Smooth twin bounded support vector machine with pinball loss

Kai Li¹ · Zhen Lv¹

Accepted: 17 November 2020 / Published online: 9 January 2021
© The Author(s) 2021

Abstract

The twin support vector machine improves the classification performance of the support vector machine by solving two small quadratic programming problems. However, this method has the following defects: (1) For the twin support vector machine and some of its variants, the constructed models use a hinge loss function, which is sensitive to noise and unstable in resampling. (2) The models need to be converted from the original space to the dual space, and their time complexity is high. To further enhance the performance of the twin support vector machine, the pinball loss function is introduced into the twin bounded support vector machine, and the problem of the pinball loss function not being differentiable at zero is solved by constructing a smooth approximation function. Based on this, a smooth twin bounded support vector machine model with pinball loss is obtained. The model is solved iteratively in the original space using the Newton-Armijo method. A smooth twin bounded support vector machine algorithm with pinball loss is proposed, and theoretically the convergence of the iterative algorithm is proven. In the experiments, the proposed algorithm is validated on the UCI datasets and the artificial datasets. Furthermore, the performance of the presented algorithm is compared with those of other representative algorithms, thereby demonstrating the effectiveness of the proposed algorithm.

Keywords Original space · Iteration · Pinball loss · Twin bounded support vector machines

1 Introduction

The support vector machine (SVM) proposed by Vapnik et al. [1] is a machine learning method that is based on the principle of the Vapnik-Chervonekis (VC) dimension and structural risk minimization in statistical learning. It has been widely used in many fields, such as text recognition [2], feature extraction [3], multiview learning [4, 5], and sample screening [6]. This method uses the margin maximization strategy to find the optimal classification hyperplane that classifies the training datasets correctly with a sufficiently large certainty factor. This means that the hyperplane not only classifies positive and negative samples, but also has a sufficiently large certainty factor for classifying samples that are close to the hyperplane. Therefore, SVM has many advantages in terms of classification and prediction of unknown samples. In addition, SVM is formalized to solve a convex quadratic programming problem. However, with the continuous growth of the

scale of data, some problems with SVM have been exposed. For example, SVM has high complexity with $O(m^3)$, where m is the number of training samples. It has difficulty coping with the processing requirements of large-scale data, and its anti-noise performance is weak. To this end, researchers have proposed some improved methods. Mangasarian et al. [7] proposed the proximal support vector machine (PSVM). This algorithm uses equation constraints to transform a quadratic programming problem into a problem of solving linear equations, and this improves the training speed of the algorithm. On this basis, they further proposed a generalized eigenvalue proximal support vector machine GEPSVM [8]. In their proposed algorithm, by relaxing the parallel constraints of the hyperplane, the original problem is transformed into solving two generalized eigen equations.

Inspired by the PSVM and GEPSVM, Jayadeva et al. [9] proposed the twin support vector machine (TWSVM) by minimizing the empirical risk, turning the single large quadratic programming problem into a pair of smaller quadratic programming problems. Unlike the GEPSVM, the TWSVM requires one hyperplane at least one unit distance away from other class. The complexity of the algorithm becomes 1/4 that of the SVM, and this further saves computation time. It has been shown that the TWSVM has more favorable properties than the SVM in terms of computing complexity and

✉ Kai Li
likai@hbu.cn

¹ School of Cyber Security and Computer, Hebei University, Baoding 071000, China

classification accuracy. Since TWSVM has excellent classification performance, it has attracted increasing attention. Therefore, many variants of the TWSVM have been proposed. For example, the twin bounded support vector machine (TBSVM) proposed by Shao et al. [10] is implemented by adding a regularization term to the objective function. It further improves the generalization performance of the TWSVM. In addition, researchers have proposed different twin support vector machines by exploring the structures of datasets and the different roles of samples [11–18].

From the above, to establish a support vector machine model, researchers mainly use a hinge loss function, which is sensitive to noise and unstable for resampling. To cope with these problems, Huang et al. [19] introduced the pinball loss into the support vector machine and proposed the Pin-SVM, which better solved the problems of noise sensitivity and resampling instability than previous SVMs. After that, researchers introduced the pinball loss into twin support vector machines and proposed different versions [20–22]. However, these models still need to solve two quadratic programming problems in the dual space. In fact, for the solution of the model in the original space, Mangasarian et al. [23] studied the support vector machine using smoothing technique and proposed a smooth support vector machine (SSVM). Moreover, researchers applied the smoothing technique to the twin support vector machines and proposed different algorithms for such twin support vector machines [24–27]. To obtain the required decision surface, the corresponding models of the twin support vector machines mentioned above are mainly solved in the dual space. Even if the researchers proposed a twin support vector machine solved in the original space, the model is still based on the hinge loss function.

The contributions of this paper are as follows:

1. We introduce the pinball loss into the twin bounded support vector machine to construct a twin bounded support vector machine model with pinball loss, which overcomes noise sensitivity and resampling instability in the hinge loss function.
2. To solve the nondifferentiable term in the objective function of the model, we construct a smooth approximation function for the pinball loss function so that the model in the original space can be solved directly. At the same time, the smooth twin bounded support vector machine with pinball loss is proposed.
3. A proof of the convergence of the algorithm is given theoretically.
4. We select UCI datasets and artificially generated datasets to validate the presented algorithm and compare its performance with the representative algorithms.

The remainder of this paper is organized as follows. Section 2 recalls three kinds of support vector machine models

and analyzes the pinball loss function. Moreover, the corresponding smooth approximation function for the pinball loss function is constructed. Section 3 introduces the smooth twin bounded support vector machine model with pinball loss. The Newton-Armijo method is used to solve the model. Based on this, a smooth twin bounded support vector machine algorithm with pinball loss is proposed. Section 4 provides the theoretical analysis and proof of convergence for the presented algorithm. Experiments are conducted on UCI datasets and artificially generated datasets in section 5. In addition, we also discuss the other approximation function based on the integral of sigmoid function for the pinball loss function and experiments on the toy dataset. The detailed derivations are provided in the appendix. Finally, the conclusion is given.

2 Related works

In this section, we briefly review the formulations of the smooth support vector machine (SSVM), twin support vector machine (TWSVM), twin bounded support vector machine (TBSVM) and a smooth approximation function for the pinball loss function.

2.1 SSVM

Lee et al. [23] introduced a smooth method into the SVM and conducted a new formulation of SVM.

The standard support vector machine is

$$\begin{aligned} \min_{(w,b,\xi) \in \mathbb{R}^{n+1+m}} \quad & v e^T \xi + \frac{1}{2} \|w\|^2 \\ \text{s.t.} \quad & D(Aw - eb) + \xi \geq e, \\ & \xi \geq 0. \end{aligned} \quad (1)$$

The matrix A contains m points in n -dimensional place, $v > 0$, $w \in \mathbb{R}^n$, $b \in \mathbb{R}$. ξ is the slack variable and e is a column vector of ones that is of arbitrary dimension and D is an $m \times m$ order matrix with ones or minus ones on its diagonal. In the smooth approach, using twice the reciprocal squared of the margin instead, the modified SVM problem is as follows:

$$\begin{aligned} \min_{(w,b,\xi) \in \mathbb{R}^{n+1+m}} \quad & \frac{v}{2} \xi^T \xi + \frac{1}{2} (\|w\|^2 + b^2) \\ \text{s.t.} \quad & D(Aw - eb) + \xi \geq e, \\ & \xi \geq 0. \end{aligned} \quad (2)$$

According to [28], the plus function is defined as

$$(u)_+ = \max\{u, 0\}, \quad (3)$$

and ξ is given by

$$\xi = (e - D(Aw - eb))_+ \quad (4)$$

Furthermore, the equivalent unconstrained problem is formulated as

$$\min_{w,b} \frac{\nu}{2} \|(e - D(Aw - eb))_+\|_2^2 + \frac{1}{2} (\|w\|^2 + b^2). \quad (5)$$

It is clear that (5) is strongly convex and has a unique solution. However, it is not twice differentiable at zero. Therefore, the integral of the sigmoid function

$$p(x, \alpha) = x + \frac{1}{\alpha} \log(1 + e^{-\alpha x}), \alpha > 0, \quad (6)$$

is used here to replace the first term in (5), and a smooth support vector machine can be obtained:

$$\min_{w,b} \Phi_\alpha(w, b) = \min_{w,b} \frac{\nu}{2} \|p(e - D(Aw - eb), \alpha)\|_2^2 + \frac{1}{2} (\|w\|^2 + b^2). \quad (7)$$

It can be seen that (7) is smooth and twice differentiable, so it can be solved by using the Newton-Armijo method [29–31].

2.2 TWSVM

Consider a binary classification dataset $X = \{(x_i, y_i) | i = 1, 2, \dots, m\}$, $x_i \in R^n$, $y_i \in \{+1, -1\}$. It contains m_1 positive samples and m_2 negative samples, which are represented by the matrices A and B , respectively, with $m_1 + m_2 = m$. Jayadeva et al. [9] proposed the twin support vector machine (TWSVM) using the hinge loss function, turning the optimization problem into a pair of smaller quadratic programming problems. Two nonparallel classification hyperplanes $f_1(x) = w_1^T x + b_1 = 0$ and $f_2(x) = w_2^T x + b_2 = 0$ need to be found to separate the samples correctly.

