Information Sciences 494 (2019) 311–327

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

General twin support vector machine with pinball loss function

M. Tanveer[a,*], A. Sharma[a], P.N. Suganthan[b,*]

[a] Discipline of Mathematics, Indian Institute of Technology Indore, Simrol, Indore 453552, India
[b] School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Article history:
Received 3 October 2018
Revised 18 April 2019
Accepted 20 April 2019
Available online 26 April 2019

Keywords:
Hinge loss
Pinball loss
Quantile distance
Pin-SVM
Noise insensitivity
TSVM

ABSTRACT

The standard twin support vector machine (TSVM) uses the hinge loss function which leads to noise sensitivity and instability. In this paper, we propose a novel general twin support vector machine with pinball loss (Pin-GTSVM) for solving classification problems. We show that the proposed Pin-GTSVM is noise insensitive and more stable for re-sampling. Further, the computational complexity of the proposed Pin-GTSVM is similar to that of the TSVM. Thus, the pinball loss function does not increase the computation time of the proposed Pin-GTSVM. Numerical experiments with different noise are performed on 17 UCI and KEEL benchmark real-world datasets and the results are compared with other baseline methods. The comparisons clearly show that the proposed Pin-GTSVM has better generalization performance for noise corrupted datasets.

1. Introduction

Support vector machines (SVMs) and its variants are very powerful methodologies for solving classification and regression problems [2,5–7,21,37,38,40,41,44]. Within a few years after its introduction, SVM has been widely applied to many real-world problems, such as face detection [26], text categorization [16], electroencephalogram signal classification [32], web mining [4], remote sensing [27] and feature extraction [22]. SVM tries to reduce the generalization error by maximizing the margin. It formulates a convex quadratic programming problem (QPP), leading to a unique global solution. SVM implements the structural risk minimization (SRM) principle which seeks to minimize an upper bound on the Vapnik-Chervonenkis (VC) dimension.

One of the main drawbacks of standard SVM is its high computational complexity of order $\mathcal{O}(\ell^3)$, where ℓ is the total number of training data points in the classification problem. To reduce the training time of the standard SVM, the concept of two non-parallel hyperplanes was applied in generalized eigenvalue proximal support vector machine (GEPSVM) [24]. Jayadeva et al. [15] modified GEPSVM and proposed a novel twin support vector machine (TSVM) for classification problems. TSVM seeks two non-parallel proximal hyperplanes such that the data points of one class generate the constraints for the other class and vice-versa. TSVM solves two smaller QPPs, instead of one larger QPP, which makes the algorithm approximately four times faster than standard SVM. Tanveer [42] proposed a novel linear programming formulation of exact 1-norm TSVM to improve the robustness and sparsity. Recently, several algorithms have been introduced for regression and classification problems, based on the non-parallel proximal hyperplanes [15,18,23,29,31–35,43]. TSVM and its variants

* Corresponding author.
E-mail addresses: mtanveer@iiti.ac.in (M. Tanveer), msc1603141001@iiti.ac.in (A. Sharma), epnsugan@ntu.edu.sg (P.N. Suganthan).

https://doi.org/10.1016/j.ins.2019.04.032

[3,17,28,30,45,46,48] consider that noise level on the dataset is uniform throughout the domain. However, the assumption of uniform noise on the dataset is not always true. To address this problem, Hao [11] proposed a novel algorithm termed as parametric-margin-ν-support vector machine (par-ν-SVM). Peng [29] proposed an algorithm called twin parametric-margin support vector machine (TPMSVM) which aims to generate two nonparallel hyperplanes such that each one determines the positive or negative parametric-margin hyperplane of the separating hyperplane. Numerical experiments on several artificial and UCI benchmark real-world datasets indicate that the TPMSVM is not only fast, but also shows better generalization performance.

In machine learning, the appropriate loss function is a requirement for designing a robust algorithm. Recently, different margin-based loss functions, such as 0–1 loss, hinge loss, squared loss, etc. have been used in classification and regression problems. In SVM, penalty on the training data points is regulated by hinge loss. However, hinge loss suffers from the shortest distance, which easily leads to noise sensitivity and instability for re-sampling. To address this problem, Huang et al. [14] introduced pinball loss in SVM instead of hinge loss and proposed support vector machine with pinball loss (Pin-SVM). Pinball loss is associated with quantile distance [19,39], which brought noise insensitivity when solving classification problem. Shen et al. [36] proposed truncated pinball loss SVM to present a flexible framework for the trade-off between sparsity and feature noise insensitivity. The theoretical and experimental analysis showed that Pin-SVM is insensitive to feature noise and more stable for re-sampling as compared to the standard SVM. However, Pin-SVM still deals with a large size QPP, with a higher time complexity. To mitigate the higher time complexity, Xu et al. [49] introduce pinball loss function in TPMSVM and proposed a twin support vector machine with pinball loss (Pin-TSVM) to cope with the quantile distance instead of shortest distance, thereby making it noise insensitive and fast.

Motivated by the works of [14,49], we introduce pinball loss function to the standard TSVM [15] and propose a novel general twin support vector machine with pinball loss (Pin-GTSVM) for classification problems. The proposed Pin-GTSVM possesses the following advantages:

- The proposed Pin-GTSVM is less sensitive to noise and more stable for re-sampling.
- The computational complexity of the proposed Pin-GTSVM is equivalent to the TSVM, and thus pinball loss function does not intensify the computation cost of the proposed Pin-GTSVM.
- Pin-GTSVM reduces to the original TSVM when pinball loss parameters in the proposed Pin-GTSVM tend to zero, and thus TSVM could be considered as a particular case of the proposed Pin-GTSVM.
- Pin-GTSVM computes a pair of non-parallel hyperplanes similar to original TSVM. However, Pin-TSVM acquires a pair of parametric-margin hyperplanes the same as in TPMSVM.
- The proposed Pin-GTSVM model is generic and can be used with any TSVM variant.
- Numerical experiments with three different noise levels are performed on 17 UCI and KEEL benchmark real-world datasets and their results are compared with TSVM [15], TPMSVM [29] and Pin-TSVM [49]. The comparative results clearly show the effectiveness and feasibility of the proposed Pin-GTSVM for solving noise corrupted classification problems.

The rest of the paper is organized as follows. We briefly review loss functions, Pin-SVM, TSVM, TPMSVM and Pin-TSVM in Section 2. General twin support vector machine with pinball loss (Pin-GTSVM) is proposed in Section 3. In Section 4, we discuss the properties of the proposed Pin-GTSVM. In Section 5, we compare the proposed Pin-GTSVM to Pin-SVM, TSVM and Pin-TSVM. Numerical experiments on 17 UCI and KEEL benchmark datasets show the effectiveness and feasibility of the proposed Pin-GTSVM in Section 6. Conclusions and future directions are presented in Section 7.

2. Preliminaries

In this section, we provide a brief description of loss functions, Pin-SVM, TSVM, TPMSVM, and Pin-TSVM formulations. For detailed description, the interested readers referred to [14,15,29,49].

2.1. Loss functions

Loss functions are used in machine learning tasks, such as regression and classification. Consider the training set $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_\ell, y_\ell)\}$, where $x_i \in \mathbb{R}^n$ are inputs and $y_i \in Y = \{-1, 1\}$ are the corresponding outputs for $i = 1, 2, \ldots, \ell$. Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is a mapping from $x_i \in \mathbb{R}^n$ to $y_i \in \{-1, 1\}$. Loss function represents the price paid for inaccuracy in classification problems.

2.1.1. 0–1 loss function

It takes the value zero if the predicted class is the same as the true class and one if the predicted class does not match the true class and it is defined by

$$\mathcal{L}(x, y, f(x)) = I(y \neq f(x)), \tag{1}$$

where I is the indicator function.

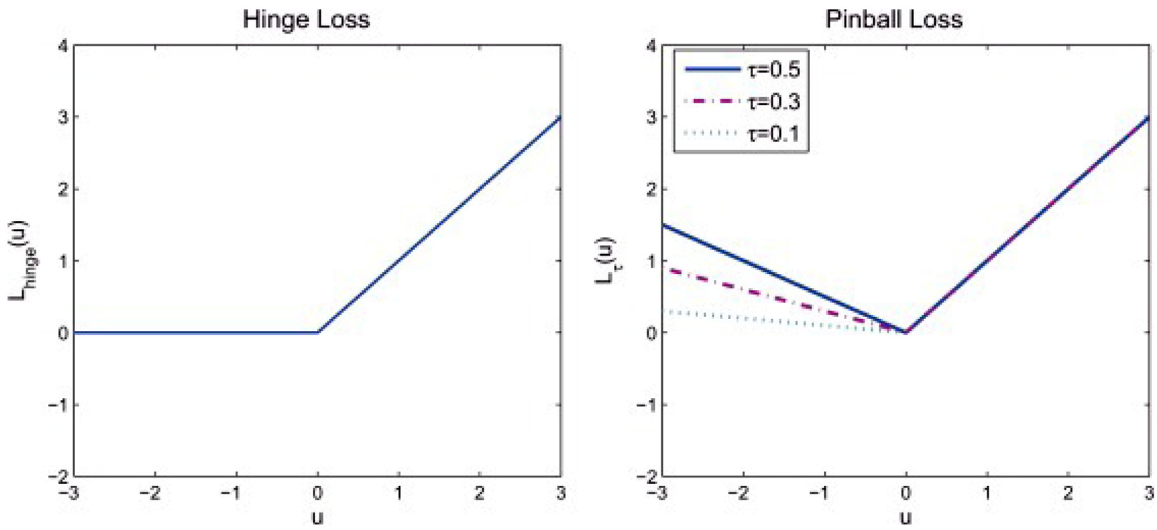


Fig. 1. Hinge and pinball loss function [50].

2.1.2. Hinge loss function

The hinge loss function is defined by

$$\mathcal{L}_{hinge}(x, y, f(x)) = \max(0, 1 - yf(x)). \tag{2}$$

Hinge loss function deals with the shortest distance between classes, which accomplishes noise sensitivity with bad re-sampling.

2.1.3. Pinball loss function

The pinball loss function is defined by

$$\mathcal{L}_\tau(x, y, f(x)) = \begin{cases} -\tau(1 - yf(x)), & 1 - yf(x) < 0, \\ 1 - yf(x), & 1 - yf(x) \geq 0, \end{cases} \tag{3}$$

where $\tau \in [0, 1]$. Pinball loss gives a supplementary penalty to the correctly classified points. Pinball loss is associated with quantile distance.

