Appl Intell (2018) 48:1041–1053
DOI 10.1007/s10489-017-0984-2

An improved ν-twin bounded support vector machine

Huiru Wang[1] · Zhijian Zhou[1] · Yitian Xu[1]

Published online: 15 August 2017

Abstract In this paper, we propose a new classifier termed as an improved ν-twin bounded support vector machine (Iν-TBSVM) which is motivated by ν-twin support vector machine (ν-TSVM). Similar to the ν-TSVM, Iν-TBSVM determines two nonparallel hyperplanes such that they are closer to their respective classes and are at least ρ_+ or ρ_- distance away from the other class. The significant advantage of Iν-TBSVM over ν-TSVM is that Iν-TBSVM skillfully avoids the expensive matrix inverse operation when solving the dual problems. Therefore, the proposed classifier is more effective when dealing with large scale problem and has comparable generalization ability. Iν-TBSVM also implements structural risk minimization principle by introducing a regularization term into the objective function. More importantly, the kernel trick can be applied directly to the Iν-TBSVM for nonlinear case, so the nonlinear Iν-TBSVM is superior to the nonlinear ν-TSVM theoretically. In addition, we also prove that ν-SVM is the special case of Iν-TBSVM. The property of parameters in Iν-TBSVM is discussed and testified by two artificial experiments. Numerical experiments on twenty-two benchmarking datasets are performed to investigate the validity of our proposed algorithm in both linear case and nonlinear case. Experimental results show the effectiveness of our proposed algorithm.

Keywords ν-TSVMs · Structural risk minimization · Matrix inverse operation · Classification

1 Introduction

The support vector machine (SVM) [1], motivated by Vapnik's statistical learning theory, is the state of the art algorithm in nonlinear pattern classification, together with multilayered neural network models. Compared with other machine learning methods like artificial neural networks[2], SVM gains many advantages. SVM solves a quadratic programming problem (QPP) which assures that once an optimal solution is obtained, it is the unique global solution. SVM implements the structural risk minimization principle rather than the empirical risk minimization principle, which minimizes the upper bound of the generalization error. The introduction of the kernel function [3] in SVM maps training vectors into a high-dimensional space directly. On the basis of the techniques above, SVM has been successfully applied in many fields [4–8] and various amendments have been suggested [9–17].

It is well known that solving the entire QPP in the SVMs is time consuming, which still remains challenging. To improve the computational speed, Jayadeva et al. have proposed a twin support vector machine (TSVM) [18] for the binary classification data. TSVM attempts to seek two nonparallel proximal hyperplanes for the two classes of samples, such that each hyperplane is closer to one class and as far as possible from the other. The greatest advantage of TSVM over SVM is that it solves two smaller-sized QPPs rather than a single large one,

✉ Zhijian Zhou
zhijianzh@163.com

[1] College of Science, China Agricultural University,
No.17 Qinghua East Road, 100083 Haidian, Beijing, China

which makes TSVM work faster than SVM. Later, many variants of TSVM have been proposed [19], such as smooth TSVM [20], least squares TSVM [21], twin support vector regression [22], twin bounded SVM (TBSVM) [23], and structure information based TSVM [24]. Tian et al. proposed a new improved TSVM [25] to avoid solving the corresponding inverse matrices in most of the existing TSVMs.

In TSVM, the patterns of one class are at least a unit distance away from the hyperplane of other class, this may increase the number of support vectors thus leading to poor generalization ability. Recently, Peng proposed ν-TSVM [26]. The unit distance of TSVM is modified to variable ρ which needs to be optimized in the primal problem. And the parameter ν in ν-TSVM is used to control the bounds on the fractions of support vectors and error margins. Besides, ν-TSVM can be interpreted as a pair of minimum generalized Mahalanobis-norm problems on two reduced convex hulls. Based on the thoughts of ν-TSVM, many algorithms are studied extensively, such as ν-twin bounded SVM (ν-TBSVM) [27], rough margin based ν-TSVM [28, 29] and a novel improved ν-TSVM [30]. Nevertheless, the aforementioned ν-TSVMs all involve expensive matrix inverse operation, which makes them time consuming even though the Sherman-Morrison-Woodbury [31–33] for matrix inversion can be used. When using the linear kernel, the ν-TSVMs cannot transform to the linear case. However, our algorithm can achieve it.

In this paper, we present an improvement on ν-TSVM, called the improved ν-twin bounded support vector machine (Iν-TBSVM). Our Iν-TBSVM possesses the following attractive advantages:

1. Iν-TBSVM leads to less computation time because it skillfully avoids the matrix inverse operation when solving dual QPPs. Therefore, it is more suitable to solve large scale problems.
2. Unlike the ν-TSVM, when using the linear kernel, the Iν-TBSVM can degenerate to the linear case.
3. Similar to ν-TBSVM, the structural risk is minimized by introducing a regularization term in the objective function in Iν-TBSVM, which makes sure the enhanced algorithm yields high testing accuracy.

The remainder of the paper is organized as follows. In Section 2 we give a brief overview on ν-TSVM and ν-TBSVM. In Section 3 we introduce our Iν-TBSVM in detail. In Section 4, we compare our Iν-TBSVM with ν-TBSVM and TBSVM. Section 5 performs two artificial experiments to verify the property of parameters in Iν-TBSVM, and twenty-two experiments to investigate the effectiveness of our proposed algorithm. We make conclusions in Section 6.

2 Related works

In this section, we review the basics of ν-TSVM and ν-TBSVM and summarize their drawbacks.

2.1 ν-twin support vector machine

Consider a binary classification problems with p samples belonging to class $+1$ and q samples belonging to class -1 in the n-dimensional real space $\mathbb{R}^n$. Let matrix $\mathbf{A} \in \mathbb{R}^{p\times n}, \mathbf{B} \in \mathbb{R}^{q\times n}$ stand for the positive and negative samples, respectively.

The ν-TSVM [26] generates two nonparallel hyperplane instead of a single one as in the standard SVM. The two nonparallel hyperplanes are obtained by solving two smaller-sized QPPs rather than a single large one. The ν-TSVM seeks the following pair of nonparallel hyperplanes:

$$\langle \mathbf{w}_+, \mathbf{x}\rangle + b_+ = 0 \quad \text{and} \quad \langle \mathbf{w}_-, \mathbf{x}\rangle + b_- = 0 \tag{1}$$

for linear case, and it seeks the following kernel surfaces:

$$K(\mathbf{x}^T, \mathbf{C}^T)\mathbf{w}_+ + b_+ = 0 \text{ and } K(\mathbf{x}^T, \mathbf{C}^T)\mathbf{w}_- + b_- = 0 \tag{2}$$

for nonlinear case, where $\mathbf{C} = [\mathbf{A}^T\ \mathbf{B}^T] \in \mathbb{R}^{n\times(p+q)}$, such that each hyperplane is closer to one class and is as far as possible from the other.

The dual problems of ν-TSVM are as follows:

$$\begin{aligned} &\max_{\boldsymbol{\alpha}}\ -\frac{1}{2}\boldsymbol{\alpha}^T\mathbf{G}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{G}^T\boldsymbol{\alpha} \\ &s.t.\quad \mathbf{0} \le \boldsymbol{\alpha} \le \frac{1}{q}\mathbf{e}_-, \\ &\qquad\ \mathbf{e}_-^T\boldsymbol{\alpha} \ge \nu_1, \end{aligned} \tag{3}$$

and

$$\begin{aligned} &\max_{\boldsymbol{\gamma}}\ -\frac{1}{2}\boldsymbol{\gamma}^T\mathbf{H}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{H}^T\boldsymbol{\gamma} \\ &s.t.\quad \mathbf{0} \le \boldsymbol{\gamma} \le \frac{1}{p}\mathbf{e}_+, \\ &\qquad\ \mathbf{e}_+^T\boldsymbol{\gamma} \ge \nu_2, \end{aligned} \tag{4}$$

where $\mathbf{G} = [\mathbf{B}\ \mathbf{e}_-]$ and $\mathbf{H} = [\mathbf{A}\ \mathbf{e}_+]$ for linear case, and $\mathbf{G} = [K(\mathbf{B}, \mathbf{C}^T)\ \mathbf{e}_-]$ and $\mathbf{H} = [(\mathbf{A}, \mathbf{C}^T)\ \mathbf{e}_+]$ for nonlinear case.

A new sample $\mathbf{x}$ is assigned to a class i $(i = +1, -1)$ by

$$class\ i = \arg\min_{i=+,-} \frac{|\langle \mathbf{w}_i, \mathbf{x}\rangle + b_i|}{\|\mathbf{w}_i\|}, \tag{5}$$

where $|\cdot|$ is the perpendicular distance of the new sample $\mathbf{x}$ from the two hyperplanes (1).

