Machine Learning (2022) 111:3525–3548
https://doi.org/10.1007/s10994-021-06061-z

Large-scale pinball twin support vector machines

M. Tanveer[1] · A. Tiwari[2] · R. Choudhary[2] · M. A. Ganaie[1]

Received: 22 October 2020 / Revised: 9 August 2021 / Accepted: 27 August 2021 /
Published online: 4 October 2021

Abstract
Twin support vector machines (TWSVMs) have been shown to be effective classifiers for a range of pattern classification tasks. However, the TWSVM formulation suffers from a range of shortcomings: (i) TWSVM uses hinge loss function which renders it sensitive to dataset outliers (noise sensitivity). (ii) It requires a matrix inversion calculation in the Wolfe-dual formulation which is intractable for datasets with large numbers of features/ samples. (iii) TWSVM minimizes the empirical risk instead of the structural risk in its formulation with the consequent risk of overfitting. This paper proposes a novel large scale pinball twin support vector machines (LPTWSVM) to address these shortcomings. The proposed LPTWSVM model firstly utilizes the pinball loss function to achieve a high level of noise insensitivity, especially in relation to data with substantial feature noise. Secondly, and most significantly, the proposed LPTWSVM formulation eliminates the need to calculate inverse matrices in the dual problem (which apart from being very computationally demanding may not be possible due to matrix singularity). Further, LPTWSVM does not employ kernel-generated surfaces for the non-linear case, instead using the kernel trick directly; this ensures that the proposed LPTWSVM is a fully modular kernel approach in contrast to the original TWSVM. Lastly, structural risk is explicitly minimized in LPTWSVM with consequent improvement in classification accuracy (we explicitly analyze the properties of classification accuracy and noise insensitivity of the proposed LPTWSVM). Experiments on benchmark datasets show that the proposed LPTWSVM model may be effectively deployed on large datasets and that it exhibits similar or better performance on most datasets in comparison to relevant baseline methods.

Editors: João Gama, Alípio Jorge, Salvador García.

✉ M. Tanveer
mtanveer@iiti.ac.in

A. Tiwari
artiwari@iiti.ac.in

R. Choudhary
cse150001027@iiti.ac.in; rahulc.0212@gmail.com

M. A. Ganaie
phd1901141006@iiti.ac.in

[1] Department of Mathematics, Indian Institute of Technology Indore, Simrol, Indore 453552, India

[2] Department of Computer Science and Engineering, Indian Institute of Technology Indore, Simrol, Indore 453552, India

Keywords Support vector machines · Pinball loss function · Twin support vector machines

1 Introduction

Support vector machines (SVMs), introduced by Vapnik and co-workers (Cortes & Vapnik, (1995; Vapnik, 1999), are a class of highly effective machine learning models for pattern classification. SVMs are based on statistical learning theory (Trafalis & Ince, 2000; Vapnik, 1998, 2013; González-Castano et al., 2004; Fung & Mangasarian, 2005) and have been applied extensively in relation to binary classification problems. The traditional SVM model works by margin maximisation; deriving two unique parallel supporting hyperplanes such that the distance between the samples of two classes is maximized. The fact that relatively few training objects are required for this support gives SVMs their characteristic robustness. Further, casting the problem in dual form enables explicit kernelization, greatly extending the method's utility. The versatility of SVMs has enabled their widespread adoption in various fields such as financial time-series forecasting (Cao & Tay, 2003), computational biology (Borgwardt, 2011; Noble, 2004), face recognition (Déniz et al. 2003), cancer recognition (Valentini et al., 2004), and EEG signal classification (Richhariya & Tanveer, 2018). To address the issue of parameter tuning, an automated procedure (Chapelle et al., 2002) for selecting kernel parameters was introduced in Chapelle et al. (2002) as exhaustive search may become intractable. However, in the era of big data, with significantly increasing number of features, many traditional SVM models fail to perform satisfactorily on reference datasets (Van Gestel et al., 2004; Fernández-Delgado et al., 2014).

To address this, SVMs have recently been enhanced by the development of several non-parallel hyperplane-based classifiers; e.g. the generalized eigen-value proximal SVM (GEPSVM) proposed by Mangasarian and Wild (2006), and the twin support vector machines, proposed by Jayadeva et al. (2007). TWSVM, in particular, improves SVM classification accuracy through the calculation of two non-parallel hyperplanes, each of which aims to be as close as possible to its corresponding class while being as far away as possible from the other. TWSVM, hence, modifies its hyperplanes so as to better accommodate the different distributions of the two classes, i.e., by changing the parameters of the two sub-problems TWSVM solves. These sub-problems (TWSVM solves two smaller quadratic programming problems (QPPs) unlike the original SVM which solves a single QPP) also endow TWSVM with additional computational efficiency, being approximately 4 times faster than the original support vector machine. As a result of these advantages, TWSVMs have been widely studied (Chen et al., 2011; Kumar & Gopal, 2008, 2009; Peng, 2010; Qi et al., 2013; Tanveer et al., 2016a; Richhariya & Tanveer, 2020).

Despite the merits of TWSVM, it still suffers from noise sensitivity and instability under resampling as a consequence of hinge loss function. To address these limitations within a standard SVM context, Huang et al. (2014) introduced a novel pinball loss function ($L_\tau(u)$) within the SVM formulation. The pinball loss function uses a quantile metric to measure margin distances in order to reduce noise sensitivity and increase stability under re-sampling. However, introduction of the pinball loss function leads to a loss of sparsity, a key component of SVM classification performance. To handle this drawback, a ϵ -insensitive zone pinball loss ($L^\epsilon_\tau(u)$) SVM (Huang et al., 2014) is introduced to reduce the effect of noise while obtaining sparse solutions.