2.2. Support vector machine with pinball loss (pin-SVM)

Huang et al. [14] introduced pinball loss into the SVM resulting in many favourable properties such as noise insensitivity for classification. Consider the data points to be classified denoted by the set T. To solve the classification problem, we need to find $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ such that

$$\begin{aligned} &w^T\phi(x_i) + b \geq 1 \text{ for } y_i = 1, \\ &w^T\phi(x_i) + b \leq -1 \text{ for } y_i = -1, \end{aligned} \tag{4}$$

where $\phi(\cdot)$ is a nonlinear mapping from the input space to a new feature space. The optimal hyperplane $w^T\phi(x) + b = 0$, lies midway between the supporting hyperplanes given by:

$$w^T\phi(x) + b = 1 \text{ and } w^T\phi(x) + b = -1, \tag{5}$$

and classify the two classes from each other with a margin of $\frac{1}{\|w\|}$ on each side. Data points lying on the supporting hyperplanes given by Eq. (5) are termed as support vectors. The classifier is obtained by maximizing the margin. If the data points of two different classes are not linearly separable in feature space then to separate all the data points correctly, we allow the existence of data points that violate the constraints $y_i(w^T\phi(x_i) + b) \geq 1$ by introducing the pinball loss $\mathcal{L}_\tau(x, y, f(x))$. Finally, we obtain the nonlinear Pin-SVM formulation as follows:

$$\min_{w, b} \; \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{\ell}\mathcal{L}_\tau(x, y, f(x)). \tag{6}$$

After employing the pinball loss function in Eq. (6), we obtain the following quadratic programming problem (QPP) of Pin-SVM:

$$\min_{w, b, \xi} \; \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{\ell}\xi_i$$

$$
\begin{aligned}
s.t.\quad & y_i\left(w^T\phi(x_i)+b\right) \geq 1-\xi_i, \\
& y_i\left(w^T\phi(x_i)+b\right) \leq 1+\frac{\xi_i}{\tau}, \quad i=1,2,\ldots,\ell,
\end{aligned}
\tag{7}
$$

where $\xi = (\xi_1, \xi_2, \ldots, \xi_\ell)^T$ is a slack variable and $c > 0$ is a penalty parameter. The parameter c determines the weight between the two terms $\|w\|^2$ and $\sum_{i=1}^{\ell}\xi_i$. Pin-SVM has an advantage of noise insensitivity. Meanwhile, Pin-SVM has similar time complexity to that of standard SVM.

Note that the second constraint of Eq. (7) becomes $\xi \geq 0$ when $\tau = 0$, and thus Pin-SVM reduces to the hinge loss SVM.

2.3. Twin support vector machine (TSVM)

Both conventional SVM and Pin-SVM have a drawback of high computation complexity, approximately of order $\mathcal{O}(\ell^3)$, where ℓ is the number of data points. To reduce the computational complexity, Jayadeva et al. [15] proposed a variant of SVM termed as twin support vector machine (TSVM) for pattern classification. The formulation of SVM requires all data points. However, in TSVM, they are distributed in such a way that the patterns of one class give the constraints to the other class and vice-versa. So, TSVM solves two smaller QPPs rather than one large QPP.

Consider a binary classification problem with classes +1 and -1. Let the number of data points belonging to the class +1 and -1 be ℓ_1 and ℓ_2, respectively in the n-dimensional real space $\mathbb{R}^n$ represented by matrices A and B respectively. Nonlinear TSVM seeks for two kernel generated surfaces defined as follows:

$$
K\left(x^T, D^T\right)u_+ + b_+ = 0 \text{ and } K\left(x^T, D^T\right)u_- + b_- = 0,
\tag{8}
$$

where $D = [A; B]$; $u_+, u_- \in \mathbb{R}^n$ and K is an arbitrary kernel function. The nonlinear TSVM formulation can be expressed as follows:

$$
\begin{aligned}
\min_{u_+, b_+, \xi_1}\quad & \frac{1}{2}\|K\left(A, D^T\right)u_+ + e_1 b_+\|^2 + c_1 e_2^T \xi_1 \\
s.t.\quad & -\left(K(B, D^T)u_+ + e_2 b_+\right) + \xi_1 \geq e_2, \ \xi_1 \geq 0,
\end{aligned}
\tag{9}
$$

and

$$
\begin{aligned}
\min_{u_-, b_-, \xi_2}\quad & \frac{1}{2}\|K\left(B, D^T\right)u_- + e_2 b_-\|^2 + c_2 e_1^T \xi_2 \\
s.t.\quad & K\left(A, D^T\right)u_- + e_1 b_- + \xi_2 \geq e_1, \ \xi_2 \geq 0,
\end{aligned}
\tag{10}
$$

where c_1, c_2 are positive parameters, e_1, e_2 are vectors of ones of appropriate dimensions and ξ_1, ξ_2 are slack variables. By introducing the Lagrange multipliers $\alpha \geq 0$, $\beta \geq 0$ and using the Karush-Kuhn-Tucker (K.K.T.) [20] optimality conditions, we can derive the duals of Eqs. (9) and (10) as follows:

$$
\begin{aligned}
\max_{\alpha}\quad & e_2^T\alpha - \frac{1}{2}\alpha^T Q(P^T P)^{-1} Q^T \beta \\
s.t.\quad & 0 \leq \alpha \leq c_1
\end{aligned}
\tag{11}
$$

and

$$
\begin{aligned}
\max_{\beta}\quad & e_1^T\beta - \frac{1}{2}\beta^T P(Q^T Q)^{-1} P^T \beta \\
s.t.\quad & 0 \leq \beta \leq c_2,
\end{aligned}
\tag{12}
$$

where $P = [K(A, D^T)\ e_1]$ and $Q = [K(B, D^T)\ e_2]$. Finally solutions of (11) and (12) are given by

$$
\begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = -(P^T P + \delta I)^{-1} Q^T \alpha,
\tag{13}
$$

$$
\begin{bmatrix} u_- \\ b_- \end{bmatrix} = (Q^T Q + \delta I)^{-1} P^T \beta,
\tag{14}
$$

where $\delta I (\delta > 0)$ is a regularization term used to avoid the possible ill-conditioning of matrices $P^T P$ and $Q^T Q$. A new data point $x \in \mathbb{R}^n$ is assigned to class $i\,(i = +1, -1)$ depending on which of the kernel generated surface in Eq. (8) is closer to x, i.e.,

$$
class(i) = \operatorname{sign}\left(\frac{K(x^T, D^T)u_+ + b_+}{\|u_+\|} + \frac{K(x^T, D^T)u_- + b_-}{\|u_-\|}\right),
\tag{15}
$$

where $sign(\cdot)$ is signum function.

2.4. Twin parametric-margin support vector machine (TPMSVM)

In general, we assume that the noise level on training data in SVM and its extensions is uniform throughout the domain. However, the assumption of uniform noise level is not always true. To resolve this problem, Hao [11] proposed a novel parametric-margin-ν-support vector machine (par-ν-SVM). Peng [29] proposed an extension of TSVM, in the spirit of par-ν-SVM, called twin parametric-margin support vector machine (TPMSVM). Similar to the TSVM, nonlinear TPMSVM also derives a pair of kernel generated surfaces given in Eq. (8) around the data points through the following QPPs:

$$\begin{aligned}
&\min_{u_+, b_+, \xi_1} \frac{1}{2}\|u_+\|^2 + \frac{\nu_1}{\ell_2} e_2^T \left(K(B, D^T)u_+ + e_2 b_+\right) + \frac{c_1}{\ell_1} e_1^T \xi_1 \\
&\quad s.t. \quad K(A, D^T)u_+ + e_1 b_+ \geq -\xi_1, \\
&\qquad\quad\;\; \xi_1 \geq 0
\end{aligned} \tag{16}$$

and

$$\begin{aligned}
&\min_{u_-, b_-, \xi_2} \frac{1}{2}\|u_-\|^2 - \frac{\nu_2}{\ell_1} e_1^T \left(K(A, D^T)u_- + e_1 b_-\right) + \frac{c_2}{\ell_2} e_2^T \xi_2 \\
&\quad s.t. \quad -\left(K(B, D^T)u_- + e_2 b_-\right) \geq -\xi_2, \\
&\qquad\quad\;\; \xi_2 \geq 0,
\end{aligned} \tag{17}$$

where $\nu_1, \nu_2 > 0$ are margin parameters. Formulation of TPMSVM is different from TSVM. The first term of the objective function of QPP in Eq. (16) controls the model complexity for obtaining the positive surface, the second term minimizes the projection of negative class data points on positive surface with parameter ν_1, which leads the negative class to be as far as possible from the positive surface. The constraint of QPP in Eq. (16) restricts the projection of positive class data points on the positive surface be not less than zero. Otherwise, a slack variable $\xi_1 \geq 0$ is implemented to measure the error. The last term of the objective function in QPP (16) minimizes the sum of error variables, which endeavor to over-fit the positive data points. The similar justifications are applicable for the QPP in Eq. (17). Similar to nonlinear TSVM, we can find the dual formulation of (16) and (17) by introducing the Lagrange function and using the K.K.T. conditions as follows:

$$\begin{aligned}
&\max_{\alpha} \; \frac{\nu_1}{\ell_2} e_2^T K(B, A)\alpha - \frac{1}{2}\alpha^T K(A, A)\alpha \\
&\quad s.t. \;\; e_1^T \alpha = \nu_1, \;\; 0 \leq \alpha \leq \frac{c_1}{\ell_1} e_1
\end{aligned} \tag{18}$$

and

$$\begin{aligned}
&\max_{\beta} \; \frac{\nu_2}{\ell_1} e_1^T K(A, B)\beta - \frac{1}{2}\beta^T K(B, B)\beta \\
&\quad s.t. \;\; e_2^T \beta = \nu_2, \;\; 0 \leq \beta \leq \frac{c_2}{\ell_2} e_2.
\end{aligned} \tag{19}$$

After optimizing these QPPs, we get the $u_i \, (i = +, -)$ as follows:

$$u_+ = K(A, D^T)^T \alpha - \frac{\nu_1}{\ell_2} K(B, D^T)^T e_2 \tag{20}$$

and

$$u_- = -K(B, D^T)^T \beta + \frac{\nu_2}{\ell_1} K(A, D^T)^T e_1. \tag{21}$$

Value of the bias term (b_+) is given by:

$$O_+ = \{i : \alpha_i > 0\}, \;\; b_+ = -\frac{1}{|O_+|} \sum_{i \in O_+} K(x^T, D^T)u_+.$$

Similarly, value of bias term (b_-) is obtained by:

$$O_- = \{i : \beta_i > 0\}, \;\; b_+ = -\frac{1}{|O_-|} \sum_{i \in O_-} K(x^T, D^T)u_-.$$

A new data point $x \in \mathbb{R}^n$ is assigned to class $i \, (i = +1, -1)$ according to the Eq. (15).

