-norm projection on the feasible region, that is

$$((u)_\#)_i = \begin{cases} 0, & \text{if } u_i \leq 0 \\ u_i, & \text{if } 0 < u_i < c \\ c, & \text{if } u_i \geq c \end{cases}$$

And define $L + D + L^T = Q$, where L and D are the strictly lower triangular matrix and the diagonal matrix, respectively;
Step 3. Stop if $||\alpha^{k+1} - \alpha^k|| < \varepsilon$, where ε is desired tolerance; else replace α^k by α^{k+1}, k by $k + 1$ and go to Step2.

In our proposed NTSVM, most of the computational cost is incurred in solving the dual QPP (42). In order to solve the above QPP quickly, we use a very efficient optimization technique called successive overrelaxation (SOR) algorithm, which can be seen in [24,29,38]. SOR is an excellent QPP solver because it is able to deal with very large datasets that need not reside in memory [38]. The experimental results in the following section will show that the SOR technique has remarkable acceleration effect on our proposed NTSVM. Note that the SOR technique can also be used in the original TBSVM [24] and NHSVM [29]. In practice, for nonlinear NTSVM, if the number of training points in positive class or negative class is large, then the rectangular kernel technique [39] can be applied to reduce the dimensionality of (37) and (38).

4 Experimental Results

In order to evaluate the proposed NTSVM, we investigate its classification accuracies and computational efficiencies on two artificial datasets, a number of real-world UCI benchmark datasets and five NDC datasets. In the experiments, we focus on the comparison between our proposed NTSVM and some state-of-the-art classifiers, e.g. GEPSVM [21], TWSVM [22], TBSVM [24] and NHSVM [29]. All methods were implemented in MATLAB R2013a on a personal computer (PC) with an Intel (R) Core (TM) processor (3.40GHz) and 4 GB random-access memory (RAM). TBSVM, NHSVM and our proposed NTSVM are solved by SOR algorithm. The eigenvalue problem in GEPSVM is solved by MATLAB function 'eig.m' and the QPP problems in TWSVM are solved by the optimization toolbox QP in MATLAB. In our evaluation of the classifiers, we used "Accuracy" which is defined as Accuracy = (TP+TN)/(TP+FP+TN+FN), where TP, TN, FP and FN are the number of true positives, true negatives, false positives and false negatives, respectively. Note that all of these classifiers contained different parameters, which were selected from the set $\{2^{-8}, \ldots, 2^8\}$ by employing the standard 10-fold cross-validation methodology [40]. The parameters used with the different classifiers for linear case are listed in Table 1.

4.1 Toy Examples

In this subsection, two artificial datasets, including crossplane (XOR) and Ripley's synthetic datasets [41] have been used to show that our proposed NTSVM can deal with linearly inseparable problems. Ripley's synthetic dataset contains 250 training points and 1000 test points. The average results of linear GEPSVM, TWSVM, TBSVM, NHSVM and our proposed NTSVM on crossplane (XOR) dataset are reported in Table 2 and the average results

Table 1 The parameters for different classifiers for linear kernel

GEPSVM	TWSVM	TBSVM	NHSVM	NTSVM
δ	(c_1, c_2, ε)	(c_1, c_2, c_3, c_4)	(c_1, c_2)	(c_1, c_2, c_3, c_4)

Table 2 Classification accuracy (%) on crossplane (XOR) datasets

Datasets	GEPSVM	TWSVM	TBSVM	NHSVM	NTSVM
	Acc ± Std (%)	Acc ± Std (%)	Acc ± Std (%)	Acc ± Std (%)	Acc ± Std (%)
	Time (s)	Time (s)	Time (s)	Time (s)	Time (s)
Crossplane	**99.00** ± **2.11**	96.55 ± 1.57	97.80 ± 1.06	98.50 ± 2.42	**99.00** ± **2.11**
200 × 2	0.0003	0.0033	0.0043	0.0034	0.0183

Bold values indicate best accuracy

Table 3 Classification accuracy (%) on ripley datasets

Datasets	GEPSVM	TWSVM	TBSVM	NHSVM	NTSVM
	Acc %	Acc %	Acc %	Acc %	Acc %
Ripley 1000 × 2	84.80	86.40	87.20	**90.20**	89.20