Several modifications to the TWSVM methodology have been proposed in order to reduce time complexity and improve overall performance (Kumar & Gopal, 2008; Gao et al., 2011;

Kumar et al., 2010; Sharma et al., 2021; Tanveer, 2015; Tanveer et al., 2016b; Tian & Ping, 2014; Xu & Wang, 2014; Yan et al., 2019). Attempts have also been made to handle multi-class classification problems (Cheong et al., 2004; Madzarov et al., 2009; Shao et al., 2013). These TWSVM based formulations, however, are based on the hinge-loss function which, as indicated, suffers from noise sensitivity and re-sampling instability on large datasets. Also, the pin-SVM (Huang et al., 2014) solves a single large QPP further reducing its applicability to large-scale datasets. To resolve these issues (noise, resampling instability and high computational complexity), a pinball loss TWSVM was proposed in Tanveer et al. (2019a). However, introduction of pinball loss function within the TWSVM leads to a loss of sparsity; in order to gain the benefits of noise insensitivity, resampling stability, *and* sparsity, the sparse pinball twin support vector machine (SPTWSVM) was introduced in Tanveer et al. (2019b), Wang et al. (2020) and Singla et al. (2020) in which a $\epsilon-$insensitive zone pinball loss function is introduced in the standard TWSVM. For more details, interested readers can refer to comprehensive review on TWSVM (Tanveer et al., 2021).

However, all of the discussed formulations remain inapplicable to large scale problems due to the requirement of computationally expensive or intractable matrix inversion. Hence, extension of the above problems to large scale problems is still an open challenge.

Motivated by the existing twin bounded SVMs (TBSVM) (Shao et al., 2011) and Pin-general twin SVMs (Pin-GTSVM) (Tanveer et al., 2019a), we here propose a novel large scale pinball twin SVM (LPTWSVM). In particular, the merits of the proposed LPTWSVM formulation are as follows:

- LPTWSVM, by virtue of changes directly introduced to the primal form of TBSVM, can be feasibly applied to real-world large-scale datasets. This is done by the elimination of the matrix inverse calculation in the dual problem of our model, which can be intractable for large scale datasets or even impossible for singular matrices.
- LPTWSVM explicitly minimizes the structural risk in its formulation in accordance with statistical learning theory, and consequently matches or improves generalization/classification accuracy compared to baseline methods.
- LPTWSVM obviates the requirement for the computation of the matrix inverse that exists in the TBSVM and Pin-GTSVM formulations, and the consequent risk of intractability.
- LPTWSVM does not implicate kernel-generated surfaces in its methodology in contrast to the majority of twin support vector machine formulations, and is thus free to incorporate the kernel trick directly into its dual problem.
- LPTWSVM achieves outlier-insensitivity by virtue of the introduction of pinball loss into the modified TBSVM problem.

A broad paper outline is given as follows: Sect. 2 gives the background of the previous work, Sect. 3 outlines the proposed model. Section 4 covers the theoretical properties of our model in detail. Experimental results are given in Sect. 5 and conclusions are provided in Sect. 6.

2 Background

The formulations of Pin-SVM, TBSVM and Pin-GTSVM are given briefly in this section. For further details, readers are referred to Jayadeva et al. (2007), Huang et al. (2014), Shao et al. (2011) and Tanveer et al. (2019a). We also have a notation subsection before these formulations.

Consider a binary dataset $\mathbf{z} = \{x_i, y_i\}_{i=1}^{m}$ where $x_i \in \mathbb{R}^n$ and $y_i \in \{-1, 1\}$. Let the number of samples in class $+1$ and -1 be m_1 and m_2, respectively. Let A represent the positive class $(+1)$ samples and B represent the negative class (-1) samples.

2.1 Notations used

Table 1 list the various abbreviations and symbols used throughout the paper here.

2.2 Pinball support vector machines

Huang et al. (2014) first formulated the noise insensitive pinball loss function based SVM classifier. The optimization problem of the pinball SVM is:

$$\min_{\mathbf{w},\mathbf{b}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{m} L_\tau\big(1 - y_i\big(\mathbf{w}^T\phi(x_i) + b\big)\big), \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^n$ and $\phi(x)$ is the weight vector and Hilbert space transformation of x, $b \in \mathbb{R}$ is bias and L_τ is the pinball loss function. The optimal separating hyperplane is given as $\mathcal{H} : \mathbf{w}^T\phi(x) + b = 0$; a test data sample $x \in \mathbb{R}^n$ is hence assigned to the respective class of $+1$ or -1 based on the sign of $\mathbf{w}^T x + b$ i.e. if the sign is positive it is classified as positive class sample otherwise it is negative class sample. In pinball loss SVM, correctly classified samples are additionally penalized via the loss function with the intention of reducing noise sensitivity:

Table 1 Notation table

Symbol/abbreviation	Interpretation
$\mathbf{w}$	Weight vector
b	Bias
ξ	Slack variable
A	Class of positive samples $(+1)$
B	Class of negative samples (-1)
L_{hinge}	Hinge loss function
L_τ	Pinball loss function
τ	Pinball loss function penalisation parameter
QPP	Quadratic programming problem
SVM	Support vector machine
Pin-SVM	Pinball support vector machines
TWSVM	Twin support vector machines
TBSVM	Twin bounded support vector Machines
Pin-GTSVM	General twin support vector machine with pinball loss function
LPTWSVM	Large scale pinball twin support vector machines
SPTWSVM	Sparse pinball twin support vector machines

$$L_\tau(q) = \begin{cases} q, & q \geq 0, \\ -\tau q, & q < 0. \end{cases} \tag{2}$$

Here, $\tau \in [0, 1]$ is a penalisation parameter that controls the magnitude of negative loss values. Equation (2) is a generalized ℓ_1 loss with both vectors close to the decision boundary as well as vectors further away from it contributing to the weight vector $\mathbf{w}$. In particular, quantile distance is maximized by the Pin-SVM model.