2.2 ν-twin bounded support vector machine

Xu and Guo [27] introduced a regularization term into the objective function in the ν-TBSVM. The linear QPPs of ν-TBSVM are denoted as follows,

$$\begin{aligned}\min_{\mathbf{w}_+,b_+,\rho_+,\boldsymbol{\xi}_-} \ & \frac{c_1}{2}(||\mathbf{w}_+||^2+b_+^2)+\frac{1}{2}\|\mathbf{A}\mathbf{w}_+ +\mathbf{e}_+b_+\|^2 \\ & -\nu_1\rho_+ + \frac{1}{q}\mathbf{e}_-^T\boldsymbol{\xi}_- \\ s.t. \ & -(\mathbf{B}\mathbf{w}_+ +\mathbf{e}_-b_+)\geq \rho_+\mathbf{e}_- - \boldsymbol{\xi}_-, \\ & \rho_+\geq 0, \boldsymbol{\xi}_-\geq \mathbf{0},\end{aligned} \tag{6}$$

and

$$\begin{aligned}\min_{\mathbf{w}_-,b_-,\rho_-,\boldsymbol{\xi}_+} \ & \frac{c_2}{2}(||\mathbf{w}_-||^2+b_-^2)+\frac{1}{2}\|\mathbf{B}\mathbf{w}_- +\mathbf{e}_-b_-\|^2 \\ & -\nu_2\rho_- + \frac{1}{p}\mathbf{e}_+^T\boldsymbol{\xi}_+ \\ s.t. \ & \mathbf{A}\mathbf{w}_- +\mathbf{e}_+b_-\geq \rho_-\mathbf{e}_+ - \boldsymbol{\xi}_+, \\ & \rho_-\geq 0, \boldsymbol{\xi}_+\geq \mathbf{0}.\end{aligned} \tag{7}$$

By introducing the Lagrange multipliers $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$, their dual problems are derived as,

$$\begin{aligned}\max_{\boldsymbol{\alpha}} \ & -\frac{1}{2}\boldsymbol{\alpha}^T\mathbf{G}(\mathbf{H}^T\mathbf{H}+c_1\mathbf{I})^{-1}\mathbf{G}^T\boldsymbol{\alpha} \\ s.t. \ & \mathbf{0}\leq\boldsymbol{\alpha}\leq\frac{1}{q}\mathbf{e}_-, \\ & \mathbf{e}_-^T\boldsymbol{\alpha}\geq\nu_1,\end{aligned} \tag{8}$$

and

$$\begin{aligned}\max_{\boldsymbol{\gamma}} \ & -\frac{1}{2}\boldsymbol{\gamma}^T\mathbf{H}(\mathbf{G}^T\mathbf{G}+c_2\mathbf{I})^{-1}\mathbf{H}^T\boldsymbol{\gamma} \\ s.t. \ & \mathbf{0}\leq\boldsymbol{\gamma}\leq\frac{1}{p}\mathbf{e}_+, \\ & \mathbf{e}_+^T\boldsymbol{\gamma}\geq\nu_2,\end{aligned} \tag{9}$$

where $\mathbf{G}=[\mathbf{B}\ \mathbf{e}_-]$ and $\mathbf{H}=[\mathbf{A}\ \mathbf{e}_+]$.

From the dual QPPs (3), (4), (8) and (9), we can clearly see that ν-TSVM and ν-TBSVM inevitably need to calculate the inverse matrices $\mathbf{H}^T\mathbf{H}$ and $\mathbf{G}^T\mathbf{G}$, $\mathbf{H}^T\mathbf{H}+c_1\mathbf{I}$ and $\mathbf{G}^T\mathbf{G}+c_2\mathbf{I}$, respectively, which is time consuming. Besides, the size of the inverse matrices is roughly $(n+1)\times(n+1)$ which means the linear ν-TSVM and ν-TBSVM only works on smaller n. Therefore, they are not suitable for, and have difficulty calculating, the high dimensional datasets. For the nonlinear case, the size is $(l+1)\times(l+1)$, which means they only work for the problem with smaller scale. And the nonlinear ν-TSVM and ν-TBSVM with the linear kernel is not equivalent to the linear ν-TSVM and ν-TBSVM, respectively. Here we illustrate it by a toy example, see Fig. 1.

3 An improved ν-twin bounded support vector machine

In this section, we propose a new algorithm based on ν-TSVM, termed as improved ν-twin bounded support vector machine (Iν-TBSVM), which inherits the essence of the traditional ν-SVM and overcomes the aforementioned drawbacks in ν-TSVM and ν-TBSVM. Further speaking, Iν-TBSVM constructs two nonparallel hyperplanes by implementing the structural risk minimization principle. Due to the introduction of two new variables, the Iν-TBSVM does not require the expensive matrix inverse operation and the nonlinear case can degenerate to the linear case directly when linear kernel is applied.

3.1 Linear case

For the linear case, the Iν-TBSVM finds the following QPPs.

$$\begin{aligned}\min_{\mathbf{w}_+,b_+,\boldsymbol{\eta}_+,\rho_+,\boldsymbol{\xi}_-} \ & \frac{c_1}{2}(||\mathbf{w}_+||^2+b_+^2)+\frac{1}{2}\boldsymbol{\eta}_+^T\boldsymbol{\eta}_+ \\ & -\nu_1\rho_+ + \frac{1}{q}\mathbf{e}_-^T\boldsymbol{\xi}_- \\ s.t. \ & \mathbf{A}\mathbf{w}_+ +\mathbf{e}_+b_+ = \boldsymbol{\eta}_+, \\ & -(\mathbf{B}\mathbf{w}_+ +\mathbf{e}_-b_+)\geq \rho_+\mathbf{e}_- - \boldsymbol{\xi}_-, \\ & \boldsymbol{\xi}_-\geq \mathbf{0},\ \rho_+\geq 0,\end{aligned} \tag{10}$$

and

$$\begin{aligned}\min_{\mathbf{w}_-,b_-,\boldsymbol{\eta}_-,\rho_-,\boldsymbol{\xi}_+} \ & \frac{c_2}{2}(||\mathbf{w}_-||^2+b_-^2)+\frac{1}{2}\boldsymbol{\eta}_-^T\boldsymbol{\eta}_- \\ & -\nu_2\rho_- + \frac{1}{p}\mathbf{e}_+^T\boldsymbol{\xi}_+ \\ s.t. \ & \mathbf{B}\mathbf{w}_- +\mathbf{e}_-b_- = \boldsymbol{\eta}_-, \\ & \mathbf{A}\mathbf{w}_- +\mathbf{e}_+b_-\geq \rho_-\mathbf{e}_+ - \boldsymbol{\xi}_+, \\ & \boldsymbol{\xi}_+\geq \mathbf{0},\ \rho_-\geq 0,\end{aligned} \tag{11}$$

where c_1, c_2 are the penalty parameters; $\boldsymbol{\xi}_+$, $\boldsymbol{\xi}_-$ are slack vectors; $\mathbf{e}_+, \mathbf{e}_-$ are vectors of ones of appropriate dimensions; $\boldsymbol{\eta}_+$, $\boldsymbol{\eta}_-$ are vectors of appropriate dimensions; ν_1, ν_2 are chosen a priori; ρ_+, ρ_- are additional variables.

It is worth noting that we only introduce two parameters $\boldsymbol{\eta}_+$ and $\boldsymbol{\eta}_-$ in (10) and (11) compared with QPPs (6) and (7). For the sake of conciseness, we only take the dual problem of (10) into account.

The first term in the objective function is the regularization term. That means our Iν-TBSVM still takes the structural risk minimization principle into consideration to improve the generalization ability. The distance between the hyperplane $\langle\mathbf{w}_+,\mathbf{x}\rangle+b_+=0$ and the boundary hyperplane $\langle\mathbf{w}_+,\mathbf{x}\rangle+b_+=-\rho_+$ can be measured by $\rho_+/\sqrt{||\mathbf{w}_+||^2+b_+^2}$, which is maximized by minimization

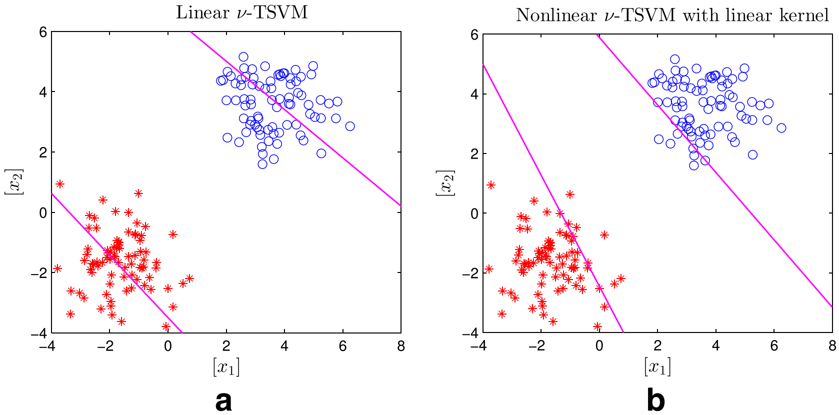

Fig. 1 Nonlinear ν-TSVM with the linear kernel is not equivalent to the linear ν-TSVM. The different points "*" and "o" are generated following two normal distributions respectively. Here the parameter $\nu_1 = \nu_2 = 0.5$. **a** Two nonparallel hyperplanes obtained from the linear ν-TSVM; **b** Two nonparallel hyperplanes obtained from nonlinear ν-TSVM with linear kernel

of $\frac{c_1}{2}(||\mathbf{w}_+||^2 + b_+^2) - \nu_1\rho_+$. It implies that the distance between two parallel hyperplanes is as large as possible.