2.5. Twin support vector machine with pinball loss (pin-TSVM)

Twin parametric-margin support vector machine (TPMSVM) [29] is an efficient classifier but it is noise sensitive. To further enhance the generalization performance, Xu et al. [49] introduced pinball loss to TPMSVM and proposed twin support vector machine with pinball loss (Pin-TSVM), especially for noise corrupted data. Similar to TPMSVM, nonlinear Pin-TSVM also derives a pair of kernel generated surfaces in input space given in Eq. (8). The nonlinear Pin-TSVM formulation can be expressed as follows:

$$\begin{aligned}
\min_{u_+, b_+, \xi_1} \ & \frac{1}{2}\|u_+\|^2 + \frac{\nu_1}{\ell_2} e_2^T\left(K(B, D^T)u_+ + e_2 b_+\right) + \frac{c_1}{\ell_1} e_1^T \xi_1 \\
s.t. \quad & K(A, D^T)u_+ + e_1 b_+ \geq -\xi_1, \\
& K(A, D^T)u_+ + e_1 b_+ \leq \frac{\xi_1}{\tau_1}
\end{aligned} \tag{22}$$

and

$$\begin{aligned}
\min_{u_-, b_-, \xi_2} \ & \frac{1}{2}\|u_-\|^2 - \frac{\nu_2}{\ell_1} e_1^T\left(K(A, D^T)u_- + e_1 b_-\right) + \frac{c_2}{\ell_2} e_2^T \xi_2 \\
s.t. \quad & -\left(K(B, D^T)u_- + e_2 b_-\right) \geq -\xi_2, \\
& -\left(K(B, D^T)u_- + e_2 b_-\right) \leq \frac{\xi_2}{\tau_2},
\end{aligned} \tag{23}$$

where $\nu_1, \nu_2 > 0$ are margin parameters and $\tau_1, \tau_2 \in [0, 1]$ are pinball loss function parameters. When τ_1 and τ_2 tends to zero then QPPs in Eqs. (22) and (23) are converted into the QPPs of TPMSVM. By introducing the Lagrange function and using the K.K.T. optimality conditions, we get the dual formulations of QPPs in Eqs. (22) and (23) as follows:

$$\begin{aligned}
\max_{\alpha, \beta} \ & \frac{\nu_1}{\ell_2} e_2^T K(B, A)^T(\alpha - \beta) - \frac{1}{2}(\alpha - \beta)^T K(A, A)^T(\alpha - \beta) \\
s.t. \ & e_1^T(\alpha - \beta) = \nu_1, \ \alpha + \frac{\beta}{\tau_1} = \frac{c_1}{\ell_1} e_1, \\
& \alpha \geq 0, \quad \beta \geq 0
\end{aligned} \tag{24}$$

and

$$\begin{aligned}
\max_{\gamma, \sigma} \ & \frac{\nu_2}{\ell_1} e_1^T K(A, B)^T(\gamma - \sigma) - \frac{1}{2}(\gamma - \sigma)^T K(B, B)^T(\gamma - \sigma) \\
s.t. \ & e_2^T(\gamma - \sigma) = \nu_2, \ \gamma + \frac{\sigma}{\tau_2} = \frac{c_2}{\ell_2} e_2, \\
& \gamma \geq 0, \quad \sigma \geq 0,
\end{aligned} \tag{25}$$

where α, β, γ and $\sigma \geq 0$ are Lagrange multipliers. One can obtain $u_i \, (i = +, -)$ similar to the Eqs. (20) and (21).

A new data point $x \in \mathbb{R}^n$ is assigned to class $i \, (i = +1, -1)$ according to the Eq. (15).

3. General twin support vector machine with pinball loss (pin-GTSVM)

Usual TSVM affiliates hinge loss function which is sensitive to noise and unstable for re-sampling. To overcome these drawbacks and elevate the performance of TSVM [15] similar to Pin-TSVM [49], we introduce pinball loss [14] in standard TSVM and propose a novel general twin support vector machine with pinball loss function termed as Pin-GTSVM, which is more general than Pin-TSVM. Pin-GTSVM deals with quantile distance [19], which makes it less sensitive to noise and more stable for re-sampling.

3.1. Linear pin-GTSVM

Consider a binary classification problem which classifies data points belonging to classes +1 and -1 and represented by matrices A and B, respectively. Let the data points belonging to class +1 and -1 be ℓ_1 and ℓ_2 respectively in the n-dimensional real space $\mathbb{R}^n$. To classify the positive (+1) and negative (-1) classes, we need to find two non-parallel hyperplanes defined as follows:

$$f^+(x) = w_+^T x + b_+ = 0 \text{ and } f^-(x) = w_-^T x + b_- = 0, \tag{26}$$

where $w_+, w_- \in \mathbb{R}^n$ and $b_+, b_- \in \mathbb{R}$. We introduce the pinball loss in the standard TSVM and obtain the following QPPs:

$$\min_{w_+, b_+} \frac{1}{2}\|Aw_+ + e_1 b_+\|^2 + c_1 \sum_{j=1}^{\ell_2} \mathcal{L}_{\tau_2}\left(x_j^-, y_j, f^+(x_j^-)\right) \tag{27}$$

and

$$\min_{w_-, b_-} \frac{1}{2}\|Bw_- + e_2 b_-\|^2 + c_2 \sum_{i=1}^{\ell_1} \mathcal{L}_{\tau_1}\left(x_i^+, y_i, f^-(x_i^+)\right), \tag{28}$$

where ℓ_1 and ℓ_2 denote the number of data points in positive and negative class respectively. Data point x_j^- corresponds to the negative class and x_i^+ corresponds to the positive class. $\mathcal{L}_{\tau_1}(\cdot)$ and $\mathcal{L}_{\tau_2}(\cdot)$ are pinball loss functions with parameters τ_1, $\tau_2 \in [0, 1]$ respectively. Substituting the pinball loss in QPPs in Eqs. (27) and (28), we obtain the following QPPs:

$$\begin{aligned} &\min_{w_+, b_+, \xi_1} \frac{1}{2}\|Aw_+ + e_1 b_+\|^2 + c_1 e_2^T \xi_1 \\ &s.t. \quad -(Bw_+ + e_2 b_+) + \xi_1 \geq e_2, \\ &\qquad -(Bw_+ + e_2 b_+) - \frac{\xi_1}{\tau_2} \leq e_2 \end{aligned} \tag{29}$$

and

$$\begin{aligned} &\min_{w_-, b_-, \xi_2} \frac{1}{2}\|Bw_- + e_2 b_-\|^2 + c_2 e_1^T \xi_2 \\ &s.t. \quad (Aw_- + e_1 b_-) + \xi_2 \geq e_1, \\ &\qquad (Aw_- + e_1 b_-) - \frac{\xi_2}{\tau_1} \leq e_1, \end{aligned} \tag{30}$$

where c_1, c_2 are positive penalty parameters and e_1, e_2 are vectors of ones of appropriate dimensions and ξ_1, ξ_2 are slack variables.

Remark: We observe that when τ_1 and τ_2 tend to zero then QPPs in Eqs. (29) and (30) are reduced to QPPs of linear TSVM. To obtain the solution of (29), we introduce the corresponding Lagrange function with Lagrange multipliers $\alpha \geq 0$ and $\beta \geq 0$

$$\begin{aligned} L(w_+, b_+, \xi_1, \alpha, \beta) = &\frac{1}{2}\|Aw_+ + e_1 b_+\|^2 + c_1 e_2^T \xi_1 - \alpha^T(-(Bw_+ + e_2 b_+) + \xi_1 - e_2) \\ &+ \beta^T\left(-(Bw_+ + e_1 b_+) - \frac{\xi_1}{\tau_2} - e_2\right). \end{aligned} \tag{31}$$

Using the Karush-Kuhn-Tucker (K.K.T.) [20] necessary and sufficient optimality conditions, we obtain:

$$A^T(Aw_+ + e_1 b_+) + B^T\alpha - B^T\beta = 0, \tag{32}$$

$$e_1^T(Aw_+ + e_1 b_+) + e_2^T\alpha - e_2^T\beta = 0, \tag{33}$$

$$c_1 e_2 - \alpha - \frac{\beta}{\tau_2} = 0. \tag{34}$$

By using the Eq. (34) and $\alpha \geq 0$, we obtain

$$-\tau_2 c_1 e_2 \leq (\alpha - \beta). \tag{35}$$

Combining Eqs. (32) and (33) leads to

$$\begin{bmatrix} A^T \\ e_1^T \end{bmatrix} [A \ e_1] \begin{bmatrix} w_+ \\ b_+ \end{bmatrix} + \begin{bmatrix} B^T \\ e_2^T \end{bmatrix} (\alpha - \beta) = 0. \tag{36}$$

Define $H = [A \ e_1]$, $G = [B \ e_2]$ and $z_+ = \begin{bmatrix} w_+ \\ b_+ \end{bmatrix}$. With these notations, the Eq. (36) can be rewritten as follows:

$$z_+ = -(H^T H)^{-1} G^T(\alpha - \beta). \tag{37}$$

Although $H^T H$ is always positive semi-definite, it is possible that it may not be well conditioned in some situations. So, we introduce regularization term [47] $\delta I (\delta > 0)$, to take care the possible ill-conditioning of $H^T H$. Here, I is an identity matrix of appropriate dimensions. Therefore, Eq. (37) becomes

$$z_+ = -(H^T H + \delta I)^{-1} G^T(\alpha - \beta). \tag{38}$$

Using Eq. (31) and the above K.K.T. conditions, we can obtain the dual of (29) as follows:

$$\begin{aligned} &\max_{(\alpha - \beta)} \ e_2^T(\alpha - \beta) - \frac{1}{2}(\alpha - \beta)^T G(H^T H)^{-1} G^T(\alpha - \beta) \\ &s.t. \quad -\tau_2 c_1 e_2 \leq (\alpha - \beta). \end{aligned} \tag{39}$$