Incorporating the Pinball loss function in Eq. (1), the QPP is given as follows:

$$\begin{aligned} \min_{\mathbf{w},\mathbf{b},\boldsymbol{\xi}} \quad & \tfrac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \\ \text{subject to} \quad & y_i\big(\mathbf{w}^T\phi(x_i) + b\big) \geq 1 - \xi_i, \\ & y_i\big(\mathbf{w}^T\phi(x_i) + b\big) \leq 1 + \tfrac{\xi_i}{\tau}, \quad i = 1, 2, \ldots, m, \end{aligned} \tag{3}$$

where $\mathbf{w} \in \mathbb{R}^n$ is the weight vector, $b \in \mathbb{R}$ is bias, $\xi = (\xi_1, \xi_2, \ldots, \xi_m)^T$ is a slack variable and $C > 0$ is a penalty parameter.

Both Pin-SVM and SVM solve a QPP which is used to find an optimal hyperplane. However, SVM used hinge loss function and Pin-SVM used pinball loss function. For Eq. (3), the constraints are:

$$y_i\big(\mathbf{w}^T\phi\big(x_i\big) + b\big) \geq 1 - \xi_i, \tag{4}$$

$$y_i\big(\mathbf{w}^T\phi\big(x_i\big) + b\big) \leq 1 + \frac{\xi_i}{\tau}. \tag{5}$$

When $\tau \neq 0$, Eq. (5) can be recast as:

$$\tau y_i\big(\mathbf{w}^T\phi(x_i) + b\big) \leq \tau + \xi_i, \tag{6}$$

when τ tends to zero, (6) degenerates into $\xi_i \geq 0$. The optimization problem of nonlinear SVM is expressed as follows:

$$\min_{\mathbf{w},\mathbf{b}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{m} L_{hinge}\big(1 - y_i\big(\mathbf{w}^T\phi\big(x_i\big) + b\big)\big), \tag{7}$$

where $\mathbf{w} \in \mathbb{R}^n$ is the weight vector, $b \in \mathbb{R}$ is bias and L_{hinge} is the hinge loss function. We can get the following inequalities after substituting the hinge loss in (7):

$$y_i\big(\mathbf{w}^T\phi\big(x_i\big) + b\big) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0. \tag{8}$$

After comparing Eqs. (5) and (8), one can conclude that the pinball loss gives an additional penalty to the correctly classified data points. Although Pin-SVM is developed from the original SVM, SVM can be regarded as a special case of Pin-SVM.

2.3 Twin bounded support vector machines (TBSVM)

In an attempt to improve the TWSVM model, Shao et al. (2011) proposed the twin bounded support vector machines (TBSVM) in which they directly introduced the structural risk minimization principle into the TWSVM problem and eliminate the need to ensure a well-conditioned matrix (so as to calculate its inverse) involved in the dual formulation of TWSVM. The

structural risk minimization principle is implemented via the introduction of a regularization term in the objective function of original TWSVM. An added benefit of introducing this regularization term is that it eliminates the need to derive the dual of the problem without additional assumptions unlike TWSVM. Thus, TBSVM stands as a significant improvement over TWSVM. The formulation of TBSVM is given as:

$$\begin{array}{ll} \min\limits_{\mathbf{w^{(1)}},\mathbf{b^{(1)}},\boldsymbol{\xi}} & \frac{1}{2}c_3\left(||\mathbf{w}^{(1)}||^2 + b^{(1)2}\right) + \frac{1}{2}||A\mathbf{w}^{(1)} + e_1 b^{(1)}||^2 + c_1 e_2^T \xi \\ \text{subject to} & -\left(B\mathbf{w}^{(1)} + e_2 b^{(1)}\right) + \xi \geq e_2, \quad \xi \geq 0 \end{array} \tag{9}$$

and

$$\begin{array}{ll} \min\limits_{\mathbf{w^{(2)}},\mathbf{b^{(2)}},\boldsymbol{\xi}} & \frac{1}{2}c_4(||\mathbf{w}^{(2)}||^2 + b^{(2)2}) + \frac{1}{2}||B\mathbf{w}^{(2)} + e_2 b^{(2)}||^2 + c_2 e_1^T \xi \\ \text{subject to} & \left(A\mathbf{w}^{(2)} + e_1 b^{(2)}\right) + \xi \geq e_1, \quad \xi \geq 0. \end{array} \tag{10}$$

Here, $\mathbf{w}^{(1)}, \mathbf{w}^{(2)} \in \mathbb{R}^n$ are the weight vectors with $b^{(1)}, b^{(2)} \in \mathbb{R}$ the corresponding biases for QPPs (9) and (10) respectively; e_1 and e_2 are vectors of ones of appropriate dimensions and ξ is a slack variable. The term $\frac{1}{2}c_3(||\mathbf{w}^{(1)}||^2 + b^{(1)2})$ in (9) introduces the structural risk minimization principle since the term corresponds to the distance between the proximal hyperplane, $\mathbf{w}^{(1)T}x + b^{(1)} = 0$, and the bounding hyperplane, $\mathbf{w}^{(1)T}x + b^{(1)} = -1$. A similar analysis holds for (10).