The Lagrangian function corresponding to Iν-TBSVM (10) is as follows,

$$
\begin{aligned}
L = &\frac{c_1}{2}(||\mathbf{w}_+||^2 + b_+^2) + \frac{1}{2}\boldsymbol{\eta}_+^T\boldsymbol{\eta}_+ - \nu_1\rho_+ + \frac{1}{q}\mathbf{e}_-^T\boldsymbol{\xi}_- \\
&+\boldsymbol{\lambda}^T(\mathbf{A}\mathbf{w}_+ + \mathbf{e}_+ b_+ - \boldsymbol{\eta}_+) - \boldsymbol{\beta}^T\boldsymbol{\xi}_- - s\rho_+ \\
&+\boldsymbol{\alpha}^T(\mathbf{B}\mathbf{w}_+ + \mathbf{e}_- b_+ + \rho_+\mathbf{e}_- - \boldsymbol{\xi}_-), \qquad (12)
\end{aligned}
$$

where $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_q)^T$, $\boldsymbol{\beta} = (\beta_1, ..., \beta_q)^T$, $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_p)^T$ and s are Lagrange multipliers. Differentiating the Lagrangian function L with respect to variables $\mathbf{w}_+$, b_+, $\boldsymbol{\eta}_+$, ρ_+, $\boldsymbol{\xi}_-$ yields the following Karush-Kuhn-Tucker (KKT) conditions:

$$\frac{\partial L}{\partial \mathbf{w}_+} = c_1\mathbf{w}_+ + \mathbf{A}^T\boldsymbol{\lambda} + \mathbf{B}^T\boldsymbol{\alpha} = 0, \qquad (13)$$

$$\frac{\partial L}{\partial b_+} = c_1 b_+ + \mathbf{e}_+^T\boldsymbol{\lambda} + \mathbf{e}_-^T\boldsymbol{\alpha} = 0, \qquad (14)$$

$$\frac{\partial L}{\partial \boldsymbol{\eta}_+} = \boldsymbol{\eta}_+ - \boldsymbol{\lambda} = 0, \qquad (15)$$

$$\frac{\partial L}{\partial \rho_+} = \mathbf{e}_-^T\boldsymbol{\alpha} - \nu_1 - s = 0, \qquad (16)$$

$$\frac{\partial L}{\partial \boldsymbol{\xi}_-} = \frac{1}{q}\mathbf{e}_- - \boldsymbol{\alpha} - \boldsymbol{\beta} = 0, \qquad (17)$$

$$\mathbf{A}\mathbf{w}_+ + \mathbf{e}_+ b_+ = \boldsymbol{\eta}_+, \qquad (18)$$

$$-(\mathbf{B}\mathbf{w}_+ + \mathbf{e}_- b_+) \geq \rho_+\mathbf{e}_- - \boldsymbol{\xi}_-,\ \boldsymbol{\xi}_- \geq \mathbf{0},\ \rho_+ \geq 0, \qquad (19)$$

$$\boldsymbol{\alpha}^T(\mathbf{B}\mathbf{w}_+ + \mathbf{e}_- b_+ + \rho_+\mathbf{e}_- - \boldsymbol{\xi}_-) = 0, \boldsymbol{\alpha} \geq \mathbf{0}, \qquad (20)$$

$$\boldsymbol{\beta}^T\boldsymbol{\xi}_- = 0, \boldsymbol{\beta} \geq \mathbf{0}, \qquad (21)$$

$$s\rho_+ = 0, s \geq 0. \qquad (22)$$

Since $\boldsymbol{\beta} \geq \mathbf{0}$, from (17) we have

$$\mathbf{0} \leq \boldsymbol{\alpha} \leq \frac{1}{q}\mathbf{e}_-. \qquad (23)$$

Since $s > 0$, from (16) we also get

$$\mathbf{e}_-^T\boldsymbol{\alpha} \geq \nu_1. \qquad (24)$$

Equations (13) and (14) imply that

$$\mathbf{w}_+ = -\frac{1}{c_1}(\mathbf{A}^T\boldsymbol{\lambda} + \mathbf{B}^T\boldsymbol{\alpha}), \qquad (25)$$

$$b_+ = -\frac{1}{c_1}(\mathbf{e}_+^T\boldsymbol{\lambda} + \mathbf{e}_-^T\boldsymbol{\alpha}). \qquad (26)$$

Using (13), (14) and (15), we derive the dual formulation of the QPP (10) as follows,

$$
\begin{aligned}
&\max_{\boldsymbol{\lambda},\boldsymbol{\alpha}} \ -\frac{1}{2}(\boldsymbol{\lambda}^T\boldsymbol{\alpha}^T)\boldsymbol{Q}(\boldsymbol{\lambda}^T\boldsymbol{\alpha}^T)^T \\
&s.t. \quad \mathbf{0} \leq \boldsymbol{\alpha} \leq \frac{1}{q}\mathbf{e}_-, \qquad (27) \\
&\qquad\ \mathbf{e}_-^T\boldsymbol{\alpha} \geq \nu_1,
\end{aligned}
$$

where

$$\boldsymbol{Q} = \begin{pmatrix} \mathbf{AA}^T + c_1\mathbf{I} & \mathbf{AB}^T \\ \mathbf{BA}^T & \mathbf{BB}^T \end{pmatrix} + \mathbf{E}, \qquad (28)$$

and $\mathbf{I}$ is the $p \times p$ identity matrix, $\mathbf{E}$ is the $(p+q) \times (p+q)$ matrix with all entries equal to 1.

Similarly, the dual formulation of (11) is derived as

$$
\begin{aligned}
&\max_{\boldsymbol{\theta},\boldsymbol{\gamma}} \ -\frac{1}{2}(\boldsymbol{\theta}^T\boldsymbol{\gamma}^T)\tilde{\boldsymbol{Q}}(\boldsymbol{\theta}^T\boldsymbol{\gamma}^T)^T \\
&s.t. \quad \mathbf{0} \leq \boldsymbol{\gamma} \leq \frac{1}{p}\mathbf{e}_+, \qquad (29) \\
&\qquad\ \mathbf{e}_+^T\boldsymbol{\gamma} \geq \nu_2,
\end{aligned}
$$

where

$$\tilde{\boldsymbol{Q}} = \begin{pmatrix} \mathbf{BB}^T + c_2\mathbf{I} & \mathbf{BA}^T \\ \mathbf{AB}^T & \mathbf{AA}^T \end{pmatrix} + \mathbf{E}, \qquad (30)$$

and $\mathbf{I}$ is the $q \times q$ identity matrix, $\mathbf{E}$ is the $(p+q) \times (p+q)$ matrix with all entries equal to 1. Once the optimal solutions $(\boldsymbol{\theta}, \boldsymbol{\gamma})$ are obtained, we can derive

$$\mathbf{w}_- = -\frac{1}{c_2}(\mathbf{B}^T\boldsymbol{\theta} + \mathbf{A}^T\boldsymbol{\gamma}), \qquad (31)$$

$$b_- = -\frac{1}{c_2}(\mathbf{e}_-^T\boldsymbol{\theta} + \mathbf{e}_+^T\boldsymbol{\gamma}). \qquad (32)$$

To compute $\rho_{\pm}$, we choose the samples $\mathbf{x}_i$ (or $\mathbf{x}_j$)with $0 < \alpha_i < \frac{1}{p}$ (or $0 < \alpha_j < \frac{1}{q}$, which means $\xi_i = 0$ (or $\xi_j = 0$) and $\mathbf{w}_-^T\mathbf{x}_i + b_- = \rho_-$ (or $\mathbf{w}_+^T\mathbf{x}_j + b_+ = -\rho_+$) according to the KKT conditions. Then $\rho_{\pm}$ can be calculated by

$$\rho_+ = -\frac{1}{q_1}\sum_{j=1}^{q_1}(\mathbf{w}_+^T\mathbf{x}_j + b_+), \quad \rho_- = \frac{1}{p_1}\sum_{i=1}^{p_1}(\mathbf{w}_-^T\mathbf{x}_i + b_-). \quad (33)$$

There is similar conclusion for the QPPs (27) and QPP (29) above, so we only take QPP (27) into account. For convenience, we first give an equivalent formulation of the QPP (27). The optimal parameters ρ_+ in the QPP (10) is actually larger than zero. On the conditions above, we give the following Proposition 1.

Proposition 1 *The QPP* (27) *can be transformed into the following QPP.*

$$\begin{aligned} &\max_{\boldsymbol{\lambda},\boldsymbol{\alpha}} \ -\frac{1}{2}(\boldsymbol{\lambda}^T\boldsymbol{\alpha}^T)\boldsymbol{Q}(\boldsymbol{\lambda}^T\boldsymbol{\alpha}^T)^T \\ &s.t. \quad \mathbf{0} \le \boldsymbol{\alpha} \le \frac{1}{q}\mathbf{e}_-, \\ &\qquad\ \mathbf{e}_-^T\boldsymbol{\alpha} = \nu_1. \end{aligned} \quad (34)$$

The QPPs (27) *and* (34) *differ in the second constraint condition. In* (27)*, the second inequality constraint* $\mathbf{e}_-^T\boldsymbol{\alpha} \ge \nu_1$ *can become an equality constraint* $\mathbf{e}_-^T\boldsymbol{\alpha} = \nu_1$.

Proof According to assumptions $\rho_+ > 0$ and the KKT condition $s\rho_+ = 0$, we can obtain that the Lagrangian multipliers $s = 0$. Then, by substituting it into (16), we can get the equality constraint $\mathbf{e}_-^T\boldsymbol{\alpha} = \nu_1$. □

As in ν-SVM, ν-TSVM and ν-TBSVM, parameter ν in our Iν-TBSVM has its property. We discuss its property in the following Propositions.