The primal problems of the TWSVM can be expressed as follows:

$$\begin{aligned} \min_{w_1, b_1, \xi_1} \quad & \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + c_1 e_2^T \xi_1 \\ \text{s.t.} \quad & -(Bw_1 + e_2 b_1) + \xi_1 \geq e_2, \end{aligned} \quad (8)$$

$$\begin{aligned} \min_{w_2, b_2, \xi_2} \quad & \frac{1}{2} \|Bw_2 + e_2 b_2\|^2 + c_2 e_1^T \xi_2 \\ \text{s.t.} \quad & (Aw_2 + e_1 b_2) + \xi_2 \geq e_1, \end{aligned} \quad (9)$$

where $c_1, c_2 > 0$, ξ_1 and ξ_2 are slack variables, e_1 and e_2 are column vectors of ones those are of arbitrary dimension. For optimization problems (8) and (9), the Lagrange method and K.K.T. optimality conditions are used to obtain their dual problems:

$$\max_{\alpha} \quad e_2^T \alpha - \frac{1}{2} \alpha^T G (H^T H)^{-1} G^T \alpha \quad \text{s.t.} \quad 0 \leq \alpha \leq c_1, \quad (10)$$

$$\max_{\alpha} \quad e_1^T \gamma - \frac{1}{2} \gamma^T P (Q^T Q)^{-1} P^T \gamma \quad \text{s.t.} \quad 0 \leq \gamma \leq c_2, \quad (11)$$

where $\alpha \geq 0$, $\gamma \geq 0$.

2.3 TBSVM

Shao et al. [10] introduced a regularization term into the TWSVM and proposed a twin bounded support vector machine (TBSVM). The primal problems of the TBSVM can be expressed as follows:

$$\begin{aligned} \min_{w_1, b_1, \xi_1} \quad & \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + \frac{1}{2} c_3 (\|w_1\|^2 + b_1^2) + c_1 e_2^T \xi_1 \\ \text{s.t.} \quad & -(Bw_1 + e_2 b_1) + \xi_1 \geq e_2, \end{aligned} \quad (12)$$

$$\begin{aligned} \min_{w_2, b_2, \xi_2} \quad & \frac{1}{2} \|Bw_2 + e_2 b_2\|^2 + \frac{1}{2} c_4 (\|w_2\|^2 + b_2^2) + c_2 e_1^T \xi_2 \\ \text{s.t.} \quad & (Aw_2 + e_1 b_2) + \xi_2 \geq e_1, \end{aligned} \quad (13)$$

where $c_1, c_2, c_3, c_4 > 0$, ξ_1 and ξ_2 are slack variables, e_1 and e_2 are column vectors of ones those are of arbitrary. For optimization problems (12) and (13), the Lagrange method and K.K.T. optimality conditions are used to obtain their dual problems:

$$\max_{\alpha} \quad e_2^T \alpha - \frac{1}{2} \alpha^T G (H^T H + c_3 I)^{-1} G^T \alpha \quad \text{s.t.} \quad 0 \leq \alpha \leq c_1, \quad (14)$$

$$\max_{\alpha} \quad e_1^T \gamma - \frac{1}{2} \gamma^T P (Q^T Q + c_4 I)^{-1} P^T \gamma \quad \text{s.t.} \quad 0 \leq \gamma \leq c_2, \quad (15)$$

where $\alpha \geq 0$, $\gamma \geq 0$.

2.4 Pinball loss and its smooth approximation function

In support vector machines or twin support vector machines, the hinge loss function is typically used to measure the correctness and error of classification. However, this function has strong sensitivity to noise and instability of resampling. Therefore, this paper uses the pinball loss function to study the twin bounded support vector machine.

We know that the Chen-Harker-Kanzow-Smale function [28] is one of the approximations of the plus function, and it is denoted by

$$\frac{u + \sqrt{u^2 + 4\varepsilon^2}}{2}, \quad (16)$$

where $\varepsilon > 0$.

The pinball loss function can be rewritten as

$$L_\tau(u) = (u)_+ + (-\tau u)_+, \quad \tau \in [0, 1]. \quad (17)$$

From (17), we know that the left and right derivatives of the function at zero are

$$\lim_{\Delta u \rightarrow 0^-} \frac{L_\tau(0 + \Delta u) - L_\tau(0)}{\Delta u} = \frac{-\tau(0 + \Delta u) - 0}{\Delta u} = -\tau \quad \text{and}$$

$$\lim_{\Delta u \rightarrow 0^+} \frac{L_\tau(0 + \Delta u) - L_\tau(0)}{\Delta u} = \frac{0 + \Delta u - 0}{\Delta u} = 1$$

It can be seen that the left and right derivatives of the pinball loss function at zero are not equal, that is, the function is not differentiable at zero. To this end, a smooth function $\phi_\tau(u, \varepsilon)$ is used to approximate it by using (16) and (17), where ε is a sufficiently small parameter, and the function $\phi_\tau(u, \varepsilon)$ is shown in (18):

$$\phi_\tau(u, \varepsilon) = \frac{u + \sqrt{u^2 + 4\varepsilon^2}}{2} + \frac{-\tau u + \sqrt{\tau^2 u^2 + 4\varepsilon^2}}{2}. \quad (18)$$

From (18), it is known that this function has arbitrary-order derivatives. Figure 1 provides a graph with the pinball loss function and its approximation function $\phi_\tau(u, \varepsilon)$ at $\tau=0.5$ and $\varepsilon=10^{-6}$. It can be intuitively seen that the approximation function $\phi_\tau(u, \varepsilon)$ is smooth everywhere when ε is small enough.

Moreover, we also study another approximation function for the pinball loss function that is defined as follows:

$$p_\tau(x, \alpha) = p(x, \alpha) + p(-\tau x, \alpha) = x + \frac{1}{\alpha} \log(1 + e^{-\alpha x}) + \left[-\tau x + \frac{1}{\alpha} \log(1 + e^{\alpha \tau x}) \right],$$

where $\alpha > 0$.

Below, we introduce a smooth twin bounded support vector machine with pinball loss based on the approximation function $\phi_\tau(u, \varepsilon)$. For the other approximation function $p_\tau(x, \alpha)$, the detailed process is presented in the appendix.

3 Smooth twin bounded support vector machine with pinball loss

3.1 Linear case

To overcome the shortcomings of the hinge loss function, pinball loss is introduced into the twin bounded support vector

Fig. 1 Pinball loss function and its smooth approximation function

machine in this paper, and the optimization problems are obtained as follows:

$$\begin{aligned} \min_{w_1, b_1, \xi} \quad & \frac{1}{2} \sum_{i=1}^{m_1} \left(w_1^T x_i^{(1)} + b_1 \right)^2 + \frac{1}{2} c_3 (\|w_1\|^2 + b_1^2) + c_1 \sum_{j=1}^{m_2} \xi_j \\ \text{s.t.} \quad & -\left(w_1^T x_j^{(2)} + b_1 \right) + \xi_j \geq 1, \quad -\left(w_1^T x_j^{(2)} + b_1 \right) - \frac{\xi_j}{\tau_2} \leq 1, \end{aligned} \quad (19)$$

$$\begin{aligned} \min_{w_2, b_2, \eta} \quad & \frac{1}{2} \sum_{q=1}^{m_2} \left(w_2^T x_q^{(2)} + b_2 \right)^2 + \frac{1}{2} c_4 (\|w_2\|^2 + b_2^2) + c_2 \sum_{p=1}^{m_1} \eta_p \\ \text{s.t.} \quad & \left(w_2^T x_p^{(1)} + b_2 \right) + \eta_p \geq 1, \quad \left(w_2^T x_p^{(1)} + b_2 \right) - \frac{\eta_p}{\tau_1} \leq 1. \end{aligned} \quad (20)$$

For (19) and (20), the solutions can be obtained by solving the quadratic programming problems in the dual space. The required classification surfaces are further obtained. This method is referred to as Pin-GTBSVM. In the following, we mainly introduce the iterative solutions of (19) and (20) in the original space.