Bold value indicates best accuracy

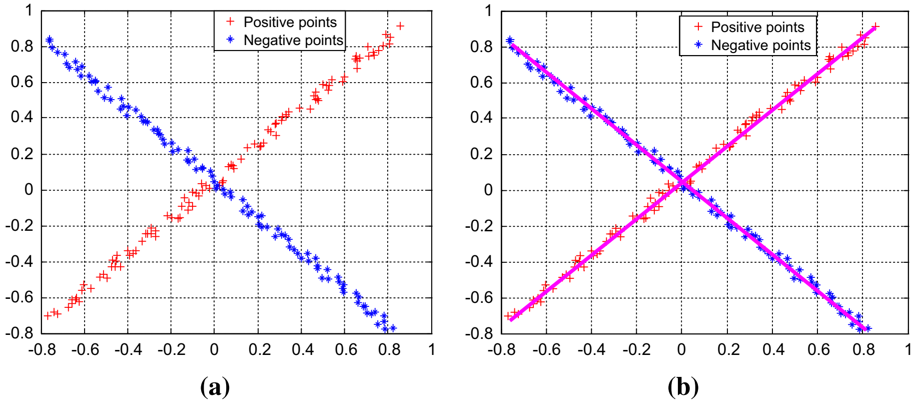


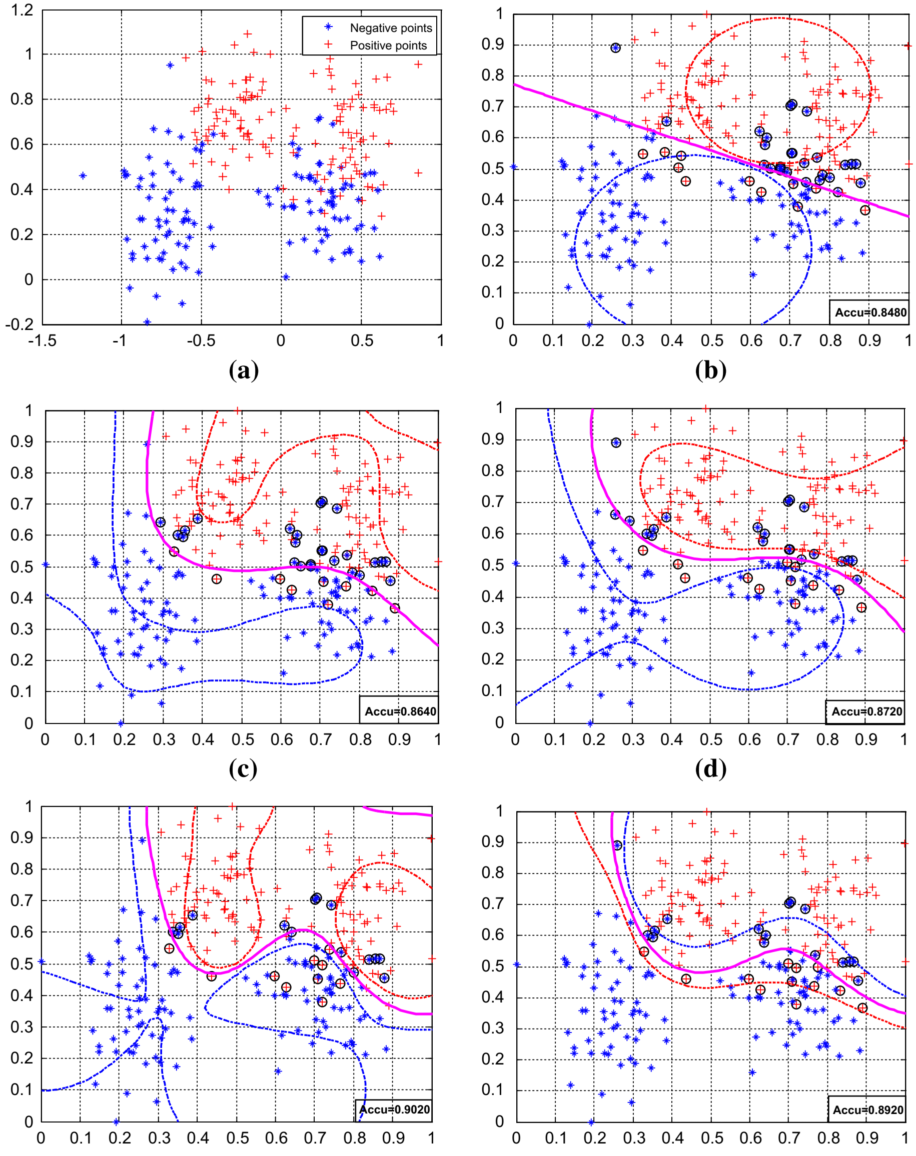
Fig. 1 Crossplane and the hyperplanes of our proposed NTSVM

of nonlinear GEPSVM, TWSVM, TBSVM, NHSVM and our proposed nonlinear NTSVM with Gaussian kernels on Ripley's synthetic dataset are listed in Table 3.