Proposition 2 *Denote by* q_2 *the number of support vectors in the negative class. Then we can obtain an inequality* $\nu_1 \le \frac{q_2}{q}$*, which implies that* ν_1 *is a lower bound on the fraction of support vectors in the negative class.*

Proposition 3 *Denote by* p_2 *the number of boundary errors in the negative class. Then we can obtain an inequality* $\nu_1 \ge \frac{p_2}{q}$*, which implies that* ν_1 *is an upper bound on the fraction of boundary errors in the negative class.*

Proof The proof of Proposition 2 and 3 is similar to that of Proposition 2 and 3 in [27]. These results can be extended to the nonlinear case by considering kernel function. □

It is worthwhile to mention that the dual problems (27) and (29) don't involve the matrix inverse operation according the expression of $\boldsymbol{Q}$ and $\tilde{\boldsymbol{Q}}$. More importantly, they can be easily extended to the nonlinear case.

3.2 Nonlinear case

By introducing the kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = (\varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j))$ and the corresponding transformation $\mathbf{x} = \varphi(\mathbf{x})$, where $\mathbf{x} \in \mathcal{H}$, $\mathcal{H}$ is the Hilbert space, we can get the nonlinear Iν-TBSVM as follows,

$$\begin{aligned} \min_{\mathbf{w}_+, b_+, \boldsymbol{\eta}_+, \rho_+, \boldsymbol{\xi}_-} \ & \frac{c_1}{2}(||\mathbf{w}_+||^2 + b_+^2) + \frac{1}{2}\boldsymbol{\eta}_+^T\boldsymbol{\eta}_+ \\ & -\nu_1\rho_+ + \frac{1}{q}\mathbf{e}_-^T\boldsymbol{\xi}_- \\ s.t. \quad & \varphi(\mathbf{A})\mathbf{w}_+ + \mathbf{e}_+ b_+ = \boldsymbol{\eta}_+, \\ & -(\varphi(\mathbf{B})\mathbf{w}_+ + \mathbf{e}_- b_+) \ge \rho_+\mathbf{e}_- - \boldsymbol{\xi}_-, \\ & \boldsymbol{\xi}_- \ge \mathbf{0},\ \rho_+ \ge 0, \end{aligned} \quad (35)$$

and

$$\begin{aligned} \min_{\mathbf{w}_-, b_-, \boldsymbol{\eta}_-, \rho_-, \boldsymbol{\xi}_+} \ & \frac{c_2}{2}(||\mathbf{w}_-||^2 + b_-^2) + \frac{1}{2}\boldsymbol{\eta}_-^T\boldsymbol{\eta}_- \\ & -\nu_2\rho_- + \frac{1}{p}\mathbf{e}_+^T\boldsymbol{\xi}_+ \\ s.t. \quad & \varphi(\mathbf{B})\mathbf{w}_- + \mathbf{e}_- b_- = \boldsymbol{\eta}_-, \\ & \varphi(\mathbf{A})\mathbf{w}_- + \mathbf{e}_+ b_- \ge \rho_-\mathbf{e}_+ - \boldsymbol{\xi}_+, \\ & \boldsymbol{\xi}_+ \ge \mathbf{0},\ \rho_- \ge 0, \end{aligned} \quad (36)$$

where c_1, c_2, ν_1, ν_2 are chosen a priori; $\boldsymbol{\xi}_+$, $\boldsymbol{\xi}_-$ are slack vectors; $\mathbf{e}_+$, $\mathbf{e}_-$ are vectors of ones of appropriate dimensions; $\boldsymbol{\eta}_+$, $\boldsymbol{\eta}_-$ are vectors of appropriate dimensions.

In an exactly similar way as the linear case, we derive the dual formulation of (35) as follows,

$$\begin{aligned} &\max_{\boldsymbol{\lambda},\boldsymbol{\alpha}} \ -\frac{1}{2}(\boldsymbol{\lambda}^T\boldsymbol{\alpha}^T)\boldsymbol{Q}_{\varphi}(\boldsymbol{\lambda}^T\boldsymbol{\alpha}^T)^T \\ &s.t. \quad \mathbf{0} \le \boldsymbol{\alpha} \le \frac{1}{q}\mathbf{e}_-, \\ &\qquad\ \mathbf{e}_-^T\boldsymbol{\alpha} \ge \nu_1, \end{aligned} \quad (37)$$

where

$$\boldsymbol{Q}_{\varphi} = \begin{pmatrix} K(\mathbf{A}, \mathbf{A}) + c_1\mathbf{I} & K(\mathbf{A}, \mathbf{B}) \\ K(\mathbf{B}, \mathbf{A}) & K(\mathbf{B}, \mathbf{B}) \end{pmatrix} + \mathbf{E}. \quad (38)$$

Similarly, the dual of the QPP (36) is derived as

$$\begin{aligned} &\max_{\boldsymbol{\theta},\boldsymbol{\gamma}} \ -\frac{1}{2}(\boldsymbol{\theta}^T\boldsymbol{\gamma}^T)\tilde{\boldsymbol{Q}}_{\varphi}(\boldsymbol{\theta}^T\boldsymbol{\gamma}^T)^T \\ &s.t. \quad \mathbf{0} \le \boldsymbol{\gamma} \le \frac{1}{p}\mathbf{e}_+, \\ &\qquad\ \mathbf{e}_+^T\boldsymbol{\gamma} \ge \nu_2, \end{aligned} \quad (39)$$

where

$$\tilde{Q}_\varphi = \begin{pmatrix} K(\mathbf{B},\mathbf{B})+c_2\mathbf{I} & K(\mathbf{B},\mathbf{A}) \\ K(\mathbf{A},\mathbf{B}) & K(\mathbf{A},\mathbf{A}) \end{pmatrix} + \mathbf{E}. \tag{40}$$

Once the optimal solutions $(\boldsymbol{\lambda}, \boldsymbol{\alpha})$ and $(\boldsymbol{\theta}, \boldsymbol{\gamma})$ are obtained, the pair of nonparallel hyperplanes in the Hilbert space can be obtained as follows,

$$K(\mathbf{x}^T, \mathbf{A})\boldsymbol{\lambda} + K(\mathbf{x}^T, \mathbf{B})\boldsymbol{\alpha} + b_+ = 0, \tag{41}$$

where $b_+ = \mathbf{e}_+^T\boldsymbol{\lambda} + \mathbf{e}_-^T\boldsymbol{\alpha}$, and

$$K(\mathbf{x}^T, \mathbf{B})\boldsymbol{\theta} + K(\mathbf{x}^T, \mathbf{A})\boldsymbol{\gamma} + b_- = 0, \tag{42}$$

where $b_- = \mathbf{e}_-^T\boldsymbol{\theta} + \mathbf{e}_+^T\boldsymbol{\gamma}$.

Obviously, the dual QPPs (37) and (39) don't involve the matrix inverse operation, and they can degenerate to the problems (27) and (29) of linear Iν-TBSVM when the linear kernel is applied.

The flowchart of the nonlinear Iν-TBSVM is described as follows.

Algorithm 1 Denote p positive samples represented by matrix $\mathbf{A}$ and q negative samples represented by matrix $\mathbf{B}$. The nonlinear **I$\boldsymbol{\nu}$**-TBSVM can be obtained by the following steps:

Step1: Choose a kernel function K, penalty parameters c_1, c_2 and parameters $\nu_1, \nu_2 \in (0, 1)$.
Step2: Solve the QPPs (37) and (39) respectively, get the optimal solutions $(\boldsymbol{\lambda}, \boldsymbol{\alpha})$ and $(\boldsymbol{\theta}, \boldsymbol{\gamma})$.
Step3: Construct two decision functions $f_+(\mathbf{x}) = K(\mathbf{x}^T, \mathbf{A})\boldsymbol{\lambda} + K(\mathbf{x}^T, \mathbf{B})\boldsymbol{\alpha} + b_+$ and $f_-(\mathbf{x}) = K(\mathbf{x}^T, \mathbf{B})\boldsymbol{\theta} + K(\mathbf{x}^T, \mathbf{A})\boldsymbol{\gamma} + b_-$.
Step4: A new sample $\mathbf{x} \in \mathbb{R}^n$ is assigned to class $i (i = +1, -1)$ by $class\ i = \arg\min_{i=+,-} |f_i(\mathbf{x})|$, where $|\cdot|$ is the perpendicular distance of the new sample $\mathbf{x}$ from the two hyperplanes $f_\pm(\mathbf{x}) = 0$.

3.3 Analysis of algorithm

In this section, we summarize the superiorities of the proposed Iν-TBSVM.