First, we convert (19) and (20) into unconstrained optimization problems and obtain the following model:

$$\begin{aligned} \min_{w_1, b_1} \Phi_1(w_1, b_1; \tau_2) = \min_{w_1, b_1} \quad & \frac{1}{2} \sum_{i=1}^{m_1} \left(w_1^T x_i^{(1)} + b_1 \right)^2 + \frac{1}{2} c_3 (\|w_1\|^2 + b_1^2) \\ & + c_1 \sum_{j=1}^{m_2} L_{\tau_2} \left(w_1^T x_j^{(2)} + b_1 + 1 \right) \end{aligned} \quad (21)$$

$$\begin{aligned} \min_{w_2, b_2} \Phi_1(w_2, b_2; \tau_1) = \min_{w_2, b_2} \quad & \frac{1}{2} \sum_{q=1}^{m_2} \left(w_2^T x_q^{(2)} + b_2 \right)^2 + \frac{1}{2} c_4 (\|w_2\|^2 + b_2^2) \\ & + c_2 \sum_{p=1}^{m_1} L_{\tau_1} \left(1 - w_2^T x_p^{(1)} - b_2 \right) \end{aligned} \quad (22)$$

It can be seen that the objective functions of optimization problems (21) and (22) are continuous but not smooth. For this reason, we use the smooth function $\phi(\cdot, \cdot)$ to approximate $L_{\tau_2}(\cdot)$, namely,

$$\phi_{\tau_2} \left(w_1^T x_j^{(2)} + b_1 + 1, \varepsilon \right) \approx L_{\tau_2} \left(w_1^T x_j^{(2)} + b_1 + 1 \right), \quad (23)$$

$$\phi_{\tau_1} \left(1 - w_2^T x_p^{(1)} - b_2, \varepsilon \right) \approx L_{\tau_1} \left(1 - w_2^T x_p^{(1)} - b_2 \right). \quad (24)$$

Here, we rewrite

$$\phi_1(w_1, b_1; \tau_2, \varepsilon) = \phi_{\tau_2} \left(w_1^T x_j^{(2)} + b_1 + 1, \varepsilon \right), \quad (25)$$

$$\phi_2(w_2, b_2; \tau_1, \varepsilon) = \phi_{\tau_1} \left(1 - w_2^T x_p^{(1)} - b_2, \varepsilon \right). \quad (26)$$

We replace the last terms of (21) and (22) with (25) and (26), respectively, to obtain the following model:

$$\min_{w_1, b_1} \Phi(w_1, b_1; \tau_2, \varepsilon) = \min_{w_1, b_1} c_1 \sum_{j=1}^{m_2} \phi_1(w_1, b_1; \tau_2, \varepsilon) + \frac{1}{2} \sum_{i=1}^{m_1} (w_1^T x_i^{(1)} + b_1)^2 + \frac{1}{2} c_3 (\|w_1\|^2 + b_1^2) \quad (27)$$

$$\min_{w_2, b_2} \Phi(w_2, b_2; \tau_1, \varepsilon) = \min_{w_2, b_2} c_2 \sum_{p=1}^{m_1} \phi_2(w_2, b_2; \tau_1, \varepsilon) + \frac{1}{2} \sum_{q=1}^{m_2} (w_2^T x_q^{(2)} + b_2)^2 + \frac{1}{2} c_4 (\|w_2\|^2 + b_2^2) \quad (28)$$

where $c_i > 0 (i = 1, 2, 3, 4)$.

3.2 Nonlinear case

In the nonlinear case, by introducing a kernel function that map the input space to the feature space, a nonlinear problem can be converted into a linear problem in the feature space. In the feature space, the two nonparallel classification hyperplanes to be found are $K(x^T, C^T)u_1 + b_1 = 0$ and $K(x^T, C^T)u_2 + b_2 = 0$, where $C^T = [X^{(1)T} X^{(2)T}]^T$, $X^{(1)} = [x_1^{(1)}, x_2^{(1)}, \dots, x_{m_1}^{(1)}]$, and $X^{(2)} = [x_1^{(2)}, x_2^{(2)}, \dots, x_{m_2}^{(2)}]$. The corresponding optimization problems are as follows:

$$\min_{u_1, b_1, \xi} \frac{1}{2} \sum_{i=1}^{m_1} (K(x_i^{(1)T}, C^T)u_1 + b_1)^2 + \frac{1}{2} c_3 (\|u_1\|^2 + b_1^2) + c_1 \sum_{j=1}^{m_2} \xi_j \quad \text{s.t.} \quad - (K(x_j^{(1)T}, C^T)u_1 + b_1) + \xi_j \geq 1, \quad (29)$$

$$- (K(x_j^{(1)T}, C^T)u_1 + b_1) - \frac{\xi_j}{\tau_2} \leq 1,$$

$$\min_{u_2, b_2, \eta} \frac{1}{2} \sum_{q=1}^{m_2} (K(x_q^{(2)T}, C^T)u_2 + b_2)^2 + \frac{1}{2} c_4 (\|u_2\|^2 + b_2^2) + c_2 \sum_{p=1}^{m_1} \eta_p \quad \text{s.t.} \quad (K(x_p^{(1)T}, C^T)u_2 + b_2) + \eta_p \geq 1, \quad (30)$$

$$(K(x_p^{(1)T}, C^T)u_2 + b_2) - \frac{\eta_p}{\tau_1} \leq 1.$$

Similar to the linear case, the following model is obtained after the smooth and unconstrained processing of (29) and (30):

$$\min_{u_1, b_1} \Phi(u_1, b_1; \tau_2, \varepsilon) = \min_{u_1, b_1} c_1 \sum_{j=1}^{m_2} \phi_1(u_1, b_1; \tau_2, \varepsilon) + \frac{1}{2} \sum_{i=1}^{m_1} (K(x_i^{(1)T}, C^T)u_1 + b_1)^2 + \frac{1}{2} c_3 (\|u_1\|^2 + b_1^2) \quad (31)$$

$$\min_{u_2, b_2} \Phi(u_2, b_2; \tau_1, \varepsilon) = \min_{u_2, b_2} c_2 \sum_{p=1}^{m_1} \phi_2(u_2, b_2; \tau_1, \varepsilon) + \frac{1}{2} \sum_{q=1}^{m_2} (K(x_q^{(2)T}, C^T)u_2 + b_2)^2 + \frac{1}{2} c_4 (\|u_2\|^2 + b_2^2) \quad (32)$$

where $c_i > 0 (i = 1, 2, 3, 4)$.

From Eqs. (27)–(28) and (31)–(32), although the objective functions are different, the solutions are similar. Therefore, for optimization problem (27), the Newton-Armijo iteration method is used below to provide an algorithm for solving the nonparallel hyperplane, and it is denoted as Pin-SGTBSVM.

Pin-SGTBSVM Algorithm:

- Step 1: Given the initial point $(w_1^0, b_1^0) \in R^{n+1}$ and error η ;
 Step 2: If $\|\nabla \Phi(w_1^i, b_1^i; \tau_2, \varepsilon)\| < \eta$ or the required number of iterations is satisfied, then (w_1^i, b_1^i) is considered to be the solution, and the algorithm ends; otherwise, the Newton direction d^i should be recalculated by solving the equations $\nabla^2 \Phi(w_1^i, b_1^i; \tau_2, \varepsilon) d^i = -\nabla \Phi(w_1^i, b_1^i; \tau_2, \varepsilon)$;
 Step 3: w_1^{i+1}, b_1^{i+1} can be calculated by choosing an appropriate step size $\lambda_i = \max\{1, \frac{1}{2}, \frac{1}{4}, \dots\}$ according to $(w_1^{i+1}, b_1^{i+1}) = (w_1^i, b_1^i) + \lambda_i d^i$, where the constraints $\Phi(w_1^i, b_1^i; \tau_2, \varepsilon) - \Phi((w_1^i, b_1^i) + \lambda_i d^i; \tau_2, \varepsilon) \geq -\delta \lambda_i \nabla \Phi(w_1^i, b_1^i; \tau_2, \varepsilon) d^i$, $\delta \in [0, \frac{1}{2}]$ should be met and δ is fixed at $1/2$ here.

For an unknown sample x , it is assigned to class $i (i = +1, -1)$ by using the following method:

$$\text{class}(x) = \text{sgn} \left(\frac{w_1^T x + b_1}{\|w_1\|} + \frac{w_2^T x + b_2}{\|w_2\|} \right) \text{ or}$$

$$\text{class}(x) = \text{sgn} \left(\frac{K(x^T, C^T)u_1 + b_1}{\|u_1\|} + \frac{K(x^T, C^T)u_2 + b_2}{\|u_2\|} \right),$$

where sgn denotes the sign function.

4 Convergence of the algorithm

In section 3, the Pin-SGTBSVM algorithm is used to obtain the solution sequences $\{(w_1^i, b_1^i)\}$ and $\{(w_2^i, b_2^i)\}$. This section mainly studies the convergence of the sequence.

Theorem 1 Given the functions $\Phi(w_1, b_1; \tau_2, \varepsilon)$, $\Phi(w_2, b_2; \tau_1, \varepsilon)$, $\Phi_1(w_1, b_1; \tau_2)$, and $\Phi_1(w_2, b_2; \tau_1)$ respectively, the following conclusions are established:

- (1) $\Phi(w_1, b_1; \tau_2, \varepsilon)$ and $\Phi(w_2, b_2; \tau_1, \varepsilon)$ are arbitrary, continuous and smooth functions with respect to w_1, b_1 and w_2, b_2 for arbitrary $(w_1, b_1) \in R^{n+1}$ and $(w_2, b_2) \in R^{n+1}$.
- (2) For arbitrary $(w_1, b_1) \in R^{n+1}$ and $(w_2, b_2) \in R^{n+1}$, when $\varepsilon > 0$, we have

$$\Phi_1(w_1, b_1; \tau_2) \leq \Phi(w_1, b_1; \tau_2, \varepsilon) \leq \Phi_1(w_1, b_1; \tau_2) + 2c_1 m_2 \varepsilon,$$

$$\Phi_1(w_2, b_2; \tau_1) \leq \Phi(w_2, b_2; \tau_1, \varepsilon) \leq \Phi_1(w_2, b_2; \tau_1) + 2c_2 m_1 \varepsilon.$$

- (3) $\Phi(w_1, b_1; \tau_2, \varepsilon)$ and $\Phi(w_2, b_2; \tau_1, \varepsilon)$ are continuously differentiable and strictly convex functions for any arbitrary $\varepsilon > 0$.