Similarly, we can obtain the dual of QPP in Eq. (30) as follows:

$$\max_{(\gamma-\sigma)} \; e_1^T(\gamma-\sigma)-\frac{1}{2}(\gamma-\sigma)^T H(G^TG)^{-1}H^T(\gamma-\sigma)$$
$$s.t. \quad (\gamma-\sigma)\geq -\tau_1 c_2 e_1, \tag{40}$$

where $\gamma\geq 0$ and $\sigma\geq 0$ are Lagrange multipliers. Finally, optimal separating hyperplanes are given by:

$$\begin{bmatrix} w_+ \\ b_+ \end{bmatrix} = -(H^TH+\delta I)^{-1}G^T(\alpha-\beta)$$

and

$$\begin{bmatrix} w_- \\ b_- \end{bmatrix} = (G^TG+\delta I)^{-1}H^T(\gamma-\sigma). \tag{41}$$

A new data point $x\in\mathbb{R}^n$ is assigned to class $i\,(i=+1,-1)$ depending on which of the two hyperplanes in (26) is closer to x, i.e.,

$$class(i) = \text{sign}\left(\frac{w_+{}^Tx+b_+}{\|w_+\|}+\frac{w_-{}^Tx+b_-}{\|w_-\|}\right). \tag{42}$$

3.2. Nonlinear pin-GTSVM

In order to extend the linear Pin-GTSVM to the nonlinear Pin-GTSVM, we consider the following kernel generated surfaces:

$$K(x^T,D^T)u_+ + b_+ = 0 \text{ and } K(x^T,D^T)u_- + b_- = 0, \tag{43}$$

where $D=[A\,;\,B]$; $u_+,\; u_-\in\mathbb{R}^n$ and K is an arbitrary kernel function. Similar to linear Pin-GTSVM, we can construct the following optimization problems:

$$\min_{u_+,b_+,\xi_1} \frac{1}{2}\|K(A,\,D^T)u_+ + e_1b_+\|^2 + c_1e_2^T\xi_1$$
$$s.t. \quad -(K(B,\,D^T)u_+ + e_2b_+)+\xi_1\geq e_2,$$
$$-(K(B,\,D^T)u_+ + e_2b_+)-\frac{\xi_1}{\tau_2}\leq e_2 \tag{44}$$

and

$$\min_{u_-,b_-,\xi_2} \frac{1}{2}\|K(B,\,D^T)u_+ + e_2b_+\|^2 + c_2e_1^T\xi_2$$
$$s.t. \quad K(A,\,D^T)u_- + e_1b_- + \xi_2\geq e_1,$$
$$K(A,\,D^T)u_- + e_1b_- - \frac{\xi_2}{\tau_1}\leq e_1, \tag{45}$$

where $\xi_1,\,\xi_2$ are slack variables. $e_i\,(i=1,\,2)$ are standard unit vectors of appropriate dimensions. By introducing the Lagrange function and applying the K.K.T. optimality conditions, the dual of Eq. (44) is given by:

$$\max_{(\alpha-\beta)} \; e_2^T(\alpha-\beta)-\frac{1}{2}(\alpha-\beta)^T Q(P^TP)^{-1}Q^T(\alpha-\beta)$$
$$s.t. \quad -\tau_2c_1e_2\leq(\alpha-\beta). \tag{46}$$

Similarly, we can obtain the dual of Eq. (45) as follows:

$$\max_{(\gamma-\sigma)} \; e_1^T(\gamma-\sigma)-\frac{1}{2}(\gamma-\sigma)^T P(Q^TQ)^{-1}P^T(\gamma-\sigma)$$
$$s.t. \quad (\gamma-\sigma)\geq -\tau_1c_2e_1, \tag{47}$$

where $P=[K(A,\,D^T)\;e_1]$ and $Q=[K(B,\,D^T)\;e_2]$. $\alpha,\beta,\gamma,$ and $\sigma\geq 0$ are Lagrange multipliers. Finally, optimal separating hyperplanes are given by:

$$\begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = -(P^TP+\delta I)^{-1}Q^T(\alpha-\beta)$$

and

$$\begin{bmatrix} u_- \\ b_- \end{bmatrix} = (Q^TQ+\delta I)^{-1}P^T(\gamma-\sigma). \tag{48}$$

It is possible that P^TP and Q^TQ may not be well conditioned in some situations. So, we introduce a regularization term [47] $\delta I(\delta > 0)$, to take care of possible ill-conditioning of P^TP and Q^TQ. Here, I is an identity matrix of appropriate dimensions. A new point $x \in \mathbb{R}^n$ is assigned to class $i\,(i = +1, -1)$ depending on which of the two kernel generated surface in (43) is closer to x, i.e.

$$class(i) = \text{sign}\left(\frac{K(x^T, D^T)u_+ + b_+}{\|u_+\|} + \frac{K(x^T, D^T)u_- + b_-}{\|u_-\|}\right). \tag{49}$$

4. Algorithm analysis

Similar to TSVM, the proposed Pin-GTSVM generates two non-parallel hyperplanes by solving a pair of smaller sized QPPs. Twin parametric-margin support vector machine (TPMSVM) is an improved version of TSVM in which Peng [29] determined the parametric-margin hyperplanes similar to par-ν-SVM [11]. Xu et al. [49] introduced pinball loss function in TPMSVM and propose an algorithm Pin-TSVM. We introduce the pinball loss function in classical TSVM which is more general than Pin-TSVM [49]. Pin-GTSVM enjoys noise insensitivity and more stable for re-sampling.

4.1. Computational complexity

It is well known that the computational complexity of SVM is $\mathcal{O}(\ell^3)$ [7] where ℓ is the number of data points. Computational complexity of TSVM is $\mathcal{O}(2 \times (\frac{\ell}{2})^3)$ because TSVM divides the data into roughly two equal size matrices of order $(\frac{\ell}{2})$, when the dataset is balanced. The Computational complexity of Pin-GTSVM and Pin-TSVM are similar to TSVM. The computational cost of Pin-GTSVM is nearly the same as TSVM i.e., pinball loss function does not increase the computation time of Pin-GTSVM.

4.2. Noise insensitivity

The main advantage of pinball loss minimization is insensitivity with respect to the noise around the optimal separating hyperplanes. For ease of comprehension, we focus on the linear case. Consider the pinball loss function as follows:

$$\mathcal{L}_\tau(x, y, f(x)) = \begin{cases} -\tau(1 - yf(x)), & 1 - yf(x) < 0, \\ 1 - yf(x), & 1 - yf(x) \geq 0, \end{cases} \tag{50}$$

where $\tau \in [0, 1]$. Sub-gradient of the pinball loss $\mathcal{L}_\tau(x, y, f(x))$ is given by sign function $sgn_\tau(x, y, f(x))$:

$$sgn_\tau(x, y, f(x)) = \begin{cases} -\tau, & -yf(x) < 0, \\ [-\tau, 1], & -yf(x) = 0, \\ 1, & -yf(x) > 0. \end{cases} \tag{51}$$

Using the K.K.T. optimality condition for QPP in Eq. (27), we obtain:

$$A^TAw_+ + A^Te_1b_+ + c_1e_1\sum_{j=1}^{\ell_2} sgn_{\tau_2}(x_j^-, y_j,\ f^+(x_j^-))x_j^- = 0. \tag{52}$$

For given w_+ and b_+, the whole index set can be separated into three distinct subsets as follows:

$$E_1 = \{j : w_+^Tx_j + b_+ < 0\},$$

$$E_2 = \{j : w_+^Tx_j + b_+ > 0\},$$

$$E_3 = \{j : w_+^Tx_j + b_+ = 0\}.$$

Using the notation E_1,E_2 and E_3, Eq. (52) can be recast as follows:

$$A^TAw_+ + A^Te_1b_+ - c_1e_1\tau_2\sum_{j\in E_1} x_j^- + c_1e_1\sum_{j\in E_2} x_j^- + c_1e_1\sum_{j\in E_3}\zeta_j^+x_j^- = 0, \tag{53}$$

where $j = 1, 2, \cdots, \ell_2$ and $\zeta_j^+ \in [-\tau_2, 1]$. We perceive that τ_2 controls the numbers of points in E_1,E_2 and E_3. When $\tau_2 \to 0$, there are enough number of points in the set E_2. Hence the number of points in E_1 and E_2 will decrease. When $\tau_2 \to 1$, all of the three sets contain approximately equivalent number of data points. Hence the result is less sensitive [49].

From Table 2, we observe that as we increase the value of pinball loss parameter $\tau_i(i = 1, 2) \in [0, 1]$ it gives better results.

Proposition 1. *Let K_1 denotes the number of negative boundary errors. Then we can obtain a relation $K_1 \leq \frac{e_2^T\beta}{c_1\tau_2}$, which implies that $\frac{e_2^T\beta}{c_1\tau_2}$ is an upper bound for the negative class boundary errors.*

Proof. We know that for any negative class boundary error B_i, its penalty $\xi_{1i} > 0$, we have

$$\alpha_i(-(B_i w_+ + e_2 b_+) + \xi_{1i} - e_2) = 0,$$

consider $\alpha_i = 0$ implies $\beta_i = c_1\tau_2$

$$\Rightarrow \beta_i K_1 \le e_2^T \beta$$

$$\Rightarrow K_1 c_1 \tau_2 \le e_2^T \beta$$

$$\Rightarrow K_1 \le \frac{e_2^T \beta}{c_1 \tau_2}.$$

□

Proposition 2. *Let K_2 denotes the number of positive boundary errors. Then we can obtain a relation $K_2 \le \frac{e_1^T \alpha}{c_2 \tau_1}$, which implies that $\frac{e_1^T \alpha}{c_2 \tau_1}$ is an upper bound for the positive class boundary errors.*

Proof. The proof is similar to Proposition 1. □

5. Discussion on pin-GTSVM

In this section, we discuss the differences between Pin-SVM, TSVM, Pin-TSVM and the proposed Pin-GTSVM.

5.1. Pin-GTSVM vs. pin-SVM

Pin-GTSVM and Pin-SVM both uses the pinball loss function instead of the hinge loss function. Similar to the Pin-SVM, the proposed Pin-GTSVM also enjoys noise insensitivity around the optimal separating hyperplanes. In addition, Pin-GTSVM is more stable for re-sampling.