1) By introducing a new pair of variables in ν-TBSVM, our Iν-TBSVM skillfully avoids the matrix inverse operation while it is inescapable in other ν-TSVMs.
2) When using the linear kernel, our Iν-TBSVM can degenerate to the linear case directly, while it is really not so in other ν-TSVMs.
3) Our Iν-TBSVM can naturally and reasonably explain the regularization terms for both linear and nonlinear cases.
4) ν-SVM is the special case of Iν-TBSVM. Let us combine QPP (10) and (11) together to be the following problem,

$$\begin{aligned}
\min\ & \frac{c_1}{2}(||\mathbf{w}_+||^2 + b_+^2) + \frac{c_2}{2}(||\mathbf{w}_-||^2 + b_-^2) \\
& + \frac{1}{2}(\boldsymbol{\eta}_+^T\boldsymbol{\eta}_+ + \boldsymbol{\eta}_-^T\boldsymbol{\eta}_-) - \nu_1\rho_+ - \nu_2\rho_- + \frac{1}{q}\mathbf{e}_-^T\boldsymbol{\xi}_- + \frac{1}{p}\mathbf{e}_+^T\boldsymbol{\xi}_+ \\
s.t.\ \ & \mathbf{A}\mathbf{w}_+ + \mathbf{e}_+b_+ = \boldsymbol{\eta}_+, \\
& \mathbf{B}\mathbf{w}_- + \mathbf{e}_-b_- = \boldsymbol{\eta}_-, \\
& -(\mathbf{B}\mathbf{w}_+ + \mathbf{e}_-b_+) \geq \rho_+\mathbf{e}_- - \boldsymbol{\xi}_-, \\
& \mathbf{A}\mathbf{w}_- + \mathbf{e}_+b_- \geq \rho_-\mathbf{e}_+ - \boldsymbol{\xi}_+, \\
& \boldsymbol{\xi}_- \geq \mathbf{0},\ \rho_+ \geq 0, \\
& \boldsymbol{\xi}_+ \geq \mathbf{0},\ \rho_- \geq 0.
\end{aligned} \tag{43}$$

It is easy to prove that the solutions of QPP (43) are the solutions of QPP (10) and (11). If we delete the term $\frac{1}{2}(\boldsymbol{\eta}_+^T\boldsymbol{\eta}_+ + \boldsymbol{\eta}_-^T\boldsymbol{\eta}_-)$ in the objective function in QPP (43), then QPP (43) can degenerate to be

$$\begin{aligned}
\min\ & \frac{c_1}{2}(||\mathbf{w}_+||^2 + b_+^2) + \frac{c_2}{2}(||\mathbf{w}_-||^2 + b_-^2) \\
& -\nu_1\rho_+ - \nu_2\rho_- + \frac{1}{q}\mathbf{e}_-^T\boldsymbol{\xi}_- + \frac{1}{p}\mathbf{e}_+^T\boldsymbol{\xi}_+ \\
s.t.\ \ & -(\mathbf{B}\mathbf{w}_+ + \mathbf{e}_-b_+) \geq \rho_+\mathbf{e}_- - \boldsymbol{\xi}_-, \\
& \mathbf{A}\mathbf{w}_- + \mathbf{e}_+b_- \geq \rho_-\mathbf{e}_+ - \boldsymbol{\xi}_+, \\
& \boldsymbol{\xi}_- \geq \mathbf{0},\ \rho_+ \geq 0, \\
& \boldsymbol{\xi}_+ \geq \mathbf{0},\ \rho_- \geq 0.
\end{aligned} \tag{44}$$

Furthermore, if we want to get the solutions satisfying $\mathbf{w}_+ = \mathbf{w}_-, b_+ = b_-, \rho_+ = \rho_-$, we only need to solve the special case of QPP (44), namely,

$$\begin{aligned}
\min\ & \frac{1}{2}(||\mathbf{w}||^2 + b^2) - \nu\rho + \frac{1}{l}(\mathbf{e}_-^T\boldsymbol{\xi}_- + \mathbf{e}_+^T\boldsymbol{\xi}_+) \\
s.t.\ \ & -(\mathbf{B}\mathbf{w} + \mathbf{e}_-b) \geq \rho\mathbf{e}_- - \boldsymbol{\xi}_-, \\
& \mathbf{A}\mathbf{w} + \mathbf{e}_+b \geq \rho\mathbf{e}_+ - \boldsymbol{\xi}_+, \\
& \boldsymbol{\xi}_- \geq \mathbf{0},\ \boldsymbol{\xi}_+ \geq \mathbf{0},\ \rho \geq 0.
\end{aligned} \tag{45}$$

It is obvious that (45) is the ν-SVM for binary classification problem. In other words, the ν-SVM with parallel hyperplane is a special case of Iν-TBSVM with nonparallel hyperplanes. Iν-TBSVM is more flexible than ν-SVM and has better generalization ability.

4 Comparison with other algorithms

4.1 Iν-TBSVM vs. ν-TBSVM

The matrix inverse operation is inescapable in ν-TBSVM. In the linear case, ν-TBSVM has to solve two matrix inverse operations of order $(n+1)$, where n denotes the number of dimensions of training samples. In the nonlinear case, ν-TBSVM has to solve two matrix inverse operations of order $(l+1)$, where l denotes the number of training samples. Therefore, ν-TBSVM is not suitable for large scale problems. Compared with ν-TBSVM, our Iν-TBSVM perfectly avoids the matrix inverse operation.

Besides, ν-TBSVM with the linear kernel is not equivalent to the linear case. Namely ν-TBSVM cannot obtain the exact solutions when using a nonlinear kernel. When using the linear kernel, our Iν-TBSVM can degenerate to the linear case easily. Therefore, our nonlinear Iν-TBSVM is superior to the nonlinear ν-TBSVM theoretically.

Their common ground is that they both implement the structural risk minimization principle.

4.2 Iν-TBSVM vs. TBSVM

Both Iν-TBSVM and TBSVM [25] find two nonparallel hyperplanes and classify two classes of training samples.

In TBSVM, the patterns of one class are at least a unit distance away from the hyperplane of other class, this may increase the number of support vectors thus leading to poor generalization ability. The unit distance of Iν-TBSVM is modified to variable ρ which is optimized in the primal problem involved therein. And the parameters ν in Iν-TBSVM are used to control the bounds on the fractions of support vectors and error margins. It implies that the parameters in our Iν-TBSVM have a better theoretical interpretation than TBSVM.

5 Numerical experiments

To validate the superiorities of our algorithm, in this section, we conduct the experiments on two artificial datasets and twenty-two benchmarking dataset. In the artificial datasets we verify the property of parameter ν in our Iν-TBSVM. In the twenty-two benchmarking experiments, we compare our Iν-TBSVM with three other algorithms, i.e. TSVM, ν-TSVM and ν-TBSVM, from both accuracy and time aspects. All experiments are carried out in Matlab R2014a on Windows 7 running on a PC with system configuration Inter Core i3-4160 Duo CPU (3.60GHz) with 4.00 GB of RAM.

5.1 Experiments on two artificial datasets

In this subsection, two artificial datasets have been used to show the properties of our proposed Iν-TBSVM. Firstly, we randomly generate two classes of points, each class has 50 points. They all follow two-dimensional normal distributions, where positive samples $X_1 \sim N(-2, 2)$, and negative samples $X_2 \sim N(3.5, 2)$. Their distributions are shown in Fig. 2. By our proposed Iν-TBSVM, we can easily obtain a pair of nonparallel hyperplanes for the two classes of samples, also shown as Fig. 2. The green and pink dashed represents the positive hyperplane ($f_+ = 0.002x_1 + 0.0018x_2 - 0.0188$) and the negative hyperplane ($f_- = -0.0022x_1 - 0.0022x_2 + 0.0044$), respectively. And the red solid line stands for the final classification hyperplane.

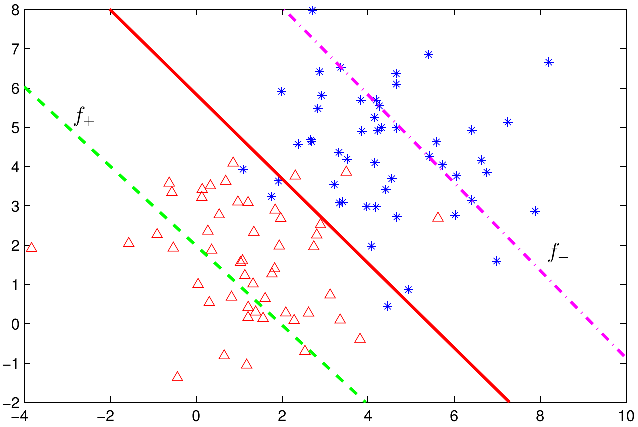

Fig. 2 The *red* "△" denotes the positive samples, and the *blue* "$*$" stands for the negative samples

To further investigate the property of parameter ν in our Iν-TBSVM, we present an intuitive Fig. 3, where the x-axis denotes the values of parameter ν. The blue curve denotes the changing curve between the fraction of support vectors in negative class and parameter ν_1. The green curve denotes the changing curve between the values of $\frac{p_2}{q}$ and parameter ν_1. This artificial experiment confirms the property of

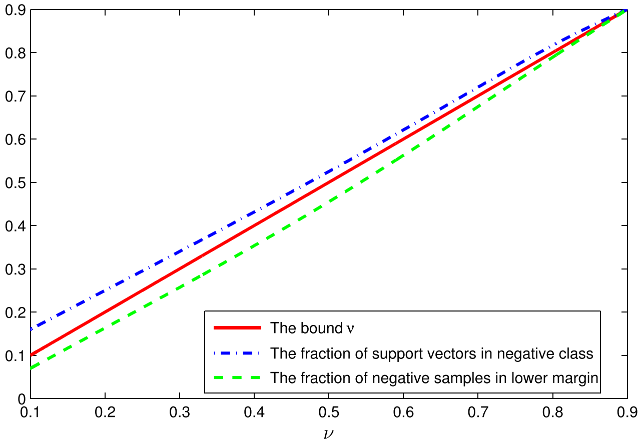

Fig. 3 The *red line* denotes a bound (ν_1). The *blue curve* denotes the fraction of support vectors in negative class. The *green curve* denotes the fraction of entirely errors in negative class in our Iν-TBSVM

parameter ν described in Propositions 2 and 3. Namely ν_1 is a lower bound on the fraction of support vectors in the negative class; ν_1 is an upper bound on the fraction of boundary errors in the negative class. This is helpful to understand our algorithm and helpful for the choice of parameter.