Proof:

- (1) According to the smoothness of the constructed function, conclusion (1) is obviously true.
 (2) We bound the difference between the smooth function and pinball loss function:

$$\phi_\tau(u, \varepsilon) - L_\tau(u) = \begin{cases} \frac{u + \sqrt{u^2 + 4\varepsilon^2}}{2} + \frac{-\tau u + \sqrt{\tau^2 u^2 + 4\varepsilon^2}}{2} - u, & u \geq 0, \\ \frac{u + \sqrt{u^2 + 4\varepsilon^2}}{2} + \frac{-\tau u + \sqrt{\tau^2 u^2 + 4\varepsilon^2}}{2} - (-\tau u), & u < 0. \end{cases}$$

Then, we can obtain

$$\phi_\tau(u, \varepsilon) - L_\tau(u) = \begin{cases} \frac{\sqrt{u^2 + 4\varepsilon^2} - u}{2} + \frac{\sqrt{\tau^2 u^2 + 4\varepsilon^2} - \tau u}{2}, & u \geq 0, \\ \frac{\sqrt{u^2 + 4\varepsilon^2} + u}{2} + \frac{\sqrt{\tau^2 u^2 + 4\varepsilon^2} + \tau u}{2}, & u < 0, \end{cases}$$

so $0 \leq \phi_\tau(u, \varepsilon) - L_\tau(u) \leq 2\varepsilon$.

As a result, $\Phi_1(w_1, b_1; \tau_2) \leq \Phi(w_1, b_1; \tau_2, \varepsilon) \leq \Phi_1(w_1, b_1; \tau_2) + 2c_1 m_2 \varepsilon$.

In the same way, we can obtain

$$\Phi_1(w_2, b_2; \tau_1) \leq \Phi(w_2, b_2; \tau_1, \varepsilon) \leq \Phi_1(w_2, b_2; \tau_1) + 2c_2 m_1 \varepsilon$$

- (3) From (1), $\Phi(w_1, b_1; \tau_2, \varepsilon)$ is a continuous and smooth function, so we see that

$$\nabla \Phi(w_1, b_1; \tau_2, \varepsilon) = \begin{pmatrix} \frac{\partial \Phi(w_1, b_1; \tau_2, \varepsilon)}{\partial w_1} \\ \frac{\partial \Phi(w_1, b_1; \tau_2, \varepsilon)}{\partial b_1} \end{pmatrix}.$$

For $\Phi(w_1, b_1; \tau_2, \varepsilon)$, its first-order partial derivatives with respect to w_1 and b_1 are as follows:

$$\begin{aligned} \frac{\partial \Phi(w_1, b_1; \tau_2, \varepsilon)}{\partial w_1} &= \frac{c_1}{2} \sum_{j=1}^{m_2} \left(x_j^{(2)} + \frac{x_j^{(2)} (w_1^T x_j^{(2)} + b_1 + 1)}{\sqrt{(w_1^T x_j^{(2)} + b_1 + 1)^2 + 4\varepsilon^2}} + \left(-\tau_2 x_j^{(2)} + \frac{\tau_2^2 x_j^{(2)} (w_1^T x_j^{(2)} + b_1 + 1)}{\sqrt{\tau_2^2 (w_1^T x_j^{(2)} + b_1 + 1)^2 + 4\varepsilon^2}} \right) \right) \\ &\quad + \sum_{i=1}^{m_1} x_i^{(1)} (w_1^T x_i^{(1)} + b_1) + c_3 w_1, \\ \frac{\partial \Phi(w_1, b_1; \tau_2, \varepsilon)}{\partial b_1} &= \frac{c_1}{2} \sum_{j=1}^{m_2} \left(1 + \frac{(w_1^T x_j^{(2)} + b_1 + 1)}{\sqrt{(w_1^T x_j^{(2)} + b_1 + 1)^2 + 4\varepsilon^2}} + \left(-\tau_2 + \frac{\tau_2^2 (w_1^T x_j^{(2)} + b_1 + 1)}{\sqrt{\tau_2^2 (w_1^T x_j^{(2)} + b_1 + 1)^2 + 4\varepsilon^2}} \right) \right) \\ &\quad + \sum_{i=1}^{m_1} (w_1^T x_i^{(1)} + b_1) + c_3 b_1. \end{aligned}$$

We further obtain

$$\nabla^2 \Phi(w_1, b_1; \tau_2, \varepsilon) = \begin{pmatrix} \frac{\partial^2 \Phi(w_1, b_1; \tau_2, \varepsilon)}{\partial w_1^2} & \frac{\partial^2 \Phi(w_1, b_1; \tau_2, \varepsilon)}{\partial b_1 \partial w_1} \\ \frac{\partial^2 \Phi(w_1, b_1; \tau_2, \varepsilon)}{\partial w_1 \partial b_1} & \frac{\partial^2 \Phi(w_1, b_1; \tau_2, \varepsilon)}{\partial b_1^2} \end{pmatrix}.$$

The second-order partial derivatives of the function $\Phi(w_1, b_1; \tau_2, \varepsilon)$ with respect to w_1 and b_1 are

$$\nabla^2 \Phi(w_1, b_1; \tau_2, \varepsilon) = \begin{bmatrix} \sum_{j=1}^{m_2} x_j^{(2)} x_j^{(2)T} \lambda_1(w_1, b_1; \tau_2, \varepsilon) + \sum_{i=1}^{m_1} x_i^{(1)} x_i^{(1)T} + c_3 I_n & \sum_{j=1}^{m_2} x_j^{(2)} \lambda_1(w_1, b_1; \tau_2, \varepsilon) + \sum_{i=1}^{m_1} x_i^{(1)} \\ \sum_{j=1}^{m_2} x_j^{(2)T} \lambda_1(w_1, b_1; \tau_2, \varepsilon) + \sum_{i=1}^{m_1} x_i^{(1)T} & \sum_{j=1}^{m_2} \lambda_1(w_1, b_1; \tau_2, \varepsilon) + m_1 + c_3 \end{bmatrix}$$

where

$$\lambda_1(w_1, b_1; \tau_2, \varepsilon) = \frac{c_1}{2} \left(\frac{4\varepsilon^2}{\left((w_1^T x_j^{(2)} + b_1 + 1)^2 + 4\varepsilon^2 \right)^{3/2}} + \frac{\tau_2^2 4\varepsilon^2}{\left(\tau_2^2 (w_1^T x_j^{(2)} + b_1 + 1)^2 + 4\varepsilon^2 \right)^{3/2}} \right).$$

It can be seen that $\lambda_1(w_1, b_1; \tau_2, \varepsilon) > 0$. For arbitrary $\xi_1^T = (\xi^T, \xi_0) \in R^{n+1}$ and $\xi_1 \neq 0, \xi \in R^n$, we have

$$\begin{aligned} \xi_1^T \nabla^2 \Phi(w_1, b_1; \tau_2, \varepsilon) \xi_1 &= \xi^T \left(\sum_{j=1}^{m_2} x_j^{(2)} x_j^{(2)T} \lambda_1(w_1, b_1; \tau_2, \varepsilon) + \sum_{i=1}^{m_1} x_i^{(1)} x_i^{(1)T} + c_3 I_n \right) \xi \\ &+ \xi_0 \left(\sum_{j=1}^{m_2} x_j^{(2)T} \lambda_1(w_1, b_1; \tau_2, \varepsilon) + \sum_{i=1}^{m_1} x_i^{(1)T} \right) \xi \\ &+ \xi^T \left(\sum_{j=1}^{m_2} x_j^{(2)} \lambda_1(w_1, b_1; \tau_2, \varepsilon) + \sum_{i=1}^{m_1} x_i^{(1)} \right) \xi_0 \\ &+ \xi_0 \left(\sum_{j=1}^{m_2} \lambda_1(w_1, b_1; \tau_2, \varepsilon) + m_1 + c_3 \right) \xi_0 \\ &= \lambda_1(w_1, b_1; \tau_2, \varepsilon) \sum_{j=1}^{m_2} (x_j^{(2)T} \xi + \xi_0)^2 + \sum_{i=1}^{m_1} (x_i^{(1)T} \xi + \xi_0)^2 + c_3 \xi_1^T \xi_1 > 0. \end{aligned}$$

This shows that $\nabla^2 \Phi(w_1, b_1; \tau_2, \varepsilon)$ is a positive definite matrix. Therefore, we can obtain $\Phi(w_1, b_1; \tau_2, \varepsilon)$ is a continuously differentiable and strictly convex function. For the function $\Phi(w_2, b_2; \tau_1, \varepsilon)$, we can obtain the same conclusion.