- As for the computation complexity of Pin-SVM and Pin-GTSVM, Pin-SVM solves a QPP with ℓ constraints, where ℓ is the total number data points, which implies that time complexity of Pin-SVM is approximately $\mathcal{O}(\ell^3)$. While the proposed Pin-GTSVM obtains two non-parallel hyperplanes by solving two QPPs of size $\frac{\ell}{2}$, which makes the learning speed of Pin-GTSVM approximately four times faster than Pin-SVM.
- In Pin-SVM, pinball loss of negative (-1) and positive (+1) class is determined by the same parameter $\tau \in [0, 1]$. On contrary, distinct values for τ_1 and τ_2 can be set for different classes in the proposed Pin-GTSVM. Therefore, Pin-GTSVM is more flexible for imbalanced datasets, at the cost of extra parameters tuning.

5.2. Pin-GTSVM vs. TSVM

Both Pin-GTSVM and TSVM solve a pair of QPPs, which are used to obtain a pair of non-parallel optimal hyperplanes. However, Pin-GTSVM and TSVM employ different loss functions viz. pinball loss and hinge loss, respectively.

- From Eq. (29), we observe Pin-GTSVM requires
$$\begin{aligned} &-(Bw_+ + e_2 b_+) + \xi_1 \ge e_2, \\ &-(Bw_+ + e_2 b_+) - \frac{\xi_1}{\tau_2} \le e_2. \end{aligned} \tag{54}$$
When $\tau_2 \neq 0$, Eq. (54) can recast as follows:
$$-\tau_2(Bw_+ + e_2 b_+) - \xi_1 \le \tau_2 e_2, \tag{55}$$
when τ_2 tends to zero, (55) will degenerate into $\xi_1 \ge 0$.
- The formulation of TSVM with hinge loss can be written as follows:
$$\min_{w_+, b_+} \frac{1}{2}\|Aw_+ + e_1 b_+\|^2 + c_1 \sum_{j=1}^{\ell_2} \mathcal{L}_{hinge}(x_j^-, y_j, f^+(x_j^-)). \tag{56}$$
After substituting the hinge loss in (56), we obtain the following inequalities:
$$-(Bw_+ + e_2 b_+) + \xi_1 \ge e_2 \quad \text{and} \quad \xi_1 \ge 0. \tag{57}$$
- After comparing Eqs. (54) and (57), we conclude that pinball loss gives an additional penalty to the correctly classified data points.
- Although Pin-GTSVM is developed from the TSVM. TSVM can be regarded as a special case of Pin-GTSVM.

Table 1
Imbalance ratio (IR) of datasets.

Dataset	No. of instances in majority class	No. of instances in minority class	Imbalance ratio(IR)
Fertility	88	12	**7.33**
Banknote	762	610	1.24
WDBC	357	212	1.68
Splice	1648	1527	1.07
Monk1	278	278	1
CMC	629	333	**1.89**
Crossplane150	75	75	1
Heart-stat	150	120	1.25
Heart-C	201	96	**2.09**
Ionosphere	225	126	1.78
Wpbc	148	46	**3.21**
Haberman	225	81	**2.78**
Cleveland	160	137	1.167
German	700	300	**2.33**
Breast-Cancer	444	239	**1.85**
Brwisconsin	444	239	**1.85**
Crossplane500	250	250	1

5.3. Pin-GTSVM vs. pin-TSVM

Pin-GTSVM and Pin-TSVM optimize two smaller sized QPPs in relation to the SVM counterpart, which are used to generate a pair of non-parallel hyperplanes to classify the data. Pin-GTSVM and Pin-TSVM employs the same pinball loss function, which makes them noise insensitive. However, there are some differences between these two algorithms.

- Pin-GTSVM finds two non-parallel hyperplanes such that each hyperplane is proximal to one of the two classes and is at least one unit apart from other class, while Pin-TSVM obtains a pair of parametric-margin hyperplanes such that each hyperplane obtains the positive or negative parametric margin.
- The construction of QPPs of the Pin-GTSVM and Pin-TSVM is totally different, including constraints and objective function. In the case of Pin-GTSVM, the same class data points are used in the objective function to minimize the distance of first class data points from the hyperplane and second class data points are used in the constraints of QPPs. In contrast, Pin-TSVM uses second class data points in the objective function and same class data points in the constraints.
- In Pin-GTSVM, the number of constraints are equal to the number of data points in the second class. Whereas, in the Pin-TSVM, the number of constraints are equal to the number of data points in the same class.

6. Numerical experiments

In classification problems, imbalanced datasets lead to erroneous classification. In such type of classification problems, there is a significant irregularity between the probability distribution of datasets in various classes, such class of datasets are known as imbalanced datasets [12]. Most of the SVM algorithms on class imbalance have shown inferior performance, which is due to the biased probabilities distribution of the datasets in different classes, given by the imbalance ratio (IR) [10]. IR defined as the ratio of the number of data points in the majority class to the number of samples in the minority class Fig. 2.

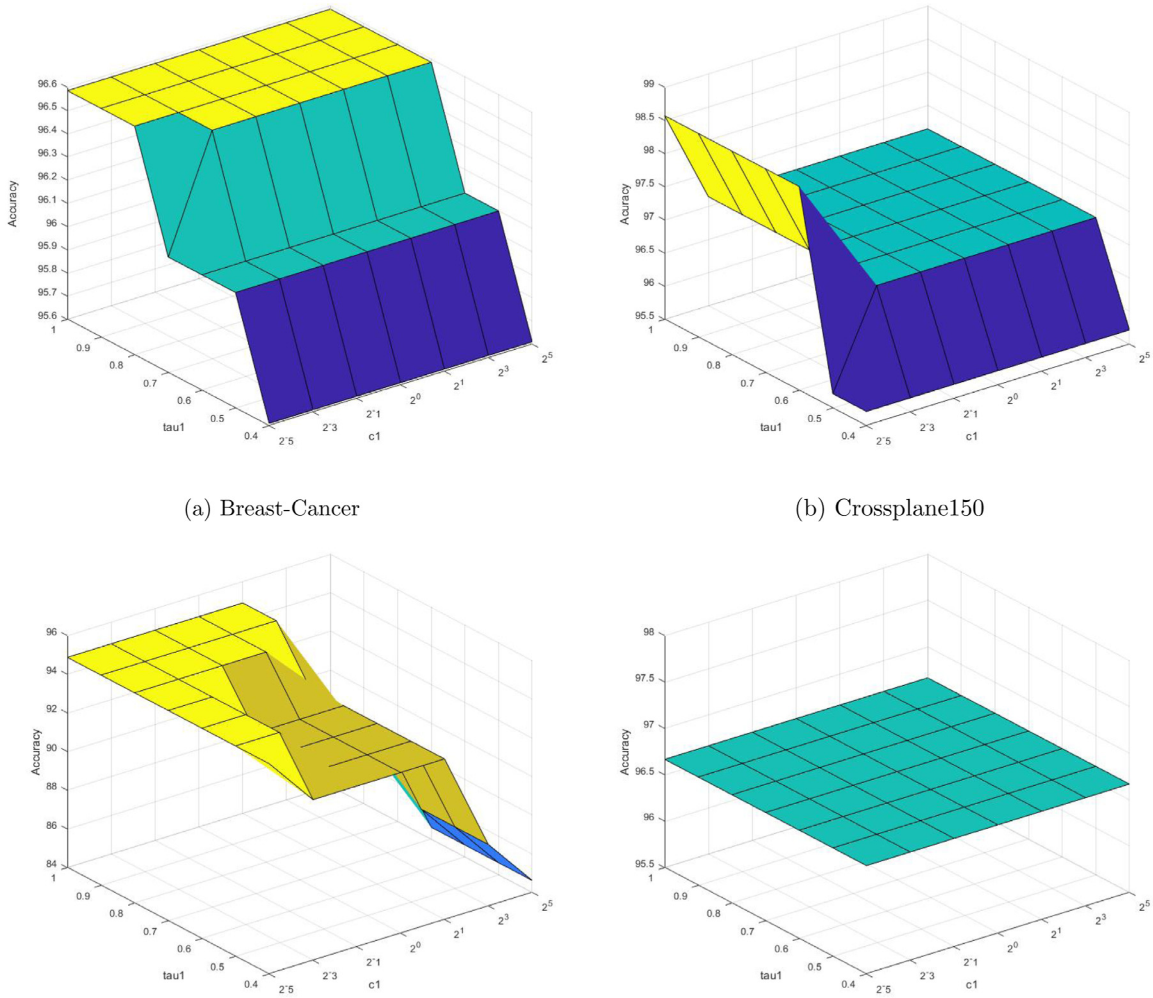
$$IR = \frac{\text{Number of data points in the majority class}}{\text{Number of data points in the minority class}}. \tag{58}$$

We exhibit the performance of four algorithms: TSVM [15], TPMSVM [29], Pin-TSVM [49] and the proposed Pin-GTSVM. In the proposed Pin-GTSVM, we perform experiments for different values of $\tau_i\,(i = 1,\, 2) = 0.5,\, 0.8,\, 1$ and investigate the variation in accuracy. We conduct experiments on 17 benchmark datasets including 8 imbalanced datasets (imbalance ratio greater than or equal to 1.85) taken from UCI machine learning repository [25] and KEEL repository [1]. The class imbalance ratios are shown in Table 1. The datasets are Fertility, Banknote, WDBC, Splice, Monk1, CMC, Crossplane150, Heart-stat, Heart-C, Ionosphere, Wpbc, Haberman, Cleveland, German, Breast-Cancer, Brwisconsin and Crossplane500 which are given in Table 2. In Table 2, the total number of samples, attributes and number of classes are denoted by "$\cdot \times \cdot \times \cdot''$ below the dataset name. For example, Fertility dataset contains 100 samples and each sample consists of the nine features classified in two classes as denoted by "$100 \times 9 \times 2''$. We test the performance of the proposed algorithm on classification accuracy and running time aspects. In our experiments, we normalize the datasets before training and testing. We incorporate the feature noise [14] in datasets with zero-mean Gaussian noise. For each attribute, standard deviation is denoted by r. The value of r is fixed $r = 0$ (i.e, noise free), 0.05 and 0.1. The testing and training sets are corrupted by a similar noise. In our experiments, we used 10-fold cross-validation to compare the performance of the proposed Pin-GTSVM with other baseline methods. In 10-fold cross-validation, the dataset is randomly partitioned into ten subsets while nine of them are used for training, one

Table 2
Performance comparison of binary-class algorithms using Gaussian kernel.