From Table 2, we can observe that all method above can gain good performance on crossplane (XOR) problem and our proposed NTSVM get the best performance. From Table 3, we can find that the accuracy of our proposed NTSVM is 89.20 %, which is little lower than NHSVM but better than other methods. Figure 1 illustrates the crossplane (XOR) dataset and hyperplanes of our proposed NTSVM. Ripley's dataset and the hyperplanes of GEPSVM, TWSVM, TBSVM, NHSVM and our proposed nonlinear NTSVM are shown in Fig. 2.

4.2 UCI Datasets

To further compare our proposed NTSVM with GEPSVM, TWSVM and TBSVM, we choose 13 datasets from the UCI machine learning repository [42]. The numerical results of their linear version are given in Table 4. We also list the mean accuracy for each of the classifiers



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Fig. 2 Ripley's dataset and results of nonlinear methods on Ripley's dataset

and the best accuracy is shown in boldface. According to the W-T-L summarization, we can find that the accuracy of our proposed linear NTSVM is better than that of GEPSVM, TWSVM and TBSVM on most of the datasets. For example, for the Australian dataset, the accuracy of our NTSVM is 87.54 %, while that of GEPSVM is 85.65 %, TWSVM is 86.81 % and TBSVM is 86.96 %.

Table 4 Ten-fold testing percentage accuracy of linear classifiers

Datasets	GEPSVM	TWSVM	TBSVM	NTSVM
	Accuracy % Time(s)	Accuracy % Time(s)	Accuracy % Time(s)	Accuracy % Time(s)
Australian	85.65 ± 4.12	86.81 ± 4.07	86.96 ± 2.65	**87.54 ± 2.57**
(690 × 14)	4.2621e−04	0.3891	0.4269	0.5505
BUPA liver	56.48 ± 7.81	69.53 ± 6.70	69.53 ± 9.45	**70.45 ± 4.98**
(345 × 6)	3.2295e−04	0.0630	0.0338	0.1620
House votes	94.02 ± 3.95	95.18 ± 5.11	95.64 ± 2.96	**95.86 ± 2.62**
(435 × 16)	4.6331e−04	0.0754	0.0569	0.1098
Heart-c	82.84 ± 5.80	85.17 ± 7.45	**85.86 ± 6.45**	85.49±6.64
(303 × 14)	3.3979e−04	0.0210	0.0303	0.0848
Heart-Statlog	83.70 ± 5.30	84.81 ± 4.77	85.19 ± 4.28	**85.56 ± 4.43**
(270 × 13)	3.5582e−04	0.0332	0.0456	0.0258
Ionosphere	79.18 ± 6.51	91.18 ± 5.42	91.16 ± 4.15	**92.88 ± 4.30**
(351 × 34)	0.0014	0.0385	0.0183	0.0526
Monk2	58.08 ± 7.75	67.16 ± 7.62	67.17 ± 7.80	**67.18 ± 8.49**
(432 × 7)	3.2721e−04	0.1620	0.1102	0.0416
Monk3	81.01 ± 6.06	**87.49 ± 7.42**	81.71 ± 6.25	83.34 ± 3.38
(432 × 7)	2.1785e−04	0.0460	0.0697	0.1036
Musk	80.01 ± 4.60	85.91 ± 3.89	86.55 ± 3.48	**88.23 ± 5.44**
(476 × 166)	0.0739	0.2313	0.1701	0.1554
Pima-Indian	76.43 ± 2.84	77.49 ± 5.10	78.00 ± 5.49	**78.13 ± 3.05**
(768 × 8)	2.8757e−04	0.2740	0.2028	0.4703
Sonar	78.90 ± 7.41	78.76 ± 7.22	80.26 ± 9.50	**80.79 ± 9.39**
(208 × 60)	0.0046	0.0315	0.0125	0.0667
Spect	79.40 ± 8.79	80.19 ± 6.97	**81.28 ± 7.24**	80.93 ± 7.79
(267 × 44)	0.025	0.0632	0.0293	0.0680
Wpbc	78.87 ± 10.38	82.89 ± 5.23	82.95 ± 8.48	**83.37 ± 7.77**
(198 × 34)	0.0013	0.1867	0.0668	0.0873
Mean Accuracy	78.04	82.51	82.48	**83.06**
W-T-L	13-0-0	12-0-1	11-0-2	

Bold values indicate best accuracy

Table 5 displays the experimental results for nonlinear GEPSVM, TWSVM, TBSVM and our NTSVM on the above chosen UCI datasets. The Gaussian kernel $K(x, y) = e^{-\frac{||x-y||^2}{2\sigma^2}}$ is used. The kernel parameter σ is obtained by searching in the range from 2^{-8} to 2^{8}. The accuracy and the training CPU time for these methods are also listed. The results in Table 5 are similar to those in Table 4, and therefore confirm the observation above.