Theorem 2 The Pin-SGTBSVM algorithm converges to the solutions of problems (21) and (22), that is, when $\{(w_1^i, b_1^i)\}$ and $\{(w_2^i, b_2^i)\}$ are the solution sequences of (27) and (28), respectively, the minimum points of (27) and (28), which are (w_1^*, b_1^*) and (w_2^*, b_2^*) , yield $\lim_{k \rightarrow \infty} (w_1^i, b_1^i) = (w_1^*, b_1^*)$ and $\lim_{k \rightarrow \infty} (w_2^i, b_2^i) = (w_2^*, b_2^*)$.

Proof:

According to Theorem 1, $\Phi(w_1, b_1; \tau_2, \varepsilon)$ and $\Phi(w_2, b_2; \tau_1, \varepsilon)$ are continuous and differentiable, strictly convex functions; therefore, there is a unique minimum because $0 \leq \Phi(w_1, b_1; \tau_2, \varepsilon) - \Phi_1(w_1, b_1; \tau_2) \leq 2c_1 m_2 \varepsilon$ and $0 \leq \Phi(w_2, b_2; \tau_1, \varepsilon) - \Phi_1(w_2, b_2; \tau_1) \leq 2c_2 m_1 \varepsilon$, so the sequences $\{(w_1^i, b_1^i)\}$ and $\{(w_2^i, b_2^i)\}$ converge to (w_1^*, b_1^*) and (w_2^*, b_2^*) , respectively.

5 Experimental results and analysis

To validate the performance of the proposed Pin-SGTBSVM algorithm, we select German, Haberman, CMC, Fertility, WPBC, Ionosphere, and Live-disorders from the UCI database [32] for experimentation. For a convenient comparison, some representative algorithms, including the TWSVM [9], TBSVM [10], Pin-GTWSVM [21] (TWSVM based on pinball loss and solved in the dual space), Pin-GTBSVM (TBSVM based on pinball loss and solved in the dual space),

and Pin-SGTWSVM (TWSVM based on pinball loss and solved in the original space), are selected. The kernel function used in the experiments is $K(x, y) = \exp(-\theta \times \|x - y\|^2)$, where θ is a parameter in the kernel function. A tenfold cross-validation method is used to find the optimal parameter value in the range of $[10^{-6}, 10^5]$. The values of τ_1 and τ_2 are 0.5, 0.8, 1, and the value range of $c_i > 0$ ($i = 1, 2, 3, 4$) is $[2^{-10}, 2^{10}]$. The value of ε in the experiments is 10^{-6} , the value of η in the algorithm is 10^{-4} , and the experimental results are the average accuracy and standard deviation. All experiments in this paper are run on an Intel (R) Core (TM) i7-5500U CPU @ 2.40 GHz with 4 GB of memory and MATLAB R2016a.

5.1 UCI datasets

In this subsection, we compare the performance of the presented TWSVM, TBSVM, and Pin-GTWSVM algorithms with those of the traditional solving method and the Pin-GTBSVM, Pin-SGTWSVM and Pin-SGTBSVM solved by the iterative method. The experimental results are shown in Table 1. The accuracy (%) and standard deviation are used for evaluation, abbreviated as Acc and sd, respectively. It can be seen that the Pin-SGTBSVM algorithm is superior to the other five methods on five of the seven datasets. It obtains the same result as Pin-SGTWSVM on the Haberman dataset. And on the Fertility dataset, it obtains the same result as those of the other five algorithms. In addition, the Pin-SGTWSVM algorithm achieves the higher accuracy on six of the seven datasets, but its accuracy on the German dataset is inferior to that of the TWSVM algorithm.

To further analyze the influence of parameters on the algorithms, we selected 6 datasets to conduct experiments with the Pin-GTWSVM, Pin-GTBSVM, Pin-SGTWSVM algorithms for different values of τ . As shown in Fig. 2, although these algorithms use the pinball loss function, for the Pin-SGTBSVM and Pin-SGTWSVM algorithms, when the parameters take different values, there exist certain impacts on the classification accuracy; for the Pin-GTBSVM and Pin-GTWSVM algorithms, the influence of the parameter values on the classification accuracy is small.

Moreover, Gaussian characteristic noise with mean value 0 and variance σ^2 are added to the selected UCI datasets according to the ratio $r = 5\%$ and 10% to test the noise sensitivity of the

Table 1 Performance of different algorithms on UCI datasets with nonlinear kernel

Datasets	TWSVM	TBSVM	Pin-GTWSVM	Pin-GTBSVM	Pin-SGTWSVM	Pin-SGTBSVM
	Acc $\pm$ sd Time (s)	Acc $\pm$ sd Time (s)	Acc $\pm$ sd Time (s)	Acc $\pm$ sd Time (s)	Acc $\pm$ sd Time (s)	Acc $\pm$ sd Time (s)
German	74.9169 ± 0.0191 0.0877	70.0997 ± 0.0000 0.0561	70.0997 ± 0.0000 0.0862	70.0997 ± 0.0000 0.0740	71.3953 ± 0.0256 0.0296	75.1495 ± 0.0162 0.0373
Haberman	73.3696 ± 0.0000 0.0297	73.3696 ± 0.0000 0.0328	73.3696 ± 0.0000 0.0279	73.3696 ± 0.0000 0.0309	73.7500 ± 0.0058 0.0239	73.7500 ± 0.0058 0.0183
CMC	69.2388 ± 0.0395 0.0697	65.3979 ± 0.0000 0.0571	65.3979 ± 0.0000 0.0632	65.3979 ± 0.0000 0.0496	71.6609 ± 0.0361 0.0175	73.7024 ± 0.0223 0.0398
Fertility	87.0968 ± 0.0000 0.0142	87.0968 ± 0.0000 0.0119	87.0968 ± 0.0000 0.0139	87.0968 ± 0.0000 0.0145	87.0968 ± 0.0000 0.0147	87.0968 ± 0.0000 0.0099
WPBC	76.2712 ± 0.0000 0.0146	76.2712 ± 0.0000 0.0199	76.2712 ± 0.0000 0.0294	76.2712 ± 0.0000 0.0293	78.8136 ± 0.0351 0.0195	79.1525 ± 0.0431 0.0208
Ionosphere	91.6981 ± 0.0255 0.0255	91.6981 ± 0.0132 0.0245	93.5849 ± 0.0326 0.0350	92.7358 ± 0.0296 0.0368	93.1132 ± 0.0214 0.0176	94.5283 ± 0.0139 0.0259
Live-disorders	64.4231 ± 0.0493 0.0636	62.6923 ± 0.0459 0.0591	57.6923 ± 0.0000 0.0504	57.6923 ± 0.0000 0.0600	66.8270 ± 0.0482 0.0163	67.1154 ± 0.0575 0.0175

algorithms, and σ^2 is obtained as the variance of different features multiplied by the ratio r . The experimental results are shown in Table 2, where Acc and sd are the accuracy (%) and standard deviation, respectively. When the noise ratio is 5%, the proposed algorithm achieves the best classification performance on five of the seven datasets. However, its classification performance on the German dataset is worse than the TWSVM algorithm but better than that of the TBSVM algorithm. A similar conclusion can be reached when the noise ratio is 10%.

To further validate the anti-noise performance of the proposed algorithm, the real-world dataset Waveform (with noise) from UCI is selected for the experiment. It includes Waveform (v1) and Waveform (v2). The number of samples is 5000, where each contains 21(v1) or 40(v2) features that take real numbers and belong to three categories. For the Waveform (v1) dataset, Gaussian noise is added to each feature with a mean of 0 and variance of 1. For the Waveform (v2) dataset, 19 features of Gaussian noise with a mean of 0 and variance of 1 are added to the Waveform (v1) dataset. As the dataset is a triple-class problem, two categories of samples are taken from the dataset each time, and ten experiments are averaged in three combinations. The experimental results are shown in Table 3.

From Table 3, it can be seen that the proposed algorithms achieve the best classification performance on both the

Waveform (v1) and Waveform (v2) datasets. It is worth mentioning that these algorithms take much different amounts of time to classify the datasets. For the proposed algorithm and the Pin-SGTWSVM algorithm using the iterative technique, on the two datasets, the time spent is 28.4918 s / 31.0971 s, and 35.8810 s / 35.0951 s, respectively. The main reason for this is that the proposed algorithm solves linear equations instead of solving the quadratic programming problems.

5.2 Friedman test

In the following, we mainly use the Friedman test to evaluate the classification performance of the proposed Pin-SGTBSVM algorithm on the UCI datasets. The Friedman test is a statistical test method, and it is a nonparametric test method that uses ranks to determine whether there are significant differences between multiple population distributions. This method can make full use of the information in the samples and is suitable for comparisons of different classifiers on many datasets.