Dataset	TSVM [15] Accuracy ± sd Time(s)	TPMSVM [29] Accuracy ± sd Time(s)	Pin-TSVM [49] Accuracy ± sd Time(s)	Pin-GTSVM ($\tau = 0.5$) Accuracy ± sd Time(s)	Pin-GTSVM ($\tau = 0.8$) Accuracy ± sd Time(s)	Pin-GTSVM ($\tau = 1$) Accuracy ± sd Time(s)
Fertility	**96.66 ± 10.54**	**96.66 ± 10.54**	**96.66 ± 10.54**	**96.66 ± 10.54**	**96.66 ± 10.54**	**96.66 ± 10.54**
(100 × 9 × 2)	0.026	0.006	0.027	0.025	0.025	0.025
$r = 0.05$	**96.66 ± 10.54**	**96.66 ± 10.54**	**96.66 ± 10.54**	**96.66 ± 10.54**	**96.66 ± 10.54**	**96.66 ± 10.54**
	0.025	0.006	0.028	0.025	0.027	0.025
$r = 0.1$	**96.66 ± 10.54**	**96.66 ± 10.54**	**96.66 ± 10.54**	**96.66 ± 10.54**	**96.66 ± 10.54**	**96.66 ± 10.54**
	0.025	0.007	0.027	0.025	0.027	0.027
Banknote	36.446 ± 7.30	**42.44 ± 3.04**	41.26 ± 5.88	36.92 ± 7.44	36.92 ± 7.44	36.92 ± 7.44
(1372 × 5 × 2)	0.027	0.02	0.026	0.025	0.027	0.025
$r = 0.05$	38.13 ± 7.58	42.23 ± 4.17	42.23 ± 4.90	**42.26 ± 8.81**	**42.26 ± 8.81**	**42.26 ± 8.81**
	0.027	0.02	0.026	0.025	0.027	0.026
$r = 0.1$	41.53 ± 8.47	42.33 ± 4.03	**42.47 ± 4.86**	42.02 ± 8.98	42.02 ± 8.98	42.02 ± 8.98
	0.026	0.02	0.027	0.026	0.027	0.027
WDBC	95.71 ± 9.64	84.52 ± 24.38	88.2 ± 24.38	90.1 ± 15.81	89.2 ± 16.91	**98.28 ± 16.91**
(569 × 30 × 2)	0.007	0.009	0.007	0.007	0.006	0.009
$r = 0.05$	97.14 ± 6.02	84.52 ± 24.38	92.75 ± 24.38	97.14 ± 6.02	97.14 ± 6.02	**98.28 ± 16.91**
	0.007	0.009	0.007	0.006	0.006	0.006
$r = 0.1$	95.71 ± 9.64	91.3 ± 24.38	91.2 ± 24.38	**97.14 ± 6.02**	**97.14 ± 6.02**	**97.14 ± 6.02**
	0.007	0.007	0.007	0.0067	0.009	0.006
Splice	91.85 ± 1.440	87.83 ± 2.47	88.89 ± 2.52	92.08 ± 1.48	92.08 ± 1.51	**92.15 ± 1.46**
(767 × 60 × 2)	1.442	0.48	0.523	1.507	1.57	1.48
$r = 0.05$	91.81 ± 1.33	84.85 ± 2.52	87.73 ± 2.52	92.04 ± 1.58	92.04 ± 1.53	**92.15 ± 1.54**
	0.42	0.48	0.522	1.507	1.54	1.45
$r = 0.1$	91.81 ± 1.40	88.71 ± 2.52	87.43 ± 2.52	**91.92 ± 1.3**	**91.92 ± 1.42**	**91.92 ± 1.42**
	1.49	0.49	0.524	1.46	1.66	1.62
Monk1	82.79 ± 26.18	79.16 ± 20.34	74.10 ± 20.54	93.625 ± 11.18	93 ± 12.73	**94.25 ± 9.72**
(556 × 8 × 2)	0.009	0.011	0.012	0.007	0.007	0.007
$r = 0.05$	**82.16 ± 23.88**	75.92 ± 21.54	75.74 ± 22.58	81.5 ± 25.37	81.5 ± 25.37	81.5 ± 25.37
	0.009	0.011	0.012	0.009	0.008	0.009
$r = 0.1$	**78.91 ± 23.26**	72.14 ± 22.97	78.86 ± 22.58	77.66 ± 21.36	77 ± 21.15	77 ± 21.15
	0.008	0.012	0.012	0.008	0.009	0.008
CMC	74.24 ± 7.07	73.83 ± 7.97	73.62 ± 7.53	**74.67 ± 7.48**	**74.67 ± 7.48**	**74.67 ± 7.48**
(1473 × 9 × 2)	0.046	0.030	0.028	0.009	0.040	0.040
$r = 0.05$	**74.67 ± 7.48**	**74.67 ± 7.48**	**74.67 ± 7.48**	**74.67 ± 7.48**	**74.67 ± 7.48**	**74.67 ± 7.48**
	0.045	0.031	0.029	0.009	0.044	0.046
$r = 0.1$	**74.67 ± 7.48**	**74.67 ± 7.48**	**74.67 ± 7.48**	**74.67 ± 7.48**	**74.67 ± 7.4**	**74.67 ± 7.4**
	0.045	0.033	0.028	0.044	0.045	0.045
Crossplane150	94.28 ± 7.37	92.85 ± 26.12	90 ± 21.29	**95.71 ± 13.55**	**95.71 ± 13.55**	**95.71 ± 13.55**
(150 × 2 × 2)	0.007	0.014	0.017	0.005	0.008	0.007
$r = 0.05$	**98.57 ± 4.15**	94.28 ± 20.54	95.71 ± 28.57	95.71 ± 13.55	95.71 ± 13.55	95.71 ± 13.55
	0.006	0.015	0.017	0.005	0.008	0.005
$r = 0.1$	**97.14 ± 6.02**	95.71 ± 22.38	94.28 ± 26.76	94.28 ± 7.37	95.71 ± 6.9	95.71 ± 6.9
	0.006	0.014	0.017	0.005	0.006	0.0057
Heart-stat	**86.85 ± 11.76**	75.71 ± 14.20	83.71 ± 14.20	78.57 ± 12.14	78.57 ± 12.14	78.57 ± 12.14
(270 × 14 × 2)	0.007	0.011	0.0107	0.0078	0.008	0.005
$r = 0.05$	82.85 ± 11.26	85.71 ± 14.20	**86.66 ± 14.20**	80 ± 12.04	81.42 ± 9.64	78.57 ± 12.14
	0.008	0.011	0.0105	0.006	0.006	0.006
$r = 0.1$	81.42 ± 13.55	**86.66 ± 14.20**	85.72 ± 14.20	74.28 ± 14.75	75.71 ± 11.76	75.71 ± 11.76
	0.006	0.009	0.0102	0.006	0.007	0.006
Heart-C	75 ± 13.02	60.83 ± 18.86	60.833 ± 18.86	66.66 ± 13.60	70.67 ± 13.60	**76.67 ± 13.60**
(297 × 14 × 2)	0.008	0.011	0.012	0.007	0.007	0.017
$r = 0.05$	73.33 ± 15.11	60.833 ± 18.86	60.833 ± 18.86	70 ± 11.91	70 ± 11.91	**75.12 ± 11.91**
	0.008	0.012	0.012	0.007	0.007	0.007
$r = 0.1$	70 ± 14.27	60.833 ± 18.86	60.833 ± 18.86	70.33 ± 12.29	68.33 ± 11.29	**76.33 ± 12.29**
	0.007	0.011	0.013	0.007	0.007	0.011
Ionosphere	95.18 ± 5.08	88.54 ± 24.09	**96.72 ± 22.09**	94.27 ± 4.94	94.27 ± 4.94	94.27 ± 4.94
(351 × 34 × 2)	0.008	0.012	0.011	0.006	0.008	0.010
$r = 0.05$	**95.18 ± 5.08**	93.8 ± 24.09	93.8 ± 24.09	89.45 ± 11.86	88.54 ± 12.82	94.27 ± 12.95
	0.007	0.009	0.011	0.007	0.007	0.008
$r = 0.1$	96.18 ± 4.9	95.34 ± 23.97	**96.69 ± 24.09**	92.45 ± 3.99	90.54 ± 6.388	**96.69 ± 6.07**
	0.009	0.011	0.011	0.0104	0.010	0.007
Wpbc	80.33 ± 21.22	80.33 ± 21.22	80.33 ± 21.22	81.66 ± 13.64	**82 ± 14.67**	**82 ± 14.67**
(194 × 35 × 2)	0.007	0.008	0.007	0.006	0.007	0.008
$r = 0.05$	81 ± 15.69	80.33 ± 21.22	80.33 ± 21.22	**83.66 ± 13.64**	**83.66 ± 13.64**	**83.66 ± 13.64**

(continued on next page)



(a) Breast-Cancer (b) Crossplane150

(c) Crossplane500 (d) Fertility

Fig. 2. Effect of τ and $c_1 = c_2$ on accuracy for proposed Pin-GTSVM.

is used for testing. This process is repeated ten times and performance measure is taken as the average of ten iterations. All the algorithms are implemented in MATLAB R2010b on Windows 10 Education on a PC with system configuration Intel (R) Core (TM) i7-6700 CPU @ 3.40 GHZ with 8 GB of RAM. Gaussian kernel function $K(x, y) = \exp^{-(\|x-y\|^2/\mu^2)}$ is used, where μ is a parameter.

6.1. Parameter selection

It is clear that the performance of different algorithms depends on the choices of parameters. In our experiments, the optimal value of parameters are found by the grid search method [13]. For all algorithms, we have to choose three parameters penalty parameters c_1 and c_2 and Gaussian kernel parameter μ. We select from three values of τ_1, τ_2 as 0.5, 0.8 and 1. The optimal value for parameters c_1 and c_2 are selected from the set $\{2^j \mid j = -5, -3, \cdots, 3, 5\}$ and μ is selected over the range $\{2^j \mid j = -10, -9, \cdots, 9, 10\}$. To reduce the computation cost of parameter selection, we set $\tau_1 = \tau_2$.