We only consider the dual problem of (9) since a similar approach can be followed for the others. The Lagrangian of (9) is given as,

$$\begin{aligned} \mathcal{L} = & \frac{1}{2}c_3\left(||\mathbf{w}^{(1)}||^2 + b^{(1)2}\right) + \frac{1}{2}\left(A\mathbf{w}^{(1)} + e_1 b^{(1)}\right)^T \left(A\mathbf{w}^{(1)} + e_1 b^{(1)}\right) \\ & + c_1 e_2^T \xi - \alpha^T\left(-\left(B\mathbf{w}^{(1)} + e_2 b^{(1)}\right) + \xi - e_2\right) - \beta^T \xi, \end{aligned} \tag{11}$$

where $\alpha \in \mathbb{R}^{m_2}$, $\beta \in \mathbb{R}^{m_2}$ are Lagrangian multipliers corresponding to the different constraints. After applying the necessary and sufficient K.K.T. conditions, the QPP (9) in dual form is given as:

$$\begin{array}{ll} \min\limits_{\delta} & \frac{1}{2}\delta^T G\left(H^T H + c_3 I\right)^{-1} G^T \delta - e_2^T \delta \\ \text{subject to} & 0 \leq \delta \leq c_1 e_2, \end{array} \tag{12}$$

where $H = \begin{bmatrix} A & e_1 \end{bmatrix}$, $G = \begin{bmatrix} B & e_2 \end{bmatrix}$, $\delta \in \mathbb{R}^{m_2}$, I is the identity matrix of size $m_1 \times m_1$. As is evident, $(H^T H + c_3 I)^{-1}$ is naturally nonsingular and, hence, invertible without making any extra assumptions unlike TWSVM's dual problems. However, despite the differences in formulation, the decision function of TBSVM is similar to that of TWSVM.

2.4 General twin support vector machine with pinball loss function (Pin-GTSVM)

To remove the limitations of hinge loss based support vector machines, in particular, sensitivity to noise and resampling, Tanveer et al. (2019a) formulated general twin SVMs using the pinball loss function, demonstrating that their Pin-GTSVM model successfully reduces sensitivity to feature noise and exhibits stability under re-sampling while retaining the same computational complexity as that of the TWSVM with hinge loss. The formulation of the linear Pin-GTSVM is given as follows:

$$\begin{aligned} \min_{\mathbf{w}^{(1)},b^{(1)},\xi_1} \quad & \frac{1}{2}\|A\mathbf{w}^{(1)} + e_1 b^{(1)}\|^2 + c_1 e_2^T \xi_1 \\ \text{subject to} \quad & -\left(B\mathbf{w}^{(1)} + e_2 b^{(1)}\right) + \xi_1 \geq e_2, \\ & -\left(B\mathbf{w}^{(1)} + e_2 b^{(1)}\right) - \frac{\xi_1}{\tau_2} \leq e_2 \end{aligned} \tag{13}$$

and

$$\begin{aligned} \min_{\mathbf{w}^{(2)},b^{(2)},\xi_2} \quad & \frac{1}{2}\|B\mathbf{w}^{(2)} + e_2 b^{(2)}\|^2 + c_2 e_1^T \xi_2 \\ \text{subject to} \quad & \left(A\mathbf{w}^{(2)} + e_1 b^{(2)}\right) + \xi_2 \geq e_1 \\ & \left(A\mathbf{w}^{(2)} + e_1 b^{(2)}\right) - \frac{\xi_2}{\tau_1} \leq e_1, \end{aligned} \tag{14}$$

where $\mathbf{w}^{(1)}, \mathbf{w}^{(2)} \in \mathbb{R}^n$ are the weight vectors with $b^{(1)}, b^{(2)} \in \mathbb{R}$ the corresponding biases, c_1, c_2 are positive penalty parameters, e_1, e_2 are vectors of ones of appropriate dimensions and ξ_1, ξ_2 are the slack variables.

Similarly, the formulation of the non-linear Pin-GTSVM is given as follows:

$$\begin{aligned} \min_{u^{(1)},b^{(1)},\xi_1} \quad & \tfrac{1}{2}\|K(A, D^T)u^{(1)} + e_1 b^{(1)}\|^2 + c_1 e_2^T \xi_1 \\ \text{subject to} \quad & -\left(K(B, D^T)u^{(1)} + e_2 b^{(1)}\right) + \xi_1 \geq e_2 \\ & -\left(K\left(B, D^T\right)u^{(1)} + e_2 b^{(1)}\right) - \frac{\xi_1}{\tau_2} \leq e_2 \end{aligned} \tag{15}$$

and

$$\begin{aligned} \min_{u^{(2)},b^{(2)},\xi_2} \quad & \tfrac{1}{2}\|K\left(B, D^T\right)u^{(2)} + e_2 b^{(2)}\|^2 + c_2 e_1^T \xi_2 \\ \text{subject to} \quad & \left(K(A, D^T)u^{(2)} + e_1 b^{(2)}\right) + \xi_2 \geq e_1 \\ & \left(K(A, D^T)u^{(2)} + e_1 b^{(2)}\right) - \frac{\xi_2}{\tau_1} \leq e_1, \end{aligned} \tag{16}$$

where c_1, c_2 are positive penalty parameters, e_1, e_2 are the vector of ones with appropriate dimensions and ξ_1, ξ_2 are the slack variables, $D = [A;B]$; $u^{(1)}, u^{(2)} \in \mathbb{R}^n$ and $K(\cdot)$ is a kernel function.

To solve the QPPs (13–16), we derive their dual form. We consider QPP (15) for this purpose. The dual of QPP (15) can be written as:

$$\begin{aligned} \max_{(\alpha-\beta)} \quad & e_2^T(\alpha - \beta) - \tfrac{1}{2}(\alpha - \beta)^T Q(P^T P)^{-1} Q^T(\alpha - \beta) \\ \text{subject to} \quad & -\tau_2 c_1 e_2 \leq (\alpha - \beta). \end{aligned} \tag{17}$$

Likewise the dual of QPP (16) is given as :

$$\begin{aligned} \max_{(\gamma-\sigma)} \quad & e_1^T(\gamma - \sigma) - \tfrac{1}{2}(\gamma - \sigma)^T P(Q^T Q)^{-1} P^T(\gamma - \sigma) \\ \text{subject to} \quad & (\gamma - \sigma) \geq -\tau_1 c_2 e_1, \end{aligned} \tag{18}$$

where $P = \left[K(A, D^T)\ e_1\right]$ and $Q = \left[K(B, D^T)\ e_2\right]$. α, β, γ and σ are Lagrangian multipliers.