In order to compare the performance of different classifiers, the two-dimensional scatter plots were used in [19,26,28,30]. The corresponding scatter plots are shown in Fig. 3 for Australian UCI datasets with about 20 % of data points, where the coordinates (d_i^+, d_i^-) are the respective distances of a test point x_i given in (33). In Fig. 3, the figures (a–d) are the results obtained by GEPSVM, TWSVM, TBSVM and our NTSVM respectively.

Table 5 Ten-fold testing percentage accuracy of nonlinear classifiers

Datasets	GEPSVM	TWSVM	TBSVM	NTSVM
	Accuracy% Time(s)	Accuracy% Time(s)	Accuracy% Time(s)	Accuracy% Time(s)
Australian	86.38 ± 2.83	86.96 ± 4.27	87.10 ± 3.83	**87.39 ± 4.38**
(690 × 14)	5.3616	0.6897	1.6723	1.1396
BUPA liver	70.47 ± 7.87	74.82 ± 5.75	73.55 ± 8.10	**75.35 ± 7.68**
(345 × 6)	0.4689	0.2307	0.2391	0.1535
House votes	90.36 ± 5.49	95.87 ± 3.17	95.86 ± 1.80	**96.32 ± 1.95**
(435 × 16)	1.0466	0.2210	0.4401	0.2831
Heart-c	81.82 ± 5.03	83.49 ± 4.15	**85.51 ± 3.76**	84.82 ± 5.69
(303 × 14)	0.3666	0.1442	0.1916	0.1751
Heart-Statlog	82.96 ± 7.45	82.59 ± 8.74	84.07 ± 6.54	**84.81 ± 5.91**
(270 × 13)	0.2488	0.1130	0.1916	0.0975
Ionosphere	87.75 ± 5.73	94.59 ± 2.84	95.72 ± 3.87	**96.29 ± 4.05**
(351 × 34)	0.5045	0.1352	0.3562	0.2327
Monk2	84.46 ± 7.07	**97.46 ± 1.70**	93.51 ± 3.07	96.06 ± 3.79
(432 × 7)	0.9839	0.2739	0.3823	0.4029
Monk3	92.36 ± 3.96	97.45 ± 1.72	97.47 ± 2.52	**97.68 ± 1.90**
(432 × 7)	0.9985	0.1832	0.4620	0.3060
Musk	85.48 ± 7.08	94.77 ± 3.83	95.17 ± 1.74	**95.38 ± 1.33**
(476 × 166)	1.7225	0.6172	0.9081	0.5824
Pima-Indian	74.61 ± 4.27	78.26 ± 5.83	**78.39 ± 3.31**	78.24 ± 6.88
(768 × 8)	8.0461	0.6741	1.4226	0.9513
Sonar	86.55 ± 8.02	90.88 ± 4.72	**90.95 ± 9.10**	90.90 ± 5.20
(208 × 60)	0.1152	0.0706	0.1194	0.1033
Spect	80.93 ± 4.76	82.78 ± 7.50	82.78 ± 6.08	**83.90 ± 4.28**
(267 × 44)	0.2467	0.0823	0.1644	0.1473
Wpbc	79.76 ± 5.43	82.87 ± 6.24	82.34 ± 5.83	**83.87 ± 8.75**
(198 × 34)	0.1109	0.0995	0.1260	0.0967
Mean Accuracy	83.38	87.91	87.88	**88.54**
W-T-L	13-0-0	11-0-2	10-0-3	

Bold values indicate best accuracy

As we know, there are four parameters $c_i\,(i = 1, 2, 3, 4)$ and a kernel parameter σ in our linear NTSVM and nonlinear NTSVM, respectively. We will analyze the effect of c_i to linear NTSVM and the effect of kernel parameter σ to nonlinear NTSVM. All the experiments in this subsection will be carried out on four benchmark datasets: "Australian", "House votes", "Heart-c" and "Ionosphere". We first evaluate the influence of the parameters $c_i\,(i = 1, 2, 3, 4)$. In real experiments, for convenience, we assume $c_1 = c_2$ and $c_3 = c_4$ for each dataset. The results are depicted in Fig. 4. As we can see, proper selection of parameters c_i will improve the performance of the classifier in a certain extent. We next illustrate the relationship between kernel parameter σ and classification accuracy in the nonlinear case. To analyze the effect of kernel parameter σ on the performance of NTSVM for the nonlinear case, the comparison is carried out under the condition that the parameter c_i are all selected as 1 for