First, the null hypothesis and the opposite hypothesis are given:

H_0 : There is no significant difference among the 6 classifiers;

Fig. 2 Influence of different parameter values on the algorithms, (a) CMC, (b) Ionosphere, (c) German, (d) WPBC, (e) Haberman, (f) Live-disorders

H_1 : There are significant differences among the 6 classifiers.

We discuss k classifiers ($k=6$) on N datasets ($N=7$) for comparison purposes. The corresponding classifiers are sorted according to their accuracy on each dataset, and the classifier corresponding to the highest accuracy has the smallest rank r_i . Table 4 shows the average ranks of different algorithms on the UCI datasets.

In the Friedman test, the chi-square distribution is

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

where $R_j = \frac{1}{N} \sum_{i=1}^N r_{ij}$. Its degree of freedom is $(k-1)$. In addition, on the basis of the Friedman test, the F distribution is further used for testing, where the F distribution is

Table 2 Performance of different algorithms on datasets with noise

Datasets	TWSVM Acc±sd Time (s)	TBSVM Acc±sd Time (s)	Pin- GTWSVM Acc±sd Time (s)	Pin- GTBSVM Acc±sd Time (s)	Pin- SGTWSVM Acc±sd Time (s)	Pin- SGTBSVM Acc±sd Time (s)
German ($r = 5\%$)	73.4884 ±0.0177	70.8970 ±0.0129	70.0997 ±0.0000	70.0997 ±0.0000	71.9601 ±0.0191	72.9900 ±0.0184
($r = 10\%$)	0.0781	0.1820	0.0827	0.0681	0.0342	0.0305
	72.6910 ±0.0126	71.0299 ±0.0119	70.0997 ±0.0000	70.0997 ±0.0000	70.3654 ±0.0156	73.3555 ±0.0187
	0.0851	0.1341	0.0700	0.2318	0.0304	0.0287
Haberman ($r = 5\%$)	73.3696 ±0.0000	73.3696 ±0.0000	73.3696 ±0.0000	73.3696 ±0.0000	73.5326 ±0.0367	73.6104 ±0.0037
($r = 10\%$)	0.0289	0.0330	0.0265	0.0338	0.0170	0.0109
	73.3696 ±0.0000	73.3696 ±0.0000	73.3696 ±0.0000	73.3696 ±0.0000	73.5870 ±0.0458	73.6957 ±0.0038
	0.0288	0.0325	0.0305	0.0306	0.0158	0.0206
CMC ($r = 5\%$)	69.6886 ±0.0281	65.3979 ±0.0000	65.3979 ±0.0000	65.3979 ±0.0000	69.0969 ±0.0345	72.1456 ±0.0201
($r = 10\%$)	0.0660	0.0563	0.0424	0.0414	0.0257	0.0403
	70.6921 ±0.0295	65.3979 ±0.0000	65.3979 ±0.0000	65.3979 ±0.0000	68.2699 ±0.0454	71.9723 ±0.0176
	0.0645	0.2262	0.0398	0.0535	0.0211	0.0270
Fertility ($r = 5\%$)	87.0968 ±0.0000	87.0968 ±0.0000	87.0968 ±0.0000	87.0968 ±0.0000	87.0968 ±0.0000	87.0968 ±0.0000
($r = 10\%$)	0.0086	0.0096	0.0140	0.0137	0.0114	0.0151
	87.0968 ±0.0000	87.0968 ±0.0000	87.0968 ±0.0000	87.0968 ±0.0000	87.0968 ±0.0000	87.0968 ±0.0000
	0.0100	0.0205	0.0148	0.0142	0.0130	0.0135
WPBC ($r = 5\%$)	76.2712 ±0.0000	76.2712 ±0.0000	76.2712 ±0.0000	76.2712 ±0.0000	78.3051 ±0.0390	79.4915 ±0.0403
($r = 10\%$)	0.0157	0.0240	0.0270	0.0263	0.0204	0.0290
	75.4237 ±3.5955	76.2712 ±0.0000	76.2712 ±0.0000	76.2712 ±0.0000	80.5085 ±0.0432	81.0169 ±0.0406
	0.0178	0.0228	0.0298	0.0220	0.0208	0.0302
Ionos phere ($r = 5\%$)	90.7574 ±0.0188	91.3207 ±0.0203	92.4528 ±0.0172	91.6556 ±0.0207	90.5660 ±0.0359	91.8868 ±0.0275
($r = 10\%$)	0.0224	0.0273	0.0362	0.0327	0.0166	0.0183
	90.9434 ±0.0299	90.7547 ±0.0277	90.7490 ±0.0320	88.0189 ±0.0400	89.7170 ±0.0337	90.8490 ±0.0218
	0.0279	0.0514	0.0355	0.0535	0.0209	0.0329
Live- disorders ($r = 5\%$)	62.5000 ±0.0363	63.0769 ±0.0445	57.6923 ±0.0000	57.6923 ±0.0000	66.5385 ±0.0473	66.8269 ±0.0489
($r = 10\%$)	0.0550	0.0487	0.0630	0.0888	0.0298	0.0202
	62.8846 ±0.0513	62.9808 ±0.0411	57.6923 ±0.0000	57.6923 ±0.0000	66.6755 ±0.0258	67.0192 ±0.0445
	0.0540	0.0560	0.0779	0.0888	0.0239	0.0216

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

And its degrees of freedom are $(k-1)$ and $(k-1) \times (N-1)$. According to the data in Table 4, we obtain $\chi_F^2 = 16.8556$

and $F_F = 5.5739$. After looking up the table, we know that the critical value of $F(5,30)$ at the confidence level of 0.05 is 2.53, and $5.5739 > 2.53$, so the null hypothesis H_0 is rejected. That is, there are significant differences among the 6 classifiers. The classification performance of Pin-SGTBSVM algorithm on the above 7 datasets is better than those of the other five algorithms.

Table 3 Classification performance of different algorithms on Waveform datasets with noise

Datasets	TWSVM Acc±sd Time (s)	TBSVM Acc±sd Time (s)	Pin- GTWSVM Acc±sd Time (s)	Pin- GTBSVM Acc±sd Time (s)	Pin- SGTWSVM Acc±sd Time (s)	Pin- SGTBSVM Acc±sd Time (s)
Waveform (version 1)	89.5181 ±0.0110 88.5691	87.9075 ±0.0100 84.6578	66.6445 ±0.0000 59.6571	66.6445 ±0.0000 64.2723	88.7209 ±0.0092 35.8810	89.6508 ±0.0101 28.4918
Waveform (version2)	88.5465 ±0.0076 83.4846	88.0432 ±0.0144 75.7846	66.6667 ±0.0000 52.7830	66.6667 ±0.0000 59.1805	88.1612 ±0.0100 35.0951	89.1726 ±0.0150 31.0971

Table 4 Average ranks of different algorithms on UCI datasets

Datasets	TWSVM	TBSVM	Pin- GTWSVM	Pin- GTBSVM	Pin- SGTWSVM	Pin- SGTBSVM
German	2	5	5	5	3	1
Haberman	4.5	4.5	4.5	4.5	1.5	1.5
CMC	3	5	5	5	2	1
Fertility	3.5	3.5	3.5	3.5	3.5	3.5
WPBC	4.5	4.5	4.5	4.5	2	1
Ionosphere	5.5	5.5	2	4	3	1
Live-disorders	3	4	5.5	5.5	2	1
Average Rank	3.7143	4.5714	4.2857	4.5714	2.4286	1.4286

According to the same method, the corresponding average ranks are calculated by using the classification results with different noise ratios from Table 2, as shown in Table 5.

In the same way, two sets of null hypotheses and opposite hypotheses are given as follows:

$H_0(r = 0.05)$: There is no significant difference among the 6 classifiers on the datasets with 5% noise added;

$H_1(r = 0.05)$: There are significant differences among the 6 classifiers on the datasets with 5% noise added.

$H_0(r = 0.1)$: There is no significant difference among the 6 classifiers on the datasets with 10% noise added;

$H_1(r = 0.1)$: There are significant differences among the 6 classifiers on the datasets with 10% noise added.

The test values are calculated according to Table 5 as follows: $\chi^2_{F(r=0.05)} = 14.2857$, $F_{F(r=0.05)} = 4.1379$, $\chi^2_{F(r=0.1)} = 15.5886$, $F_{F(r=0.1)} = 4.8184$. Since the test values 4.1379 and 4.8184 at both noise levels are greater than 2.53, the H_0

($r = 0.05$) and H_0 ($r = 0.1$) hypotheses are rejected. That is, there are significant differences among the 6 classifiers. The classification performance of Pin-SGTBSVM on the selected datasets with different noise is better than those of the other five algorithms.

In addition, we compare the proposed Pin-SGTBSVM and Pin-SGTWSVM algorithms with the TWSVM, TBSVM, Pin-GTWSVM, and Pin-GTBSVM algorithms on 7 datasets, as shown in Fig. 3. It can be seen that the algorithms using the iterative method perform significantly better on 6 noise-free datasets and 5 datasets with noise, where $r = 0$, 0.05 and 0.1 represent no noise, the 5% noise ratio and the 10% noise ratio, respectively.