Once we solve the QPP's (17) and (18), the optimal hyperplanes are given as:

$$\begin{bmatrix} u^{(1)} \\ b^{(1)} \end{bmatrix} = -\left(P^T P + \delta I\right)^{-1} Q^T(\alpha - \beta) \tag{19}$$

and

$$\begin{bmatrix} u^{(2)} \\ b^{(2)} \end{bmatrix} = \left(Q^T Q + \delta I\right)^{-1} P^T (\gamma - \sigma). \tag{20}$$

3 Proposed large scale pinball twin support vector machines (LPTWSVM)

In order to render our model suitable for large scale datasets, we aim to re-formulate the problem so as to embody the merits of both TBSVM and Pin-GTSVM while eliminating the requirement for calculating large matrix inverses. This will involve the introduction of a regularization term (as in TBSVM) along with the addition of an equality constraint. Furthermore, since pinball loss is already present in the primal problem of Pin-GTSVM, LPTWSVM is insensitive to noise and is thus more stable with respect to resampling. The LPTWSVM also allows for the kernel trick to be incorporated directly into the dual problem without having to accommodate kernel-generated surfaces. Lastly, LPTWSVM, unlike Pin-GTSVM, directly embodies a structural risk minimization principle, potentially allowing LPTWSVM to obtain better classification accuracy. LPTWSVM thus stands as a significant improvement over both TBSVM and Pin-GTSVM.

3.1 Linear LPTWSVM

We reformulate the optimization problem of TBSVM (9) by incorporating the pinball loss function. The reformulated optimization problem is given as:

$$\begin{aligned} \min_{\mathbf{w}^{(1)},\mathbf{b}^{(1)},\boldsymbol{\eta}_1,\boldsymbol{\xi}} \quad & \tfrac{1}{2}c_3\left(||\mathbf{w}^{(1)}||^2 + b^{(1)2}\right) + \tfrac{1}{2}\eta_1^T\eta_1 + c_1 e_2^T\xi \\ \text{subject to} \quad & A\mathbf{w}^{(1)} + e_1 b^{(1)} = \eta_1, \\ & -\left(B\mathbf{w}^{(1)} + e_2 b^{(1)}\right) + \xi \geq e_2, \\ & -\left(B\mathbf{w}^{(1)} + e_2 b^{(1)}\right) \leq e_2 + \tfrac{\xi}{\tau} \end{aligned} \tag{21}$$

and

$$\begin{aligned} \min_{\mathbf{w}^{(2)},\mathbf{b}^{(2)},\boldsymbol{\eta}_2,\boldsymbol{\xi}} \quad & \tfrac{1}{2}c_4\left(||\mathbf{w}^{(2)}||^2 + b^{(2)2}\right) + \tfrac{1}{2}\eta_2^T\eta_2 + c_2 e_1^T\xi \\ \text{subject to} \quad & B\mathbf{w}^{(2)} + e_2 b^{(2)} = \eta_2, \\ & \left(A\mathbf{w}^{(2)} + e_1 b^{(2)}\right) + \xi \geq e_1, \\ & \left(A\mathbf{w}^{(2)} + e_1 b^{(2)}\right) \leq e_1 + \tfrac{\xi}{\tau}. \end{aligned} \tag{22}$$

Here $\mathbf{w}^{(1)}, \mathbf{w}^{(2)} \in \mathbb{R}^n$ are the weight vectors with $b^{(1)}, b^{(2)} \in \mathbb{R}$ the corresponding biases, c_1, c_2, c_3, $c_4 > 0$, $\eta_1 \in \mathbb{R}^{m_1}$, $\eta_2 \in \mathbb{R}^{m_2}$, ξ is a slack vector and e_1 and e_2 are vectors of ones with m_1 and m_2 elements respectively. The third term in both problems is the error minimization term that arises according to whether or not the samples satisfy the constraints. Here, pinball loss (2) gives penalty to both correctly as well as incorrectly classified samples.

Compared to Pin-GTSVM problem, the proposed LPTWSVM model [QPPs (21) and (22)] introduced the regularization terms $\frac{1}{2}c_3(||\mathbf{w}^{(1)}||^2 + b^{(1)2})$ and $\frac{1}{2}c_4(||\mathbf{w}^{(2)}||^2 + b^{(2)2})$ and added an extra equality constraint in both primal problems. The addition of the regularization terms also introduces structural risk minimization since they correspond to the distance between the proximal hyperplane, $\mathbf{w}^{(1)T}x + b^{(1)} = 0$, and the bounding hyperplane, $\mathbf{w}^{(1)T}x + b^{(1)} = -1$

(both planes correspond to the first problem). The objective functions of both problems (21) and (22) minimize the distance of the data sample of one class from its corresponding hyperplane while remaining as far as possible from the hyperplane belonging to the samples of the other class. The modified Lagrangian in the proposed LPTWSVM is such that the previous large matrix inverse calculation is bypassed; it is immediately evident that eliminating these calculations will lead to significant efficiency improvements with respect to large scale datasets.