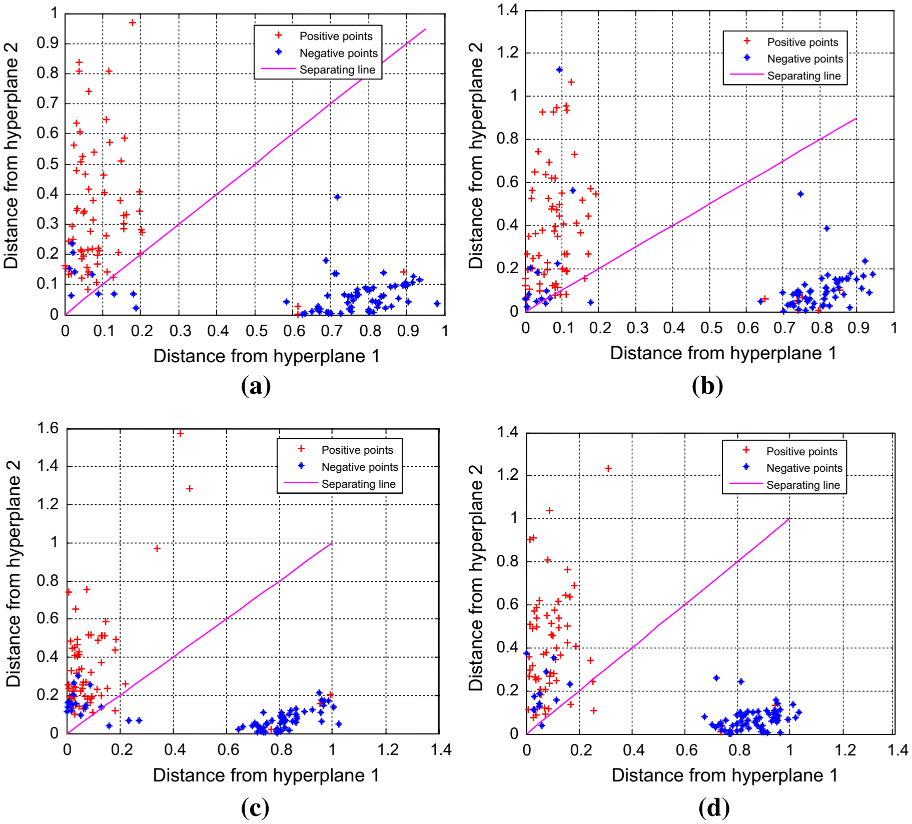


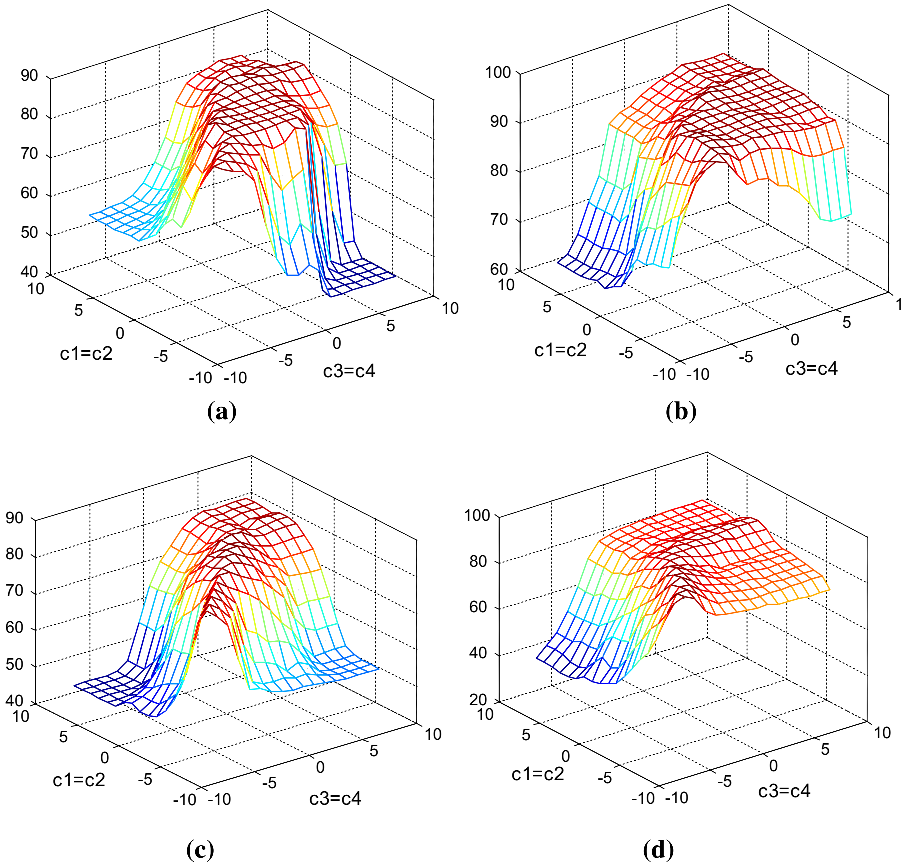
Fig. 3 Two-dimensional projection for test points from Australian dataset

convenience. To clarify the influence of the kernel parameter, we draw the parameter-accuracy curves of above four datasets with parameter σ belongs to the set $\{2^{-8}, 2^{-7}, \cdots, 2^{7}, 2^{8}\}$. Based on the results in Fig. 5, we can see that the performance of NTSVM is closely related to the chosen of kernel parameters.