Secondly, we conduct the experiment on crossplane (XOR) dataset which has 101 points for each class. Similarly, we draw the pair of nonparallel hyperplanes, which is shown in Fig. 4. And from Fig. 5, Propositions 2 and 3 are also confirmed on XOR dataset.

5.2 Experiments on benchmarking datasets

We also test the effectiveness of our proposed algorithm on a collection of twenty-two benchmarking datasets from UCI machine learning repository [1]. These datasets are constructed for binary classification problems. In order to make the results more convincing, we use 10-fold cross-validation for each experiment. More specifically, each dataset is split randomly into ten subsets, and one of those sets is reserved as a test set whereas the remaining data are considered for training. This process is repeated ten times.

5.2.1 Parameters selection

Choosing the optimal parameters is an important problem for SVMs. In our experiments, we adopt the grid search method [34] to obtain the optimal parameters. The Gaussian kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/r^2)$ is used as it is often employed and yields great generalization performance. In the four algorithms, the Gaussian kernel parameter r is selected from the set $\{2^i|i = -5, -4, ..., 10\}$. For brevity's sake, we let $c_1 = c_2$ in TSVM, $\nu_1 = \nu_2$ in ν-TSVM, and $c_1 = c_2, \nu_1 = \nu_2$ in ν-TBSVM and Iν-TBSVM. The parameter c_1 is selected from the set $\{2^i|i = -10, -9, ..., 10\}$. The parameter ν_1 is searched from the set $\{0.1, 0.2, ..., 0.9\}$.

5.2.2 Results comparisons and discussion

We report testing accuracy to evaluate the performance of classifiers. 'Accuracy' denotes the mean value of ten testing results, plus or minus the standard deviation. 'Time' denotes the mean value of the time taken by ten experiments, and each experiment's time consists of training time and testing time, and the unit of time is seconds. At the same time, we record the optimal parameters of four algorithms during the experiments. To compare it more comprehensively, we do the experiments both in the linear and nonlinear cases on the twenty-two benchmarking datasets, and the results are reported in Tables 1 and 2, respectively. The bold values indicate best mean of accuracy (in %).

By analyzing our experimental results on the twenty-two benchmarking datasets, we can easily draw the following conclusions.

- No matter the linear case or the nonlinear case, our Iν-TBSVM outperforms other three algorithms for most datasets in terms of classification accuracy. ν-TBSVM follows, it produces better testing accuracy than ν-TSVM and TSVM for most cases. The main reason is that both Iν-TBSVM and ν-TBSVM implement the structural risk minimization principle.

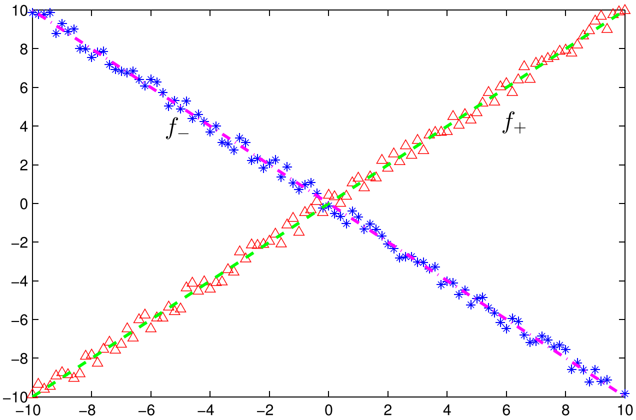

Fig. 4 The *red* "△" denotes the positive samples, and the *blue* "∗" stands for the negative samples. The separating hyperplanes of our proposed Iν-TBSVM on XOR dataset

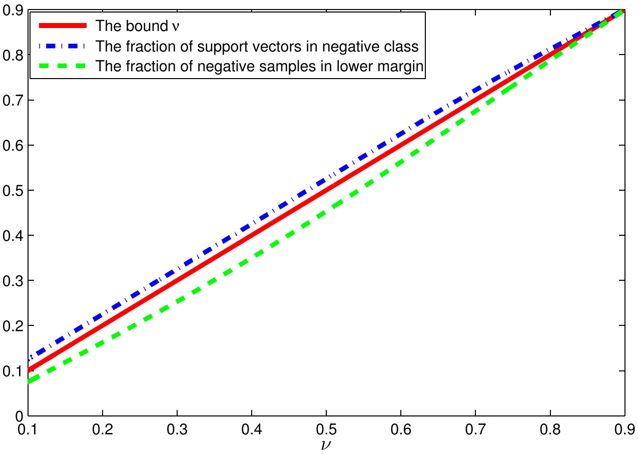

Fig. 5 The *red line* denotes a bound (ν_1). The *blue curve* denotes the fraction of support vectors in negative class. The *green curve* denotes the fraction of entirely errors in negative class in our Iν-TBSVM on XOR dataset

[1] http://archive.ics.uci.edu/ml/datasets.html.

Table 1 Performance comparisons of four linear algorithms on twenty-two datasets

Datasets	TSVM		ν-TSVM		ν-TBSVM		Iν-TBSVM	
	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
	c_1		ν_1		(ν_1, c_1)		(ν_1, c_1)	
Wine	**100.00±0.00**	0.07	**100.00±0.00**	0.03	**100.00±0.00**	0.14	**100.00±0.00**	0.16
(130 × 13)	0.5		0.7		(0.2,0.313)		(0.7,0.5)	
Monks	86.88±4.04	0.54	86.95±3.61	0.50	88.44±3.45	0.40	**89.87±3.83**	0.80
(554 × 6)	0.25		0.9		(0.3,32)		(0.4,1024)	
PimaIndians	76.84±3.44	2.23	76.63±20.91	1.40	79.59±2.20	0.79	**81.33±4.41**	7.28
(758 × 8)	1		0.8		(0.1,1)		(0.5,0.313)	
Vote	99.05±0.92	0.46	**99.26±0.71**	0.36	99.05±0.92	0.41	**99.26±0.71**	0.46
(435 × 16)	0.0625		0.8		(0.5,0.5)		(0.4,4)	
Ionosphere	76.59±20.15	0.23	76.59±20.15	0.35	**88.29±18.85**	0.53	81.95±7.11	0.42
(351 × 34)	0.002		0.2		(0.9,1)		(0.7,8)	
Australian	87.79±1.73	1.40	87.57±1.97	0.56	87.86±2.86	0.57	**88.21±3.20**	3.79
(690 × 14)	1		0.5		(0.2,32)		(0.8,16)	
German	**89.47±2.10**	2.56	75.90±1.98	1.21	87.97±1.63	0.91	83.70±2.37	2.03
(1000 × 24)	0.5		0.8		(0.2,16)		(0.3,0.0078)	
Abalone	70.27±31.85	11.00	64.63±23.13	7.57	74.43±10.43	7.05	**75.97±4.81**	6.76
(2835 × 8)	0.5		0.8		(0.2,32)		(0.1,2)	
Balance	96.08±3.48	0.57	95.17±3.64	0.48	**96.48±2.88**	0.34	96.36±2.56	1.21
(576 × 4)	64		0.2		(0.1,16)		(0.9,1024)	
Bupa	75.26±5.26	0.23	84.32±2.78	0.18	84.63±6.00	0.22	**85.47±6.23**	0.17
(345 × 6)	0.5		0.7		(0.7,8)		(0.6,128)	
Iris	**100.00±0.00**	0.10	**100.00±0.00**	0.07	**100.00±0.00**	0.05	**100.00±0.00**	0.07
(150 × 4)	0.5		0.3		(0.2,1)		(0.1,0.25)	
Heart	80.17±24.35	0.19	83.17±22.39	0.19	83.67±21.77	0.23	**84.50±5.06**	0.60
(270 × 150)	1		0.7		(0.7,0.313)		(0.2,2)	
Lung	62.22±11.94	0.12	64.44±11.48	0.09	67.78±17.72	0.15	**75.56±14.63**	0.07
(23 × 56)	256		0.9		(0.8,16)		(0.9,32)	
CMC	**100.00±0.00**	13.66	**100.00±0.00**	14.91	**100.00±0.00**	12.10	**100.00±0.00**	15.43
(1473 × 10)	9.77E-04		0.5		(0.6,16)		(0.2,8)	
Connectionist	79.57±12.97	0.18	78.70±17.08	0.18	80.00±18.92	0.20	**83.48±18.08**	0.25
(208 × 60)	0.0156		0.7		(0.2,32)		(0.4,0.039)	
Dbworld	93.57±11.88	20.37	93.57±11.88	20.41	**94.29±12.05**	20.40	**94.29±12.05**	12.37
(64 × 4702)	0.125		0.8		(0.5,4)		(0.3,0.0625)	
WBCD	99.28±1.16	2.09	99.04±1.37	1.02	99.28±1.02	0.84	**99.28±1.30**	2.63
(683 × 9)	8		0.2		(0.2,8)		(0.6,1024)	
Hepatitis	89.64±7.98	0.21	91.43±6.12	0.25	**94.64±3.47**	0.22	92.86±5.05	0.21
(155 × 19)	0.0156		0.5		(0.8,4)		(0.3,0.0625)	
Fertility	77.5±7.91	0.25	81.25±0.00	0.24	81.25±0.00	0.15	**83.75±7.91**	0.19
(100 × 9)	0.5		0.1		(0.1,0.0625)		(0.1,0.002)	
Haberman	77.50±19.22	0.20	77.22±16.86	0.32	82.22±13.75	0.21	**84.17±9.45**	0.54
(306 × 3)	0.0078		0.9		(0.1,0.0078)		(0.7,1)	
Climate	**99.33±0.86**	1.80	98.83±1.37	0.81	**99.33±1.17**	2.08	98.50±3.19	3.40
(540 × 20)	0.0625		0.5		(0.9,0.313)		(0.1,0.125)	
WPBC	78.93±8.98	0.24	79.29±3.69	0.17	**82.86±6.02**	0.23	81.79±6.83	0.33
(198 × 33)	0.125		0.3		(0.5,0.0313)		(0.9,1)	
Average	90.28	2.67	90.19	2.33	92.95	2.19	**93.35**	2.69