To solve problems (21) and (22), we thus consider their dual formulations. The dual of (21) and its Lagrangian function can be written as (a similar approach can be followed for others):

$$\begin{aligned}\mathcal{L} = {} & \frac{1}{2}c_3\left(||\mathbf{w}^{(1)}||^2 + b^{(1)2}\right) + \frac{1}{2}\eta_1^T\eta_1 + c_1 e_2^T\xi \\ & - \alpha^T\left(-(B\mathbf{w}^{(1)} + e_2 b^{(1)}) + \xi - e_2\right) \\ & - \gamma^T\left(\left(B\mathbf{w}^{(1)} + e_2 b^{(1)}\right) + e_2 + \frac{\xi}{\tau}\right) \\ & + \mu^T\left(A\mathbf{w}^{(1)} + e_1 b^{(1)} - \eta_1\right),\end{aligned} \tag{23}$$

here $\alpha \in \mathbb{R}^{m_2}$, $\alpha \geq 0$, $\gamma \in \mathbb{R}^{m_2}$, $\gamma \geq 0$ and $\mu \in \mathbb{R}^{m_1}$, $\mu \geq 0$ represent the Lagrangian multipliers. Applying the K.K.T conditions on (23), we have:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}^{(1)}} = c_3\mathbf{w}^{(1)} + B^T\alpha - B^T\gamma + A^T\mu = 0, \tag{24}$$

$$\frac{\partial \mathcal{L}}{\partial b^{(1)}} = c_3 b^{(1)} + e_2^T\alpha - e_2^T\gamma + e_1^T\mu = 0, \tag{25}$$

$$\frac{\partial \mathcal{L}}{\partial \xi} = c_1 e_2 - \alpha - \frac{\gamma}{\tau} = 0, \tag{26}$$

$$\frac{\partial \mathcal{L}}{\partial \eta_1} = \eta_1 - \mu = 0, \tag{27}$$

$$\alpha^T\left(-\left(B\mathbf{w}^{(1)} + e_2 b^{(1)}\right) + \xi - e_2\right) = 0, \tag{28}$$

$$\gamma^T\left(\left(B\mathbf{w}^{(1)} + e_2 b^{(1)}\right) + e_2 + \frac{\xi}{\tau}\right) = 0, \tag{29}$$

$$\mu^T\left(A\mathbf{w}^{(1)} + e_1 b^{(1)} - \eta_1\right) = 0. \tag{30}$$

Using the K.K.T. conditions (24)–(27) and (28)–(30) and substituting $\alpha - \gamma = \lambda$, the Lagrangian of (21) is given as:

$$\begin{aligned} \max_{\lambda,\mu} \quad & -\tfrac{1}{2}\begin{bmatrix}\mu^T & \lambda^T\end{bmatrix}\tilde{Q}\begin{bmatrix}\mu\\ \lambda\end{bmatrix} + c_3\lambda^T e_2 \\ \text{subject to} \quad & c_1 e_2 - \alpha - \tfrac{\gamma}{\tau} = 0, \\ & \alpha \geq 0, \quad \gamma \geq 0, \\ \text{where} \quad & \tilde{Q} = \begin{bmatrix} AA^T + c_3 I & AB^T \\ BA^T & BB^T \end{bmatrix} + E. \end{aligned} \tag{31}$$

Here, E is a matrix of all ones of size $(m_1 + m_2) \times (m_1 + m_2)$. The first constraint can also be written as $\lambda + \gamma(1 + \frac{1}{\tau}) = c_1 e_2$, using $\alpha = \gamma + \lambda$. Since $\gamma \geq 0$, (31) can be equivalently written as:

$$\begin{aligned} \min_{\mu,\lambda} \quad & \tfrac{1}{2}\begin{bmatrix}\mu^T & \lambda^T\end{bmatrix}\tilde{Q}\begin{bmatrix}\mu\\ \lambda\end{bmatrix} - c_3\lambda^T e_2 \\ \text{subject to} \quad & -\tau c_1 e_2 \leq \lambda \leq c_1 e_2, \\ \text{where} \quad & \tilde{Q} = \begin{bmatrix} AA^T + c_3 I & AB^T \\ BA^T & BB^T \end{bmatrix} + E. \end{aligned} \tag{32}$$

Likewise, the dual formulation of (22) is given as:

$$\begin{aligned} \min_{\theta,\phi} \quad & \tfrac{1}{2}\begin{bmatrix}\theta^T & \phi^T\end{bmatrix}\tilde{Q}\begin{bmatrix}\theta\\ \phi\end{bmatrix} - c_4\phi^T e_1 \\ \text{subject to} \quad & -\tau c_2 e_1 \leq \phi \leq c_2 e_1, \\ \text{where} \quad & \tilde{Q} = \begin{bmatrix} BB^T + c_4 I & -BA^T \\ -AB^T & AA^T \end{bmatrix} + E. \end{aligned} \tag{33}$$

Once solutions of (32) and (33) are derived so as to obtain the vectors $\begin{bmatrix}\mu & \lambda & \alpha\end{bmatrix}$ and $\begin{bmatrix}\theta & \phi & \omega\end{bmatrix}$, the hyperplanes corresponding to each class can be written as:

$$x^T \mathbf{w}^{(i)} + b^{(i)} = 0 \quad \text{for} \quad i = 1, 2. \tag{34}$$

The decision function for assigning the test data sample $x \in \mathbb{R}^n$ to a particular class is then similar to decision function used by Pin-GTSVM problem.