4.3 NDC Datasets

In this subsection, we conducted experiments on some large datasets, generated using David Musicants NDC Data Generator [43]. For experiments with NDC datasets, we fixed parameters of all classifiers to be the same (i.e.$c_i = 1$). The training time and accuracy are reported in Tables 6 and 7, respectively. Table 6 shows the comparison results for linear TWSVM, TBSVM, NHSVM and our NTSVM on NDC datasets. In Table 6, it is not hard to see that our NTSVM obtains the comparable or better accuracies and performs faster than other classifiers on most datasets.

For nonlinear case, Table 7 shows the comparison results of computing times and accuracies of all four classifiers considered on several NDC datasets with Gaussian kernel. For NDC-2k, NDC-3k and NDC-5k datasets, the rectangular Generator [39] is applied by using 10 % of total data points. The results on these datasets show that our NTSVM and TBSVM are much faster than TWSVM and NHSVM. This is because that the QPP problems in TWSVM



(a) **(b)**

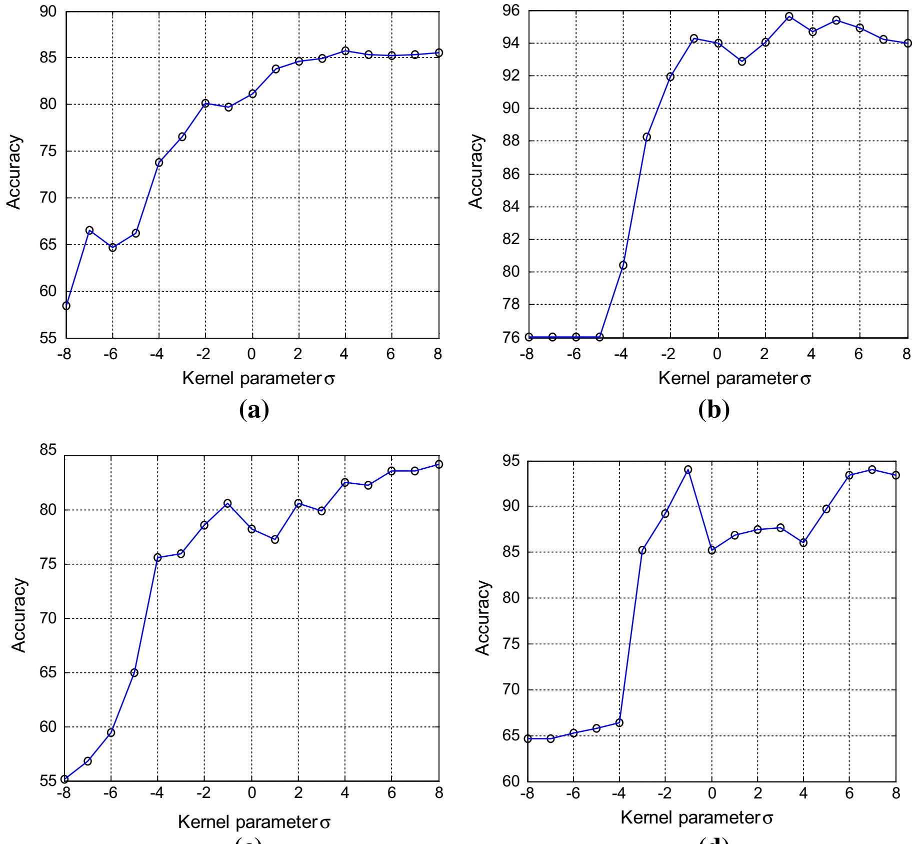
(c) **(d)**

Fig. 4 Relationship of parameters c_i and classification accuracy

are directly solved by the optimization toolbox QP in MATLAB, while our NTSVM and TBSVM are solved by SOR algorithm. Although the QPP problem in NHSVM is also solved by SOR algorithm, the computing time is more than NTSVM and TBSVM. The main reason is that the size of QPP in NHSVM is large than the size of QPPs in NTSVM and TBSVM. Tables 6 and 7 demonstrate the good generalization ability and efficiency of our NTSVM when dealing with large-scale problems.