Table 2 Performance comparisons of four nonlinear algorithms with Gauss kernel on twenty-two datasets

Datasets	TSVM		ν-TSVM		ν-TBSVM		Iν-TBSVM	
	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
	(c_1, r)		(ν_1, r)		(ν_1, r, c_1)		(ν_1, r, c_1)	
Wine	97.75±1.42	0.24	98.25±2.06	0.26	98.75±1.32	0.28	**98.83±1.53**	0.18
	(4,1024)		(0.8,1024)		(0.9,1024,4)		(0.8,512,2)	
Monks	**99.68±0.34**	1.73	98.96±1.47	1.56	97.86±1.56	2.53	99.09±1.02	1.01
	(8,128)		(0.1,64)		(0.1,4,0.125)		(0.1,1,0.313)	
PimaIndians	75.54±3.90	3.80	76.31±5.20	5.93	76.80±4.90	4.31	**77.56±4.13**	2.27
	(4,256)		(0.9,256)		(0.9,512,16)		(0.4,512,32)	
Votes	95.58±2.78	1.16	98.63±1.22	1.14	**98.95±1.22**	1.12	96.51±2.26	0.88
	(0.0313,1024)		(0.4,1024)		(0.2,256,0.625)		(0.8,1024,4)	
Ionosphere	94.88±12.03	0.73	95.37±12.14	1.36	95.12±12.06	0.94	**95.61±4.11**	0.54
	(0.0625,2)		(0.8,2)		(0.1,128,0.25)		(0.2,0.25,0.25)	
Australian	74.64±17.5	3.24	83.14±2.94	4.33	82.79±2.06	2.44	**84.79±6.26**	1.69
	(0.5,512)		(0.9,512)		(0.8,1024,0.1)		(0.1,256,0.25)	
German	**88.10±10.20**	6.39	83.87±17.06	6.59	84.23±16.55	4.11	80.17±7.23	3.47
	(0.5,1024)		(0.3,64)		(0.2,1024,2)		(0.4,128,0.125)	
Abalone	74.69±19.46	58.29	75.09±17.93	36.19	75.89±21.24	30.08	**76.06±18.83**	28.24
	(0.125,1)		(0.3,1)		(0.5,0.0625,2)		(0.3,256,8)	
Balance	97.78±2.25	1.65	97.9±2.91	1.51	98.58±2.59	1.64	**98.98±1.16**	1.15
	(8,512)		(0.7,256)		(0.9,8,0.25)		(0.9,8,2)	
Bupa	70.95±4.01	0.69	70.21±4.49	0.95	73.89±4.54	0.64	**74.32±5.04**	0.43
	(1,64)		(0.1,64)		(0.9,128,0.5)		(0.1,128,1)	
Iris	**100.00±0.00**	0.20	**100.00±0.00**	0.21	**100.00±0.00**	0.23	**100.00±0.00**	0.12
	(0.313,0.25)		(0.1,0.25)		(0.7,128,4)		(0.1,0.625,0.313)	
Heart	80.00±1.91	0.64	84.44±3.4	0.80	83.70±3.12	0.57	**84.81±1.17**	0.38
	(2,256)		(0.4,1024)		(0.2,1024,32)		(0.9,512,4)	
Lung	65.56±14.3	0.10	73.33±14.05	0.13	75.56±14.63	0.12	**84.44±11.94**	0.07
	(32,32)		(0.9,1024)		(0.8,128,0.125)		(0.4,1024,0.313)	
CMC	**100.00±0.00**	20.08	**100.00±0.00**	26.76	**100.00±0.00**	10.67	99.45±0.89	8.55
	(9.7656e-04,1024)		(0.1,512)		(0.5,1024,1)		(0.5,2,4)	
Connectionist	85.22±12.5	0.48	83.04±7.52	0.47	84.35±6.55	0.43	**85.65±5.82**	0.31
	(4,64)		(0.1,1)		(0.5,1,0.0625)		(0.1,0.0625,32)	
Dbworld	95.00±8.94	0.54	94.29±8.78	0.55	**95.71±9.04**	0.54	93.57±8.55	0.40
	(16,512)		(0.5,256)		(0.4,512,0.0625)		(0.1,1024,2)	
WBCD	99.28±1.02	4.85	99.16±1.28	6.24	99.16±1.14	3.14	**99.28±1.02**	2.20
	(0.0625,32)		(0.3,64)		(0.9,512,0.25)		(0.9,512,0.25)	
Hepatitis	81.18±2.48	0.38	81.76±3.34	0.52	**84.71±4.11**	0.28	83.40±2.55	0.15
	(4,512)		(0.8,512)		(0.7,1024,0.313)		(0.7,0.5,8)	
Fertility	84.00±8.43	0.12	84.00±8.43	0.15	84.00±8.43	0.22	**85.00±5.27**	0.08
	(0.313,0.125)		(0.1,0.125)		(0.1,0.125,0.0313)		(0.6,0.0313,1)	
Haberman	74.72±13.38	0.60	77.78±15.38	0.67	83.06±12.99	0.64	**83.33±8.59**	0.54
	(0.0313,0.125)		(0.1,0.125)		(0.7,512,8)		(0.6,512,1)	
Climate	90.50±1.12	3.00	90.33±0.7	2.97	**92.50±1.96**	2.48	91.17±1.58	1.46
	(0.3,1024)		(0.2,1024)		(0.9,32,0.0625)		(0.7,0.313,1024)	
WPBC	78.21±1.38	0.26	79.01±2.73	0.28	**79.80±3.40**	0.25	**79.80±3.40**	0.20
	(0.0313,0.0625)		(0.1,0.0625)		(0.1,0.313,0.313)		(0.1,0.125,0.125)	
Average	90.63	4.96	91.66	4.53	92.64	3.08	**92.94**	2.47

Table 3 Average rank on accuracy of four linear algorithms on twenty-two datasets

Datasets	TSVM	ν-TSVM	ν-TBSVM	Iν-TBSVM
Wine	2.5	2.5	2.5	2.5
Monks	4	3	2	1
PimaIndians	3	4	2	1
Vote	3.5	1.5	3.5	1.5
Ionosphere	3.5	3.5	1	2
Australian	3	4	2	1
German	1	4	2	3
Abalone	3	4	2	1
Balance	3	4	2	1
BUPA	4	3	2	1
Iris	2.5	2.5	2.5	2.5
Heart	4	3	2	1
Lung	4	3	2	1
CMC	2.5	2.5	2.5	2.5
Connectionist	4	3	2	1
Dbworld	3.5	3.5	1.5	1.5
WBCD	2	4	2	2
Hepatitis	4	3	1	2
Fertility	4	2.5	2.5	1
Haberman	3	4	2	1
Climate	1.5	3	1.5	4
WPBC	4	3	1	2
Average rank	3.159	3.205	1.932	1.705

Table 4 Average rank on accuracy of four nonlinear algorithms with Gauss kernel on twenty-two datasets

Datasets	TSVM	ν-TSVM	ν-TBSVM	Iν-TBSVM
Wine	4	3	2	1
Monks	1	3	4	2
PimaIndians	4	3	2	1
Vote	4	2	1	3
Ionosphere	4	2	3	1
Australian	4	2	3	1
German	1	3	2	4
Abalone	4	3	2	1
Balance	3	4	2	1
BUPA	3	4	2	1
Iris	2.5	2.5	2.5	2.5
Heart	4	2	3	1
Lung	4	3	2	1
CMC	2	2	2	4
Connectionist	3	4	2	1
Dbworld	2	3	1	4
WBCD	1.5	3.5	3.5	1.5
Hepatitis	4	3	1	2
Fertility	3	3	3	1
Haberman	4	3	2	1
Climate	3	4	1	2
WPBC	4	3	1.5	1.5
Average rank	3.136	2.955	2.159	1.750

- It is worthwhile to mention that our Iν-TBSVM takes the least computational time on all twenty-two benchmarking dataset in nonlinear case. The mean running time of Iν-TBSVM for twenty-two datasets is 2.47 seconds as compared to 3.08, 4.53 and 4.96 seconds for ν-TBSVM, ν-TSVM and TSVM. The main reason is also that our Iν-TBSVM doesn't involve matrix inverse operation when solving the dual QPPs which greatly reduced the time spent to run the algorithms.
- Compared with the other three algorithms, our Iν-TBSVM has better performance on Dbworld datasets. This implies that our Iν-TBSVM is suitable for the high dimensional dataset. At the same time, when the number of training sample is large, such as Abalone dataset, our Iν-TBSVM still performs best in both testing accuracy and running time. As the aforementioned analysis, the other three algorithms take much time to do matrix inverse operation when solving the dual QPPs.