5.3 Artificial datasets

In this section, six algorithms are tested on two datasets Art1 and Halfmoons [27]. The samples of each class follow a Gaussian distribution. Each class consists of two clusters, each cluster

Table 5 Average ranks of different algorithms on UCI datasets with noise

Noise ratio(r)	TWSVM	TBSVM	Pin-GTWSVM	Pin-GTBSVM	Pin-SGTWSVM	Pin-SGTBSVM
$r = 0.05$	3.5	4.2143	4.2143	4.5	3.0714	1.6428
$r = 0.1$	3.0714	3.7857	4.6427	4.9286	3.0714	1.5

Fig. 3 Comparisons of different algorithms on 7 datasets

contains 100 samples, where positive samples $x_i(y_i = 1, i = 1, 2, \dots, 100) \sim N(\mu_1, \Sigma_1)$, $x_i(y_i = 1, i = 101, 102, \dots, 200) \sim N(\mu_2, \Sigma_1)$, and negative samples $x_j(y_j = -1, j = 1, 2, \dots, 100) \sim N(\mu_3, \Sigma_2)$, $x_j(y_j = -1, j = 101, 102, \dots, 200) \sim N(\mu_4, \Sigma_2)$. Among them, $\mu_1 = [5, 1]$, $\mu_2 = [1, 0]$, $\mu_3 = [0, 5]$, $\mu_4 = [4, 7]$, $\Sigma_1 = [0.5, 0; 0, 4]$, $\Sigma_2 = [1, 0; 0, 1]$. The two types of samples in the Halfmoons dataset consist of two half months, and each half month contains 250 samples. The experimental results are shown in Fig. 4. It can be seen that the computational time of the algorithms that use the iterative method is lower on both datasets than those of the other algorithms for the linear case.

At the same time, we also add 5% and 10% Gaussian characteristic noise to the above two datasets, respectively. The experimental results are shown in Fig. 5. It can be seen that the accuracy of the proposed algorithm on both datasets is higher than those of the other five algorithms.

Fig. 4 Performance of different algorithms with linear kernel on artificial datasets

5.4 NDC datasets

To further study the classification performance of the proposed algorithm on large-scale datasets, experiments are carried out on the NDC [33] datasets, and 5% characteristic noise is added to the generated datasets. Detailed information about the datasets is shown in Table 6. The experimental results are shown in Fig. 6.

It can be seen that the Pin-SGTBSVM algorithm has higher classification accuracy on the NDC datasets than those of the other five algorithms. At the same time, we also analyze the execution time of the algorithms. Experiments show that the running time of the Pin-SGTBSVM and Pin-SGTWSVM algorithms become much lower than those of the other four algorithms as the sizes of the NDC datasets continues to grow. Their time complexity is better than that of the algorithms without using the iterative method, for which the execution time of the Pin-SGTBSVM algorithm is slightly lower than that of the Pin-SGTWSVM algorithm. To clarify the comparison of the time consumed by the six algorithms on the NDC datasets, the logarithms of the execution time for the six algorithms are calculated here, and the experimental results are shown in Fig. 7.

5.5 Toy dataset

In this subsection, we discuss a smooth twin bounded support vector machine model with approximate pinball loss $\phi_\tau(u, \alpha)$ and $p_\tau(u, \alpha)$. The corresponding algorithms using the function $p_\tau(u, \alpha)$ are denoted as Pin-PSGTWSVM ($\alpha = 5$), Pin-

Fig. 5 Performance of different algorithms with Gaussian kernel on artificial datasets, **(a)** Art1, **(b)** Halfmoons

PSGTWSVM ($\alpha = 1000$), Pin-PSGTBSVM ($\alpha = 5$), and Pin-PSGTBSVM ($\alpha = 1000$). In the following, experiments are

carried out on the toy dataset (named “crossplane”), which is seen in Fig. 8, and the experimental results are shown in Table 7.

Table 6 Information about the NDC datasets

Datasets	Number of training samples	Number of testing samples	Number of features
NDC-600	600	60	32
NDC-800	800	80	32
NDC-1 K	1000	100	32
NDC-2 K	2000	200	32
NDC-3 K	3000	300	32

Fig. 6 Classification performance of different algorithms on the NDC datasets, **a** Noise free, **b** With noise

Fig. 7 Comparisons of time performance of different algorithms on NDC datasets, **a** Noise free **b** With noise

Fig. 8 Toy dataset

From Table 7, it can be seen that the accuracy of the Pin-SGTBSVM algorithm is the highest in the linear case, followed by Pin-PSGTWSVM ($\alpha = 5$). In the nonlinear case, the accuracy rates of the Pin-SGTBSVM and Pin-SGTWSVM algorithms are higher than those of the other algorithms. The experimental results show that $\phi_r(u, \alpha)$ is better than $p_r(u, \alpha)$.

6 Conclusion

Aiming at the shortcomings of the twin support vector machine, we introduce the pinball loss function into the twin bounded support vector machine. By constructing a smooth approximation function, a smooth twin bounded support vector machine model with pinball loss is obtained. On this basis,

Table 7 Comparison of performance achieved by different algorithms

	Pin-PSGTWSVM ($\alpha = 5$)	Pin-PSGTWSVM ($\alpha = 1000$)	Pin-PSGTBSVM ($\alpha = 5$)	Pin-PSGTBSVM ($\alpha = 1000$)	Pin-SGTWSVM	Pin-SGTBSVM
Linear	95.7500 ± 0.0182	50.0000 ± 0.0000	95.3333 ± 0.0399	94.0833 ± 0.0405	95.2000 ± 0.0593	95.8333 ± 0.0171
Non Linear	95.0000 ± 0.0118	50.0000 ± 0.0000	94.3333 ± 0.0353	50.0000 ± 0.0000	96.7500 ± 0.0178	96.2530 ± 0.0259

a twin bounded support vector machine algorithm with pinball loss is proposed, and the convergence of the algorithm is theoretically proven. We compare the proposed algorithms with other representative algorithms on the UCI datasets and artificial datasets. Under the premise of ensuring the accuracy, the defects of the twin support vector machine with regard to noise sensitivity and resampling instability are solved, and the time complexity is improved as well, thereby demonstrating the effectiveness of the proposed algorithm.

Acknowledgments The authors are grateful for the supports provided by Natural Science Foundation of Hebei Province (No. F2018201060).

Appendix 1

According to (17), we also obtain another approximation smooth function of the pinball loss function as follows:

$$p_{\tau}(x, \alpha) = p(x, \alpha) + p(-\tau x, \alpha) = x + \frac{1}{\alpha} \log(1 + e^{-\alpha x}) + \left[-\tau x + \frac{1}{\alpha} \log(1 + e^{\alpha \tau x}) \right], \alpha > 0$$

We further apply it to the (23) and (24):

$$p_{\tau_2}(w_1^T x_h^{(2)} + b_1 + 1, \alpha) \approx L_{\tau_2}(w_1^T x_h^{(2)} + b_1 + 1), \\ p_{\tau_1}(1 - w_2^T x_p^{(1)} - b_2, \alpha) \approx L_{\tau_1}(1 - w_2^T x_p^{(1)} - b_2).$$

Here, we rewrite

$$p_1(w_1, b_1; \tau_2, \alpha) = p_{\tau_2}(w_1^T x_h^{(2)} + b_1 + 1, \alpha), \\ p_2(w_2, b_2; \tau_1, \alpha) = p_{\tau_1}(1 - w_2^T x_p^{(1)} - b_2, \alpha).$$

Finally, we can obtain

$$\min_{w_1, b_1} P(w_1, b_1; \tau_2, \alpha) = \min_{w_1, b_1} \frac{1}{2} c_1 \left\| \sum_{h=1}^{m_2} p_1(w_1, b_1; \tau_2, \alpha) \right\|^2 \\ + \frac{1}{2} \sum_{g=1}^{m_1} (w_1^T x_g^{(1)} + b_1)^2 + \frac{1}{2} c_3 (\|w_1\|^2 + b_1^2),$$

$$\min_{w_2, b_2} P(w_2, b_2; \tau_2, \alpha) = \min_{w_2, b_2} \frac{1}{2} c_2 \left\| \sum_{k=1}^{m_1} p_2(w_2, b_2; \tau_1, \alpha) \right\|^2 \\ + \frac{1}{2} \sum_{l=1}^{m_2} (w_2^T x_l^{(2)} + b_2)^2 + \frac{1}{2} c_4 (\|w_2\|^2 + b_2^2).$$