6.2. Results comparison and discussion

We compare the proposed Pin-GTSVM with TSVM [15], TPMSVM [29] and Pin-TSVM [49]. The experimental results are presented in Table 2. From the perspective of prediction accuracy, we see that the proposed Pin-GTSVM outperforms on 11 datasets out of 17. Pin-GTSVM attains best prediction accuracy in 37, 18 and 16 cases out of 51, when $\tau_i = 1$, 0.8 and 0.5, respectively. We notice that the proposed Pin-GTSVM gives better results for noise corrupted datasets due to the use

Table 2 (*continued*)

Dataset	TSVM [15] Accuracy ± sd Time(s)	TPMSVM [29] Accuracy ± sd Time(s)	Pin-TSVM [49] Accuracy ± sd Time(s)	Pin-GTSVM ($\tau = 0.5$) Accuracy ± sd Time(s)	Pin-GTSVM ($\tau = 0.8$) Accuracy ± sd Time(s)	Pin-GTSVM ($\tau = 1$) Accuracy ± sd Time(s)
	0.007	0.009	0.007	0.006	0.007	0.0122
$r = 0.1$	82.33 ± 18.39	80.33 ± 21.22	80.33 ± 21.22	85.66 ± 14.49	85.66 ± 14.49	**88.66 ± 15.49**
	0.007	0.006	0.009	0.006	0.007	0.006
Haberman	75.45 ± 14.01	**76.55 ± 8.74**	75.36 ± 15.07	68.72 ± 21.10	68.72 ± 21.10	68.72 ± 21.10
$(306 \times 4 \times 2)$	0.009	0.016	0.016	0.007	0.009	0.008
$r = 0.05$	76.36 ± 15.94	77.32 ± 7.35	76.36 ± 15.73	75.45 ± 16.22	75.45 ± 16.22	**77.45 ± 16.22**
	0.007	0.016	0.016	0.0081	0.009	0.008
$r = 0.1$	70.74 ± 15.12	**78.37 ± 7.35**	77.16 ± 15.73	45.74 ± 15.12	40.78 ± 15.12	42.45 ± 14.23
	0.016	0.016	0.016	0.015	0.017	0.017
Cleveland	45.83 ± 16.78	40 ± 16.41	47.5 ± 17.14	75.833 ± 13.86	**76.66 ± 12.29**	**76.66 ± 12.29**
$(297 \times 14 \times 2)$	0.009	0.014	0.015	0.007	0.009	0.007
$r = 0.05$	67.5 ± 5.62	41.66 ± 16.66	47.5 ± 17.14	75.65	77.5 ± 11.14	**78.33 ± 11.91**
	0.008	0.010	0.017	0.008	0.009	0.009
$r = 0.1$	50.83 ± 7.90	41.37 ± 16.57	47.5 ± 17.14	78.33 ± 9.78	76.66 ± 10.97	**78.5 ± 13.05**
	0.008	0.011	0.017	0.009	0.009	0.0013
German	71 ± 7.37	69.4 ± 6.52	69.5 ± 7.97	**72.5 ± 8.57**	**72.5 ± 8.57**	**72.5 ± 8.57**
$(1000 \times 25 \times 2)$	0.011	0.0107	0.015	0.013	0.014	0.012
$r = 0.05$	70.05 ± 6.43	67.92 ± 5.92	69.5 ± 7.97	**70.5 ± 6.85**	**70.5 ± 6.81**	**70.5 ± 6.81**
	0.011	0.0108	0.016	0.0122	0.013	0.012
$r = 0.1$	68.5 ± 5.50	68.71 ± 5.92	**69.5 ± 7.97**	**69.5 ± 7.97**	**69.5 ± 7.97**	**69.5 ± 7.97**
	0.012	0.010	0.014	0.012	0.013	0.012
Breast-Cancer	**98.54 ± 2.33**	89.16 ± 10.50	93.79 ± 11.27	96.07 ± 3.92	96.07 ± 3.92	96.07 ± 3.92
$(683 \times 10 \times 2)$	0.014	0.017	0.021	0.017	0.012	0.015
$r = 0.05$	98.54 ± 2.33	87.95 ± 11.96	95.5 ± 11.2	98.54 ± 2.33	98.54 ± 2.33	**98.72 ± 2.49**
	0.013	0.016	0.022	0.015	0.013	0.014
$r = 0.1$	98.54 ± 2.33	89.14 ± 10.52	89.86 ± 11.27	97.57 ± 3.48	96.61 ± 3.23	**98.72 ± 3.36**
	0.013	0.017	0.022	0.012	0.015	0.015
Brwisconsin	99.02 ± 2.058	94.63 ± 10.50	94.63 ± 11.99	96.07 ± 2.81	96.07 ± 3.92	**99.07 ± 3.92**
$(683 \times 10 \times 2)$	0.013	0.019	0.022	0.014	0.019	0.022
$r = 0.05$	99.02 ± 2.05	94.63 ± 9.67	94.63 ± 11.27	97.07 ± 2.52	96.11 ± 2.04	**99.07 ± 3.14**
	0.015	0.021	0.022	0.014	0.018	0.014
$r = 0.1$	99.02 ± 2.05	93.78 ± 11.96	94.2 ± 11.2	95.64 ± 3.54	96.09 ± 3.85	**99.09 ± 3.22**
	0.014	0.020	0.023	0.012	0.015	0.020
Crossplane500	**98.66 ± 2.81**	96.23 ± 14.48	89.42 ± 17.90	**98.66 ± 2.81**	**98.66 ± 2.81**	**98.66 ± 2.81**
$(500 \times 3 \times 2)$	0.014	0.025	0.021	0.017	0.015	0.015
$r = 0.05$	96 ± 3.44	94.28 ± 12.58	95.71 ± 14.12	**97.33 ± 4.66**	**97.33 ± 4.66**	**97.33 ± 4.66**
	0.026	0.025	0.025	0.015	0.014	0.014
$r = 0.1$	97.33 ± 3.44	95.71 ± 10.44	94.28 ± 15.15	97.33 ± 3.44	97.33 ± 3.44	**98.32 ± 3.44**
	0.014	0.025	0.022	0.012	0.0152	0.014

of pinball loss function. Clearly, one can observe from Table 2 that the proposed Pin-GTSVM outperforms on 7 out of 8 imbalanced datasets. Pin-GTSVM has the best prediction accuracy in 21, 11, 10 cases of imbalanced datasets out of 24, when $\tau_i = 1$, 0.8 and 0.5, respectively. From the perspective of training time, we observed that the proposed Pin-GTSVM costs nearly the same computation time as TSVM. The proposed Pin-GTSVM has good generalization performance on noisy and imbalanced datasets as compared to existing classification algorithms. Experimental analysis authenticates that the proposed Pin-GTSVM is having the least rank for most of the datasets which justifies its robustness to different noise corrupted and imbalanced datasets.

6.3. Statistical analysis

In Table 2, we observe that the proposed Pin-GTSVM does not outperform on all datasets. To verify the statistical significance of the proposed Pin-GTSVM with different values of τ (0.5, 0.8, 1) in comparison with TSVM [15], TPMSVM [29] and Pin-TSVM [49], we use Friedman test [8,9]. The Friedman test with corresponding post hoc tests is considered to be appropriate for comparison of multiple classifiers over multiple datasets. Friedman test ranks the algorithm for each dataset separately, the best performing algorithm getting the smallest rank value and so on. The average ranks of all algorithms based on the accuracy with Gaussian kernel function are computed and listed in Table 3.

Under the null hypothesis, the Friedman statistics are distributed according to $\mathcal{X}_F^2$ with $(k-1)$ degree of freedom as follows [8]:

$$\mathcal{X}_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right], \tag{59}$$

Table 3
Average rank on the accuracy of six algorithms on datasets with noise ($r = 0$).

Dataset	TSVM [15]	TPMSVM [29]	Pin-TSVM [49]	Pin-GTSVM ($\tau = 0.5$)	Pin-GTSVM ($\tau = 0.8$)	Pin-GTSVM ($\tau = 1$)
Fertility	3.5	3.5	3.5	3.5	3.5	3.5
Banknote	6	1	2	4	4	4
WDBC	2	6	5	3	4	1
Splice	4	6	5	2.5	2.5	1
Monk1	4	5	6	2	3	1
CMC	4	5	6	2	2	2
Crossplane150	4	5	6	2	2	2
Heart-stat	1	6	2	4	4	4
Heart-C	2	5.5	5.5	4	3	1
Ionosphere	2	6	1	4	4	4
Wpbc	5	5	5	3	1.5	1.5
Haberman	2	1	3	5	5	5
Cleveland	5	6	4	3	1.5	1.5
German	4	6	5	2	2	2
Breast-Cancer	1	6	5	3	3	3
Brwisconsin	2	5.5	5.5	3.5	3.5	1
Crossplane500	2.5	5	6	2.5	2.5	2.5
Average Rank	3.17	4.91	4.44	3.11	3	2.35

Table 4
Average ranks on the accuracy of six algorithms on datasets with noise r.

Noise (r)	TSVM [15]	TPMSVM [29]	Pin-TSVM [49]	Pin-GTSVM ($\tau = 0.5$)	Pin-GTSVM ($\tau = 0.8$)	Pin-GTSVM ($\tau = 1$)
0.05	3.20	4.79	4.38	3.23	3.23	2.14
0.1	3.29	4.32	3.97	3.38	3.61	2.5

$$F_F = \frac{(N-1)\mathcal{X}_F^2}{N(k-1) - \mathcal{X}_F^2}, \tag{60}$$

$$\mathcal{X}_F^2 = \frac{12 \times 17}{6(6+1)}\left[3.17^2 + 4.91^2 + 4.44^2 + 3.11^2 + 3^2 + 2.35^2 - \frac{6 \times 7^2}{4}\right] = 22.17, \tag{61}$$

$$F_F = \frac{(17-1) \times 22.17}{17 \times (6-1) - 22.17} = 5.64, \tag{62}$$

where $R_j = \frac{1}{N}\sum_j r_i^j$ and r_i^j denotes the rank of jth algorithm on the ith dataset out of N datasets and F_F is F-distribution with a degree of freedom $(k-1)(N-1)$, where k is the number of algorithms and N number of datasets. The critical values of $F(5, 80)$ for $\alpha = 0.05$ is 2.33, so we reject the null-hypothesis. Based on the Nemenyi test, the critical difference is $CD = q_\alpha\sqrt{\frac{k(k+1)}{6N}} = 2.728\sqrt{\frac{(5\times6)}{(6\times17)}} = 1.4795$. One can see that the proposed Pin-GTSVM (with $\tau = 0.05, 0.8, 1$) is significantly better than the TPMSVM. Also, the proposed Pin-GTSVM (with $\tau = 1$) is significantly better than the Pin-TSVM. However, the Nemenyi test fails to detect the significant difference between the proposed Pin-GTSVM and TSVM, but the proposed Pin-GTSVM achieved a lower average rank as the same [50] can be seen from Table 3 and 4.