5 Conclusions

For binary classification, an improved twin support vector machine (NTSVM) based on TWSVM is proposed in this paper. Unlike TWSVM, our proposed NTSVM seeks two non-parallel hyperplanes $w_1^T x + b_1 = 1$ and $w_2^T x + b_2 = -1$. Similar to TBSVM, a regularization term is used to overcome overfitting and singularity problem. In addition, SOR technique is utilized to speed up training process. Experimental results illustrate that our NTSVM is feasible and effective. It should be pointed out that there are four parameters in our NTSVM, so the parameter selection is a practical problem and should be investigated in the future.



(a) **(b)**

(c) **(d)**

Fig. 5 Relationship of kernel parameters σ and classification accuracy

Table 6 Comparison on NDC datasets for linear classifiers

Datasets	TWSVM	TBSVM	NHSVM	NTSVM
	Train / Test (%)	Train / Test (%)	Train / Test (%)	Train / Test (%)
	Time (s)	Time (s)	Time (s)	Time (s)
NDC-500	89.00 / 82.00	88.60 / 84.00	88.80 / 80.00	88.80 / 86.00
	0.1709	0.1137	0.2830	0.1028
NDC-1k	86.50 / 89.00	86.50 / 89.00	86.30 / 85.00	86.30 / 89.00
	0.6854	0.6094	1.0363	0.5910
NDC-2k	87.15 / 82.50	86.90 / 82.50	87.10 / 78.00	86.85 / 83.00
	2.7513	2.5243	7.3447	2.4563
NDC-3k	87.03 / 80.67	86.93 / 81.33	86.87 / 80.67	86.90 / 81.00
	7.9149	7.3595	13.2583	7.2345
NDC-5k	86.02 / 83.20	86.10 / 83.20	85.92 / 84.40	86.14 / 83.20
	25.6839	21.2434	58.4534	20.8603

Table 7 Comparison on NDC datasets for nonlinear classifiers

Datasets	TWSVM	TBSVM	NHSVM	NTSVM
	Train / Test (%)	Train / Test (%)	Train / Test (%)	Train / Test (%)
	Time (s)	Time (s)	Time (s)	Time (s)
NDC-500	100 / 90.00	91.60 / 88.00	90.20 / 84.80	91.60 / 88.00
	0.5279	0.2629	0.3974	0.1643
NDC-1k	91.80 / 89.00	86.60 / 84.00	85.80 / 84.00	86.00 / 89.00
	2.1009	0.7197	0.8276	0.6810
NDC-2k	97.95 / 91.50	88.80 / 85.00	88.55 / 83.50	89.05 / 84.50
	6.1606	2.7652	5.9162	2.3355
NDC-3k	98.08 / 92.33	87.87 / 82.67	88.03 / 84.67	88.97 / 83.33
	12.6107	6.8310	13.2398	5.8870
NDC-5k	99.62 / 95.20	88.56 / 85.00	88.66 / 86.20	88.48 / 85.80
	36.9999	27.4101	49.9451	26.6853

Furthermore, expanding the applications of our NTSVM in practical scenarios and extending our NTSVM to multi-view learning [44] are also interesting.

Acknowledgments This work is supported in part by the National Natural Science Foundation of China under grant No. 61373055 and No. 61103128. The authors would like to thank Dr. Yuan-Hai Shao from Zhejiang University of Technology for his valuable discussion and help.

References

1. Turk M, Pentland A (1991) Eigenfaces for recognition. J Cognit Neurosci 3(1):71–86
2. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720
3. Yang J, Zhang D, Frangi AF, Yang JY (2004) Two-dimensional PCA: a new approach to face representation and recognition. IEEE Trans Pattern Anal Mach Intell 26(1):131–137
4. Tao DC, Li XL, Hu WM et al (2005) Supervised tensor learning. In: Fifth IEEE international conference on data mining
5. Tao DC, Li XL, Wu XD, Maybank SJ (2007) General tensor discriminant analysis and gabor features for gait recognition. IEEE Trans Pattern Anal Mach Intell 29(10):1700–1715
6. Hong CQ, Yu J, Li J, Chen XH (2013) Multi-view hypergraph learning by patch alignment framework. Neurocomputing 118:79–86
7. Yu J, Rui Y, Tang YY, Tao DC (2014) High-order distance-based multiview stochastic learning in image classification. IEEE Trans Cybern 44(12):2431–2442
8. Wang Z, Chen SC, Sun TK (2008) MultiK-MHKS: a novel multiple kernel learning algorithm. IEEE Trans Pattern Anal Mach Intell 30(2):348–353
9. Pong KH, Lam KM (2014) Multi-resolution feature fusion for face recognition. Pattern Recognit 47(2):556–567
10. Cortes C, Vapnik VN (1995) Support vector machine. Mach Learn 20(3):273–297
11. Vapnik VN (2000) The nature of statistical learning theory. Springer, New York
12. Byun H, Lee SW (2002) Applications of support vector machines for pattern recognition: a survey, pattern recognition with support vector machines. Springer, Berlin
13. Burges C (1998) A tutorial support vector machines for pattern recognition. Data Mining Knowl Discov 2:1–43
14. Noble WS (2004) Kernel methods in computational biology. In: Vert J-P (ed) Support vector machine applications in computational biology. MIT Press, Cambridge, pp 71–92