5.3 Friedman test

In order to further analyze the performance of the four comparable algorithms on twenty-two datasets with statistic methods, we use Friedman test [35] with the corresponding post hoc tests as suggested in Dems̃ar [36] and García et al. [37]. The Friedman test is proved to be simple, nonparametric and safe for comparing three or more related samples and it makes no assumptions about the underlying distribution of the data. For this, the average ranks of four linear and nonlinear algorithms on accuracy for twenty-two datasets are calculated and listed in Tables 3 and 4, respectively. Under the null-hypothesis that all the algorithms are equivalent, one can compute the Friedman statistic [36] according to the following equation,

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right], \tag{46}$$

where $R_j = \frac{1}{N}\sum_i r_i^j$, and r_i^j denotes the jth of k algorithms on the ith of N datasets. Friedman's χ_F^2 is undesirably conservative and derives a better statistic

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} \tag{47}$$

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

We can obtain $\chi_F^2 = 24.9606$ and $F_F = 12.7724$ according to (46) and (47) for linear case. Similarly, we obtain

$\chi_F^2 = 17.0320$ and $F_F = 7.3042$ for nonlinear case where F_F is distributed according to F-distribution with (3, 63) degrees of freedom. The critical value of $F(3, 63)$ is 2.17 for the level of significance $\alpha = 0.1$, similarly, it is 2.75 for $\alpha = 0.05$ and 3.33 for $\alpha = 0.025$. These results suggest that there is significant difference among the four algorithms since the value of F_F is 12.7724 for linear case and 7.3042 for nonlinear case. Both of the values are much larger than the critical value. We can also illustrate that our Iν-TBSVM outperforms the other three algorithms in the linear case or nonlinear case, because Iν-TBSVM gets the lowest average rank both in Tables 3 and 4.

6 Conclusion

In this paper, we present a novel algorithm, i.e., the improved ν-twin bounded support vector machine. This Iν-TBSVM not only maximizes the margin between two parallel hyperplanes by introducing a regularization term into the objective function, but also inherits the merits of standard SVM. Firstly, the matrix inverse operation is skillfully avoided in our Iν-TBSVM while it is inevitable for most existing ν-TSVMs. Secondly, the kernel trick can be applied directly to Iν-TBSVM for the nonlinear case, which is essential to obtain a model with higher testing accuracy. Thirdly, we prove that the ν-SVM is a special case of Iν-TBSVM. Iν-TBSVM is more flexible than ν-SVM and has better generalization ability. Experimental results indicate that our algorithm gives better performance than others. What's more, this novel method can be applied in many fields, such as multi-class classification, semi-supervised learning and so on.

Acknowledgements The authors gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation. This work was supported in part by National Natural Science Foundation of China (No.11671010) and Beijing Natural Science Foundation (No.4172035).

References

1. Vapnik V (1995) The nature of statistical learning theory. Springer, New York
2. Ripley BD (1996) Pattern recognition and neural networks. Cambridge University Press, Cambridge
3. Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge University Press, Cambridge
4. Brown MPS, Grundy WN, Lin D, Cristianini N, Sugnet C, Furey TS, Ares JM, Haussler D (2000) Knowledge-based analysis of microarray gene expression data by using support vector machine. Proc Natl Acad Sci USA 97:262–267
5. Cao XB, Xu YW, Chen D, Qiao H (2009) Associated evolution of a support vector machine-based classifier for pedestrian detection. Inf Sci 179:1070–1077
6. Ghosh S, Mondal S, Ghosh B (2014) A comparative study of breast cancer detection based on SVM and MLP BPN classifier. In: First international conference on automation, control, energy & systems (ACES-14), pp 87–90
7. Cortes C, Vapnik V (1995) Support vector networks. Mach Learn 20:273–297
8. Osuna E, Freund R, Girosi F (1997) Support vector machines: training and applications. Technical Report, MIT Artificial Intelligence Laboratory, Cambridge, MA
9. Platt J (1998) Sequential minimal optimization: a fast algorithm for training support vector machines. In: Scholkopf et al. (eds.), Technical report MSR-TR-98-14, Microsoft research, pp 185–208
10. Schölkopf B, Burges CJC, Smola AJ (eds.) (1999) Advances in kernel methods: support vector learning. MIT Press, Cambridge
11. Keerthi SS, Shevade SK, Bhattacharyya C, Murthy K (2001) Improvements to platts SMO algorithm for SVM classifier design. Neural Comput 13(3):637–649
12. Schölkopf B, Smola AJ, Bartlett P, Williamson RC (2000) New support vector algorithms. Neural Comput 12(5):1207–1245
13. Lee Y, Mangasarian OL (2001) Ssvm: a smooth support vector machine for classification. Comput Optim Appl 20(1):5–22
14. Schölkopf B, Bartlett PL, Smola AJ, Williamson R (1999) Shrinking the tube: a new support vector regression algorithm. Advances in neural information processing systems, pp 330–336
15. Suykens JA, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Process Lett 9(3):293–300
16. Mangasarian OL, Wild EW (2001) Proximal support vector machine classifiers. In: Proceedings KDD-2001: knowledge discovery and data mining. Citeseer
17. Mangasarian OL, Wild EW (2006) Multisurface proximal support vector machine classification via generalized eigenvalues. IEEE Trans Pattern Anal Mach Intell 28(1):69–74
18. Jayadeva, Khemchandani R, Chandra S (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
19. Tian Y, Qi Z (2014) Review on: twin support vector machines. Annals Data Sci 1(2):253–277
20. Kumar MA, Gopal M (2008) Application of smoothing technique on twin support vector machines. Pattern Recogn Lett 29:1842–1848
21. Kumar MA, Gopal M (2009) Least squares twin support vector machines for pattern classification. Expert Syst Appl 36:7535–7543
22. Peng XJ (2010) TSVR: an efficient twin support vector machine for regression. Neural Netw 23(3):365–372
23. Shao YH, Zhang CH, Wang XB, Deng NY (2011) Improvements on twin support vector machines. IEEE Trans Neural Netw 22:962–968
24. Peng X, Wang Y, Xu D (2013) Structural twin parametric-margin support vector machine for binary classification. Knowl-Based Syst 49:63–72
25. Tian Y, Ju X, Qi Z, Shi Y (2014) Improved twin support vector machine. Sci China (Mathematics) 57:417–432
26. Peng XJ (2010) A ν-twin support vector machine (ν-TSVM) classifier and its geometric algorithms. Inf Sci 180(20):3863–3875
27. Xu Y, Guo R (2014) An improved ν-twin support vector machine. Appl Intell 41:42–54
28. Xu Y, Wang L, Zhong P (2012) A rough margin-based ν-twin support vector machine. Neural Comput Appl 21:1307–1317
29. Xu Y, Yu J, Zhang Y (2014) KNN-based weighted rough ν-twin support vector machine. Knowl-Based Syst 71:303–313

30. Khemchandani R, Saigal P, Chandra S (2016) Improvements on ν-twin support vector machine. Neural Netw 79:97–107
31. Duncan WJ (1944) Some devices for the solution of large sets of simultaneous linear equations. The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science, Seventh Series 35(249):660–670
32. Sherman J, Morrison WJ (1949) Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix. Ann Math Stat 20:621
33. Woodbury M (1950) Inverting modified matrices. Memorandum Report 42. Statistical Research Group Princeton University, Princeton
34. Lin CJ, Hsu CW, Chang CC (2003) A practical guide to support vector classification. National Taiwan U., www.csie.ntu.edu.tw/cjlin/papers/guide/guide.pdf
35. Holm S (1979) A simple sequentially rejective multiple test procedure. Scand J Stat 6(2):65–70
36. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
37. García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. Inform Sci 180:2044–2064

Huiru Wang received her B.S. and M.S. degrees in College of Science from China Agricultural University in 2013 and 2015, respectively. She is currently pursuing the Ph.D. degree in College of Science, China Agriculture University. Her research interests include support vector machine and data mining.

Zhijian Zhou received her Ph.D. degree in College of Engineering from China Agricultural University in 2007. Dr. Zhou is now a Professor of College of Science, China Agricultural University. Her research interests include machine learning and data mining.

Yitian Xu received the Ph.D. degree from the College of Science, China Agricultural University, Beijing, China, in 2007. He was a Visiting Scholar with the Department of Computer Science and Engineering, Arizona State University, Tempe, AZ, USA, from 2013 to 2014. He is currently a Professor and Supervisor for the Ph.D. candidates with the College of Science, China Agricultural University. He has authored about 50 papers. His current research interests include machine learning and data mining. Prof. Xu's research has appeared in IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Cybernetics, Knowledge-Based Systems, Neurocomputing, Neural Computing with Applications, Cognitive Computation.