7. Conclusions and future directions

In this paper, a novel general twin support vector machine with pinball loss (Pin-GTSVM) algorithm is proposed for solving classification problems. Pinball loss function is widely used in regression problems since there is a strong a relation between quantile regression and pinball loss function. Here, we use pinball loss function in the standard TSVM instead of the hinge loss. Some properties including noise insensitivity are discussed from both theoretical and experimental viewpoints. In addition, we compare the proposed Pin-GTSVM with TSVM, TPMSVM, and Pin-TSVM. We also investigate the effect of value τ_i $(i = 1, 2)$ in the proposed Pin-GTSVM. Pin-GTSVM is more stable for noise corrupted data than the standard TSVM which sustains its applicability to real world classification problems. In future, we will investigate techniques for compressing and re-sampling so that the proposed Pin-GTSVM can be applied to large-scale noise corrupted classification problems. Further study on this topic will also include applications of Pin-GTSVM in real life classification problems with noise. Another interesting topic would be to design a fast algorithm for the proposed Pin-GTSVM and introduce pinball loss function in other variants of TSVM.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported by Science and Engineering Research Board (SERB), Government of India under Early Career Research Award Scheme, Grant No. ECR/2017/000053 and Council of Scientific & Industrial Research (CSIR), New Delhi, INDIA under Extra Mural Research (EMR) Scheme Grant No. 22(0751)/17/EMR-II. We gratefully acknowledge the Indian Institute of Technology Indore for providing facilities and support.

References

[1] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, J. Multiple-Valued Logic Soft Comput. 17 (2011).
[2] S. Balasundaram, M. Tanveer, On proximal bilateral-weighted fuzzy support vector machine classifiers, Int. J. Adv. Intell. Paradigms 4 (3–4) (2012) 199–210.
[3] S. Balasundaram, M. Tanveer, On Lagrangian twin support vector regression, Neural Comput. Appl. 22 (1) (2013) 257–267.
[4] D. Bollegala, Y. Matsuo, M. Ishizuka, A web search engine-based approach to measure semantic similarity between words, IEEE Trans. Knowl. Data Eng. 23, no. 7 (2011) 977–990.
[5] F. Chang, C.Y. Guo, X.R. Lin, C.J. Lu, Tree decomposition for large-scale SVM problems, J. Mach. Learn. Res. 11 (2010) 2935–2972.
[6] P.H. Chen, R.E. Fan, C.J. Lin, A study on SMO-type decomposition methods for support vector machines, IEEE Trans. Neural Networks 17, no. 4 (2006) 893–908.
[7] C. Cortes, V. Vapnik, Support vector networks, Mach. Learn. 20 (1995) 273–297.
[8] J. Demsar, Statistical comparisons of classifiers over multiple datasets, J. Mach. Learn. Res. 7 (2006) 1–30.
[9] S. Garcia, A. Fernandez, J. Luengo, F. Herrera, Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining experimental analysis of power, Inf. Sci. 180 (2010) 2044–2064.
[10] V. Garc??, J.S. S?anche, R.A. Mollineda, On the effectiveness of preprocessing methods when dealing with different levels of class imbalance, Knowl. Based Syst. 25, no. 1 (2012) 13–21.
[11] P.Y. Hao, New support vector algorithms with parametric insensitive/margin model, Neural Networks 23, no. 1 (2010) 60–73.
[12] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 no. 9 (2009) 1263–1284.
[13] C. Hsu, C. Lin, A comparison of methods for multiclass support vector machine, IEEE Trans. Neural Networks. 13 (2002) 415–425.
[14] X. Huang, L. Shi, J. Suykens, Support vector machine classifier with pinball loss, IEEE Trans. Pattern Anal. Mach. Intell. 36, no. 5 (2014) 984–997.
[15] Jayadeva, R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, IEEE Trans. Pattern Anal. Mach. Intell. 29, no. 5 (2007) 905–910.
[16] T. Joachims, C. Ndellec, C. Rouveriol, Text categorization with support vector machines: learning with many relevant features, in: European Conference on Machine Learning No.10, Chemnitz, Germany, 1998, pp. 137–142.
[17] R. Khemchandani, P. Saigal, S. Chandra, Angle-based twin support vector machine, Ann. Oper. Res. (2017) 1–31.
[18] R. Khemchandani, S. Sharma, Robust least squares twin support vector machine for human activity recognition, Appl. Soft. Comput. 47 (2016) 33–46.
[19] R. Koenker, Quantile Regression, Cambridge, U. K.: Cambridge University Press, 2005.
[20] H.W. Kuhn, A.W. Tucker, Nonlinear programming, in: Proceedings of 2nd Berkeley Symposium. Berkeley: University of California Press, 1951, pp. 481–492.
[21] Y.-J. Lee, O.L. Mangasarian, RSVM: reduced support vector machines, in: Proceedings of the 2001 SIAM International Conference on Data Mining, SIAM, 2001, pp. 1–17.
[22] Y.Q. Li, C.T. Guan, Joint feature re-extraction and classification using an iterative semi-supervised support vector machine algorithm, Mach. Learn. 71, no. 1 (2008) 33–53.
[23] D. Liu, D. Li, Y. Shi, Y. Tian, Large-scale linear nonparallel SVMs, Soft. Comput. 22 (6) (2018) 1945–1957.
[24] O.L. Mangasarian, E.W. Wild, Multisurface proximal support vector classification via generalized eigenvalues, IEEE Trans. Pattern Anal. Mach. Intell. 28, no. 1 (2006) 69–74.
[25] P.M. Murphy, D.W. Aha, UCI repository of machine learning databases, http://www.ics.uci.edu/~mlearn(1992).
[26] E. Osuna, R. Freund, F. Girosi, Training support vector machines: an application to face detection, in: Proceedings of IEEE Computer Vision and Pattern Recognition, San Juan, Puerto Rico, 1997, pp. 130–136.
[27] M. Pal, P. Mather, Support vector machines for classification in remote sensing, Int. J. Remote Sens. 26 (5) (2005) 1007–1011.
[28] X. Peng, TSVR: an efficient twin support vector machine for regression, Neural Network 23 (3) (2010) 365–372.
[29] X. Peng, A novel twin parametric-margin support vector machine for pattern recognition, Pattern Recognit. 44 nos. 10–11 (2011) 2678–2692.
[30] X. Peng, D. Chen, PTSVRS: regression models via projection twin support vector machine, Inf. Sci. 435 (2018) 1–14.
[31] B. Richhariya, A. Sharma, M. Tanveer, Improved universum twin support vector machine, in: 2018 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2018, pp. 2045–2052.
[32] B. Richhariya, M. Tanveer, EEG signal classification using universum support vector machine, Expert Syst. Appl. 106 (2018) 169–182.
[33] B. Richhariya, M. Tanveer, A robust fuzzy least squares twin support vector machine for class imbalance learning, Appl. Soft Comput. 71 (2018) 418–432.
[34] Y.-H. Shao, W.-J. Chen, N.-Y. Deng, Nonparallel hyperplane support vector machine for binary classification problems, Inf. Sci. 263 (2014) 22–35.
[35] Y.-H. Shao, C.-H. Zhang, X.-B. Wang, N.-Y. Deng, Improvements on twin support vector machines, IEEE Trans. Neural Networks 22 (6) (2011) 962–968.
[36] X. Shen, L. Niu, Z. Qi, Y. Tian, Support vector machine classifier with truncated pinball loss, Pattern Recognit. 68 (2017) 199–210.
[37] S.K. Shevade, S.S. Keerthi, C. Bhattacharyya, K.R.K. Murthy, Improvements to SMO algorithm for SVM regression, IEEE Trans. Neural Networks 11, no. 5 (2000) 1188–1193.
[38] A.J. Smola, B. Scholkopf, A tutorial on support vector regression, Stat. Comput. 14, no. 3 (2004) 199–222.
[39] I. Steinwart, A. Christmann, How SVMs can estimate quantiles and the median, in: Electronic Proceedings Neural Information Processing Systems, 2007, pp. 305–312.
[40] J.A. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (3) (1999) 293–300.
[41] L. Tang, Y. Tian, P.M. Pardalos, A novel perspective on multiclass classification: regular simplex support vector machine, Inf. Sci. 480 (2019) 324–338.
[42] M. Tanveer, Robust and sparse linear programming twin support vector machines, Cognit. Comput. 7 (1) (2015) 137–149.
[43] M. Tanveer, M.A. Khan, S.-S. Ho, Robust energy-based least squares twin support vector machines, Appl. Intell. 45, no. 1 (2016) 174–186.
[44] M. Tanveer, M. Mangal, I. Ahmad, Y.-H. Shao, One norm linear programming support vector regression, Neurocomputing 173 (2016) 1508–1518.
[45] M. Tanveer, K. Shubham, A regularization on Lagrangian twin support vector regression, Int. J. Mach. Learn. Cybern. 8 (3) (2017) 807–821.

[46] M. Tanveer, K. Shubham, Smooth twin support vector machines via unconstrained convex minimization, Filomat 31 (8) (2017) 2195–2210.
[47] A.N. Tikhonov, V.Y. Arsenin, Solution of Ill Posed Problems, John Wiley and Sons, 1997.
[48] Z. Wang, Y.-H. Shao, L. Bai, C.-N. Li, L.-M. Liu, N.-Y. Deng, Insensitive stochastic gradient twin support vector machines for large scale problems, Inf. Sci. 462 (2018) 114–131.
[49] Y. Xu, Z. Yang, X. Pan, A novel twin support vector machine with pinball loss, IEEE Trans. Neural Networks Learn. Syst. 28, no. 2 (2017) 359–370.
[50] Y. Xu, Z. Yang, Y. Zhang, X. Pan, L. Wang, A maximum margin and minimum volume hyper-spheres machine with pinball loss for imbalanced data classification, Knowl. Based Syst. 95 (2016) 75–85.