3.2 Non-linear LPTWSVM

In contrast to the non-linear Pin-GTSVM case, we do not need to consider kernel generated surfaces for LPTWSVM and can thus directly introduce arbitrary kernel functions in the linear case of LPTWSVM. Hence, we introduce an explicit kernel function $K(x, y) = \phi(x)^T\phi(y)$ into the linear case, which enacts the Hilbert space transformation $\mathbf{x} = \phi(x)$, $\mathbf{x} \in \mathbb{H}$. In a similar fashion to (21) and (22), we now consider the following primal problems in the Hilbert space $\mathbb{H}$:

$$\begin{aligned} \min_{\mathbf{w}^{(1)},\mathbf{b}^{(1)},\boldsymbol{\eta}_1,\boldsymbol{\xi}} \quad & \tfrac{1}{2}c_3\left(||\mathbf{w}^{(1)}||^2 + b^{(1)2}\right) + \tfrac{1}{2}\eta_1^T\eta_1 + c_1 e_2^T\xi \\ \text{subject to} \quad & \phi(A)\mathbf{w}^{(1)} + e_1 b^{(1)} = \eta_1, \\ & -\left(\phi(B)\mathbf{w}^{(1)} + e_2 b^{(1)}\right) + \xi \geq e_2, \\ & -\left(\phi(B)\mathbf{w}^{(1)} + e_2 b^{(1)}\right) \leq e_2 + \tfrac{\xi}{\tau} \end{aligned} \tag{35}$$

and

$$\begin{aligned} \min_{\mathbf{w}^{(2)},\mathbf{b}^{(2)},\boldsymbol{\eta}_2,\boldsymbol{\xi}} \quad & \tfrac{1}{2}c_4\left(||\mathbf{w}^{(2)}||^2 + b^{(2)2}\right) + \tfrac{1}{2}\eta_2^T\eta_2 + c_2 e_1^T\xi \\ \text{subject to} \quad & \phi(B)\mathbf{w}^{(2)} + e_2 b^{(2)} = \eta_2, \\ & \left(\phi(A)\mathbf{w}^{(2)} + e_1 b^{(2)}\right) + \xi \geq e_1, \\ & \left(\phi(A)\mathbf{w}^{(2)} + e_1 b^{(2)}\right) \leq e_1 + \tfrac{\xi}{\tau}. \end{aligned} \tag{36}$$

Here, all constants and notations have similar meaning as in the linear case. We derive the dual problem of (35) and (36):

$$\begin{aligned} \min_{\mu,\lambda} \quad & \tfrac{1}{2}\begin{bmatrix}\mu^T & \lambda^T\end{bmatrix}\tilde{Q}\begin{bmatrix}\mu \\ \lambda\end{bmatrix} - c_3\lambda^T e_2 \\ \text{subject to} \quad & -\tau c_1 e_2 \leq \lambda \leq c_1 e_2, \\ \text{where} \quad & \tilde{Q} = \begin{bmatrix}K(A^T,A^T) + c_3 I & K(A^T,B^T) \\ K(B^T,A^T) & K(B^T,B^T)\end{bmatrix} + E, \end{aligned} \tag{37}$$

and

$$\begin{aligned} \min_{\theta,\phi} \quad & \tfrac{1}{2}\begin{bmatrix}\theta^T & \phi^T\end{bmatrix}\tilde{Q}\begin{bmatrix}\theta \\ \phi\end{bmatrix} - c_4\phi^T e_1 \\ \text{subject to} \quad & -\tau c_2 e_1 \leq \phi \leq c_2 e_1, \\ \text{where} \quad & \tilde{Q} = \begin{bmatrix}K(B^T,B^T) + c_4 I & -K(B^T,A^T) \\ -K(A^T,B^T) & K(A^T,A^T)\end{bmatrix} + E. \end{aligned} \tag{38}$$

All variables, constants and notations are again similar to those from the linear case. Once we solve the QPPs (37) and (38), a new test-data sample $x \in \mathbb{R}^n$ is assigned to a given class based on its distance from the corresponding hyperplanes in a manner similar to the linear case.

4 Theoretical properties

We will now examine the properties of the proposed large-scale pinball twin support vector machine (LPTWSVM) in more detail.

4.1 Noise insensitivity

For simplicity, we will discuss the noise sensistivity with respect to the linear LPTWSVM problem (21). However, a similar analysis is also applicable to both the non-linear case of the first LPTWSVM problem and also the second LPTWSVM problem.

The generalized sign function, $\text{sgn}_\tau(x)$, of the Pinball loss function is given as:

$$\text{sgn}_\tau(x) = \begin{cases} 1, & x > 0, \\ [-\tau, 1], & x = 0, \\ -\tau, & x < 0. \end{cases} \tag{39}$$

Henceforth, for typographic simplicity, let $\mathbf{w}$ and b represent $\mathbf{w}^{(1)}$ and $b(1)$ respectively, such that problem (21) can be equivalently written:

$$\min_{\mathbf{w},\mathbf{b},\boldsymbol{\eta}_1,\boldsymbol{\xi}} \ \frac{1}{2}c_3\big(||\mathbf{w}||^2+b^2\big)+\frac{1}{2}\big(A\mathbf{w}+e_1b\big)^T\big(A\mathbf{w}+e_1b\big) + c_1e_2^TL_\tau\big(e_2+\big(B\mathbf{w}+e_2b\big)\big), \tag{40}$$

where $L_\tau(e_2+(B\mathbf{w}+e_2b))$ is the Pinball loss function. Differentiating (40) with respect to $\mathbf{w}$, we have:

$$\mathbf{0}=c_3\mathbf{w}+A^T\big(A\mathbf{w}+e_1b\big)+c_1\sum_{i=1}^{m_2}\mathrm{sgn}_\tau\big(1+\big(\mathbf{w}^Tx_i^-+b\big)\big)x_i^-, \tag{41}$$

where $\mathbf{0}$ is a zero vector of appropriate dimensions and $x_i^-\in B$.