15. Isa D, Lee LH, Kallimani VP, Rajkumar R (2008) Text document preprocessing with the Bayes formula for classification using the support vector machine. IEEE Trans Knowl Data Eng 20(9):1264–1272
16. Yen SJ, Wu YC, Yang JC, Lee YS, Liu LL (2013) A support vector machine-based context-ranking model for question answering. Inf Sci 224(1):77–87
17. Trafails TB, Ince H (2000) Support vector machine for regression and applications to financial forecasting. In: International joint conference on neural networks, pp 6348–6348
18. Tao DC, Tang XO, Li XL, Wu XD (2006) Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. IEEE Trans Pattern Anal Mach Intell 28(7):1088–1099
19. Liu WF, Tao DC (2013) Multiview Hessian regularization for image annotation. IEEE Trans Image Process 22(7):2676–2687
20. Tao DC, Li XL, Wu XD, Maybank SJ (2009) Geometric mean for subspace selection. IEEE Trans Pattern Anal Mach Intell 31(2):260–274
21. Mangasarian OL, Wild EW (2006) Multisurface proximal support vector machine classification via generalized eigenvalues. IEEE Trans Pattern Anal Mach Intell 28(1):69–74
22. Jayadeva R, Khemchandani S Chandra (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
23. Kumar MA, Gopal M (2008) Application of smoothing technique in twin support vector machines. Pattern Recognit Lett 29(13):1842–1848
24. Shao YH, Zhang CH, Wang XB, Deng NY (2011) Improvements on twin support vector machines. IEEE Trans Neural Netw 22(6):962–968
25. Peng XJ (2011) TPMSVM: a novel twin parametric-margin support vector machine for pattern recognition. Pattern Recognit 44(10):2678–2692
26. Qi ZQ, Tian YJ, Shi Y (2013) Robust twin support vector machine for pattern classification. Pattern Recognit 46(1):305–316
27. Qi ZQ, Tian YJ, Shi Y (2013) Structural twin support vector machine for classification. Knowl-Based Syst 43:74–81
28. Wang Z, Shao YH, Wu TR (2014) Proximal parametric-margin support vector classifier and its applications. Neural Comput Appl 24:755–764
29. Shao YH, Chen WJ, Deng NY (2014) Nonparallel hyperplane support vector machine for binary classification problems. Inf Sci 263:22–35
30. Ye QL, Zhao CX, Ye N, Chen YN (2010) Multi-weight vector projection support vector machines. Pattern Recognit Lett 31:2006–2011
31. Chen XB, Yang J, Ye QL, Liang J (2011) Recursive projection twin support vector machine via within-class variance minimization. Pattern Recognit 44(10):2643–2655
32. Shao YH, Deng NY, Yang ZM (2012) Least squares recursive projection twin support vector machine for classification. Pattern Recognit 45(6):2299–2307
33. Shao YH, Wang Z, Chen WJ, Deng NY (2013) A regularization for the projection twin support vector machine. Knowl-Based Syst 37:203–210
34. Ding SF, Hua XP (2014) Recursive least squares projection twin support vector machines for nonlinear classification. Neurocomputing 130:3–9
35. Peng XJ (2010) Least squares twin support vector hypersphere (LS-TSVH) for pattern recognition. Expert Syst Appl 37(12):8371–8378
36. Peng XJ, Xu D (2014) Twin support vector hypersphere (TSVH) classifier for pattern recognition. Neural Comput Appl 24:1207–1220
37. Fung G, Mangasarian OL (2001) Proximal support vector machine classifiers. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 77–86
38. Mangasarian OL, Musicant DR (1999) Successive overrelaxation for support vector machines. IEEE Trans Neural Netw 10(5):1032–1037
39. Lee YJ, Huang SY (2007) Reduced support vector machines: a statistical theory. IEEE Trans Neural Netw 13(1):1–13
40. Duda RO, Hart PE, Stork DG (2001) Pattern classification, 2nd edn. Wiley, New York
41. Ripley BD (2008) Pattern recognition and neural networks. Cambridge University Press, Cambridge
42. Muphy PM, Aha DW (1992) UCI repository of machine learning databases
43. Musicant DR (1998) NDC: normally distributed clustered datasets. http://www.cs.wisc.edu/dmi/svm/ndc/. Accessed 5 Nov 2015
44. Xie XJ, Sun SL (2014) Multi-view Laplacian twin support vector machines. Appl Intell 41(4):1059–1068