The first partial derivatives of $P(w_1, b_1; \tau_2, \alpha)$ are

$$\frac{\partial P(w_1, b_1; \tau_2, \alpha)}{\partial w_1} = c_1 \sum_{h=1}^{m_2} \left(\frac{x_h^{(2)}}{1 + e^{-\alpha(w_1^T x_h^{(2)} + b_1 + 1)}} - \frac{\tau_2 x_h^{(2)}}{1 + e^{\alpha \tau_2(w_1^T x_h^{(2)} + b_1 + 1)}} \right) \sum_{h=1}^{m_2} p_1(w_1, b_1; \tau_2, \alpha) \\ + \sum_{g=1}^{m_1} x_g^{(1)} (w_1^T x_g^{(1)} + b_1) + c_3 w_1, \\ \frac{\partial P(w_1, b_1; \tau_2, \alpha)}{\partial b_1} = c_1 \sum_{h=1}^{m_2} \left(\frac{1}{1 + e^{-\alpha(w_1^T x_h^{(2)} + b_1 + 1)}} - \frac{1}{1 + e^{\alpha \tau_2(w_1^T x_h^{(2)} + b_1 + 1)}} \right) \sum_{h=1}^{m_2} p_1(w_1, b_1; \tau_2, \alpha) \\ + \sum_{g=1}^{m_1} (w_1^T x_g^{(1)} + b_1) + c_3 b_1$$

The second-order partial derivatives of the function $P(w_1, b_1; \tau_2, \alpha)$ with respect to w_1 and b_1 are

$$\nabla^2 P(w_1, b_1; \tau_2, \alpha) = \begin{bmatrix} \sum_{h=1}^{m_2} x_h^{(2)} x_h^{(2)T} \lambda_2(w_1, b_1; \tau_2, \varepsilon) + \sum_{g=1}^{m_1} x_g^{(1)} x_g^{(1)T} + c_3 I_n & \sum_{h=1}^{m_2} x_h^{(2)} \lambda_2(w_1, b_1; \tau_2, \varepsilon) + \sum_{g=1}^{m_1} x_g^{(1)} \\ \sum_{h=1}^{m_2} x_h^{(2)T} \lambda_2(w_1, b_1; \tau_2, \varepsilon) + \sum_{g=1}^{m_1} x_g^{(1)T} & \sum_{h=1}^{m_2} \lambda_2(w_1, b_1; \tau_2, \varepsilon) + m_1 + c_3 \end{bmatrix},$$

where

$$\lambda_2(w_1, b_1; \tau_2, \alpha) = c_1 \sum_{h=1}^{m_2} \left(\frac{\alpha e^{-\alpha(w_1^T x_h^{(2)} + b_1 + 1)}}{(1 + e^{-\alpha(w_1^T x_h^{(2)} + b_1 + 1)})^2} - \frac{\tau_2}{(1 + e^{\alpha\tau_2(w_1^T x_h^{(2)} + b_1 + 1)})^2} \right) \sum_{h=1}^{m_2} p_1(w_1, b_1; \tau_2, \alpha) \\ + c_1 \sum_{h=1}^{m_2} \left(\frac{1}{1 + e^{-\alpha(w_1^T x_h^{(2)} + b_1 + 1)}} - \frac{1}{1 + e^{\alpha\tau_2(w_1^T x_h^{(2)} + b_1 + 1)}} \right) \sum_{h=1}^{m_2} \left(\frac{1}{1 + e^{-\alpha(w_1^T x_h^{(2)} + b_1 + 1)}} - \frac{1}{1 + e^{\alpha\tau_2(w_1^T x_h^{(2)} + b_1 + 1)}} \right).$$

This paper studies the twin bounded support vector machine with pinball loss and its iterative solution in the original space. The main innovations are as follows:

1. The pinball loss function is introduced into the twin bounded support vector machine, and the twin bounded support vector machine with pinball loss is proposed
2. We research two methods for solving the model:
 - (1) Using the Lagrangian method, the proposed model is transformed into the dual space for the solution
 - (2) By constructing a smooth approximation function, the non-differentiable problem of the pinball loss function at the zero point is solved, and the iterative method is used to solve the proposed model in the original space, and then the smooth twin bounded support vector machine with pinball loss is proposed. The iterative sequences of the proposed algorithm are theoretically proved in the original space
3. We select the standard datasets and artificial datasets to study the proposed algorithm, and select other representative algorithms for comparisons to validate the effectiveness of the proposed algorithm

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Vapnik VN (1995) The nature of statistical learning theory. Springer-Verlag, New York
2. Joachims T (1998) Text categorization with support vector machines: learning with many relevant features. Proc of the 10th Eur Conf on Mach Learn:137–142
3. Li YQ, Guan CT (2008) Joint feature re-extraction and classification using an iterative semi-supervised support vector machine algorithm. Mach Learn 71(1):33–53
4. Sun S, Xie X, Dong C (2018) Multiview learning with generalized eigenvalue proximal support vector machines. IEEE Trans Cybern 49(2):688–697
5. Xie X, Sun S (2019) Multi-view support vector machines with the consensus and complementarity information. IEEE Trans Knowl Data Eng 99:1–1
6. Zhao J, Xu YT, Fujita H (2019) An improved non-parallel Universum support vector machine and its safe sample screening rule. Knowl-Based Syst 170:79–88
7. Fung G, Mangasarian OL (2001) Proximal support vector machine classifiers. Proceed Seventh ACM SIGKDD Int Conf Knowl Discov Data Minin:77–86
8. Mangasarian OL, Wild EW (2006) Multisurface proximal support vector machine classification via generalized eigenvalues. IEEE Trans Pattern Anal Mach Intell 28(1):69–74
9. Jayadeva RK, Khemchandani R, Chandra S (2007) Twin support vector machine for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
10. Shao YH, Zhang CH, Wang XB et al (2011) Improvements on twin support vector machines. IEEE Trans Neural Netw 22(6):962–968
11. Kumar MA, Gopal M (2009) Least squares twin support vector machines for pattern classification. Expert Syst Appl 36(4):7535–7543
12. Peng XJ (2010) A ν -twin support vector machine(ν -TSVM)classifier and its geometric algorithms. Inf Sci 180(20): 3863–3875
13. Peng XJ (2011) TPMSVM: a novel twin parametric-margin support vector machine for pattern recognition. Pattern Recogn 44(10): 2678–2692

14. Peng XJ, Wang YF, Xu D (2013) Structural twin parametric-margin support vector machine for binary classification. *Knowl-Based Syst* 49:63–72
15. Qi ZQ, Tian YJ, Shi Y (2013) Structural twin support vector machine for classification. *Knowl-Based Syst* 43(1):74–81
16. Rastogi R, Sharma S, Chandra S (2017) Robust parametric twin support vector machine for pattern classification. *Neural Process Lett* 47(1):293–323
17. Gupta D, Richhariya B, Borah P (2019) A fuzzy twin support vector machine based on information entropy for class imbalance learning. *Neural Comput & Applic* 24:7153–7164
18. Chen SG, Cao JF, Huang Z, Shen CS (2019) Entropy-based fuzzy twin bounded support vector machine for binary classification. *IEEE Access* 7:86555–86569
19. Huang XL, Shi L, Suykens JAK (2014) Support vector machine classifier with pinball loss. *IEEE Trans Pattern Anal Mach Intell* 36(5):984–997
20. Xu YT, Yang ZJ, Pan XL (2017) A novel twin support-vector machine with pinball loss. *IEEE Trans Neural Networks Learn Syst* 28(2):359–370
21. Tanveer M, Sharma A, Suganthan PN (2019) General twin support vector machine with pinball loss function. *Inf Sci* 94:311–327
22. Tanveer M, Tiwari A, Choudhary R, Jalan S (2019) Sparse pinball twin support vector machines. *Appl Soft Comput* 78:164–175
23. Lee YJ, Mangasarian OL (2001) SSVM: a smooth support vector machine for classification. *Comput Optim Appl* 20(1):5–22
24. Kumar MA, Gopal M (2008) Application of smoothing technique on twin support vector machines. *Pattern Recogn Lett* 29(13):1842–1848
25. Wang Z, Shao YH, Wu TR (2013) A GA-based model selection for smooth twin parametric-margin support vector machine. *Pattern Recogn* 46(8):2267–2277
26. Ding SF, Huang HJ, Shi ZZ (2014) Weighted smooth CHKS twin support vector machines. *J Softw* 24(11):2548–2557
27. Chen WJ, Shao YH, Hong N (2014) Laplacian smooth twin support vector machine for semi-supervised classification. *Int J Mach Learn Cybern* 5(3):459–468
28. Chen C, Mangasarian OL (1996) A class of smoothing functions for nonlinear and mixed complementarity problems. *Comput Optim Appl* 5(2):97–138
29. Larry A (1966) Minimization of functions having Lipschitz-continuous first partial derivatives. *Pac J Math* 16:1–3
30. Bertsekas D (1999) *Nonlinear programming*, 2nd edn. Athena Scientific, Belmont
31. Dennis JE, Schnabel RB (1983) *Numerical methods for unconstrained optimization and nonlinear equations*. Prentice-Hall, Englewood Cliffs
32. Dua D, Graff C (2019) UCI machine learning repository. University of California, School of Information and Computer Science, Irvine. Available: <http://archive.ics.uci.edu/ml>
33. Musicant DR (1998) NDC: Normally distributed clustered datasets. Available: <https://research.cs.wisc.edu/dmi/svm/ndc/>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Kai Li was born in 1963. He received the Ph.D. degree from Beijing Jiaotong University, China, in 2005. He is working as a professor in School of Cyber Security and Computer, Hebei University, China. His current research interests include machine learning, data mining, computational intelligence, and pattern recognition.