The index set for B is partitioned in three parts as follows:

$$\begin{aligned} S_0^{\mathbf{w},\mathbf{b}}&=\big\{i:1+\big(\mathbf{w}^Tx_i^-+b\big)>0\big\},\\ S_1^{\mathbf{w},\mathbf{b}}&=\big\{i:1+\big(\mathbf{w}^Tx_i^-+b\big)=0\big\},\\ S_2^{\mathbf{w},\mathbf{b}}&=\big\{i:1+\big(\mathbf{w}^Tx_i^-+b\big)<0\big\}, \end{aligned} \tag{42}$$

where $i=1,\ldots,m_2$. With the above notations and the existence of $\theta_i\in[-\tau,1]$ Eq. (41) can be rewritten as:

$$\frac{c_3}{c_1}\mathbf{w}+\frac{1}{c_1}A^T\big(A\mathbf{w}+e_1b\big)+\sum_{i\in S_0^{\mathbf{w},\mathbf{b}}}x_i^-+\sum_{i\in S_1^{\mathbf{w},\mathbf{b}}}\theta_ix_i^--\tau\sum_{i\in S_2^{\mathbf{w},\mathbf{b}}}x_i^-=0. \tag{43}$$

The above condition shows that when $\mathbf{w},b,c_1$, and c_3 are fixed, τ controls the number of samples of each set $S_0^{\mathbf{w},\mathbf{b}}$, $S_1^{\mathbf{w},\mathbf{b}}$ and $S_2^{\mathbf{w},\mathbf{b}}$. For small values of τ, the number of samples in $S_2^{\mathbf{w},\mathbf{b}}$ is high with fewer number of samples in other sets, and hence the result is an intrinsic sensitivity to feature noise. On the other hand, for larger values of τ, set-allocation is more evenly distributed, with data samples allocated to all of the three sets and hence the result is less sensitivity to feature noise.

Proposition 1 *If the QPPs* (32) *or* (37) *have a solution then the following inequalities must hold*:

$$\frac{-\big(c_3b+e_1^T\mu\big)}{c_1m_2}\le 1 \quad \text{and} \quad \frac{p_0}{m_2}\le 1-\frac{1+\frac{(c_3b+e_1^T\mu)}{c_1m_2}}{1+\tau}, \tag{44}$$

where p_0 is the number of samples in $S_0^{\mathbf{w},\mathbf{b}}$.

Proof Let $x_{i_0}^-\in S_0^{\mathbf{w},\mathbf{b}}$, $(1\le i_0\le m_2)$ be an arbitrary sample. From the KKT condition (29), $\gamma_{i_0}=0$. From the KKT condition (26), we then obtain $\alpha_{i_0}=c_1$ and, subsequently, $\lambda_{i_0}=\alpha_{i_0}-\gamma_{i_0}=c_1$. Also, from the KKT condition (25), we have

$$\begin{aligned} &\sum_{i\in S_0^{\mathbf{w},\mathbf{b}}}\lambda_i+\sum_{i\notin S_0^{\mathbf{w},\mathbf{b}}}\lambda_i=-\big(c_3b+e_1^T\mu\big)\\ \Longrightarrow\ &p_0c_1+\sum_{i\notin S_0^{\mathbf{w},\mathbf{b}}}\lambda_i=-\big(c_3b+e_1^T\mu\big). \end{aligned}$$

Now, since $\alpha_i\ge 0$ and $\gamma_i\ge 0$, we have $-\tau c_1\le\lambda_i\le c_1$. Therefore,

$$\begin{aligned} &\frac{-\left(c_3 b + e_1^T \mu\right)}{c_1} - \left(m_2 - p_0\right) \leq p_0 \\ &\text{and} \quad p_0 \leq \frac{-\left(c_3 b + e_1^T \mu\right)}{c_1} + \tau\left(m_2 - p_0\right), \end{aligned} \tag{45}$$

which gives us $\frac{-(c_3 b + e_1^T \mu)}{c_1 m_2} \leq 1$ and $p_0(1+\tau) \leq \frac{-(c_3 b + e_1^T \mu) + \tau c_1 m_2}{c_1}$. The second condition gives us

$$\begin{aligned} \frac{p_0}{m_2} \leq \frac{\frac{-(c_3 b + e_1^T \mu)}{m_2} + \tau c_1}{c_1(1+\tau)} &= 1 - \frac{c_1 + \frac{(c_3 b + e_1^T \mu)}{m_2}}{c_1(1+\tau)} \\ &= 1 - \frac{1 + \frac{(c_3 b + e_1^T \mu)}{c_1 m_2}}{(1+\tau)}, \end{aligned}$$

hence proving our proposition. □

One can see that an upper bound on the number of samples in $S_0^{\mathbf{w},\mathbf{b}}$ is placed by the above proposition; for small values of τ, p_0 gets smaller, hence leading to feature noise sensitivity as smaller numbers of data samples are distributed in sets other than $S_2^{\mathbf{w},\mathbf{b}}$. Hence, classification results that are affected significantly by feature noise around the decision boundary can be adjusted for by varying τ accordingly. A similar analysis holds for the other LPTWSVM problems.

5 Numerical experiments

A relative performance analysis of TBSVM, TWSVM, Pin-GTSVM and the proposed LPTWSVM model is given in this section. The experimental evaluation of the models is performed on MATLAB R2017b with a Windows 10 machine and an Intel Xeon(R) Processor (2.30×2 GHz) and 128 GB RAM. We perform the experiments on 20 benchmark UCI datasets (Dheeru & Taniskidou, 2017) and one synthetic dataset. Grid selection is used to select optimal parameters over the following parametric ranges: $c_1 = [10^{-7}, 10^{-6}, \ldots, 10^6, 10^7]$, $c_2 = [10^{-7}, 10^{-6}, \ldots, 10^6, 10^7]$, $c_3 = [10^{-7}, 10^{-6}, \ldots, 10^6, 10^7]$, $c_4 = [10^{-7}, 10^{-6}, \ldots, 10^6, 10^7]$, $\tau = [0.01, 0.2, 0.5, 1.0]$.

To avoid computational complexity, we use $c_1 = c_2$ and $c_3 = c_4$ for the TBSVM and LPTWSVM models. We use a 70 : 30 ratio for training and testing, with 10-fold cross validation for choosing optimal parameters. We employ a Gaussian kernel $K(x, y) = exp^{-||x-y||^2/\gamma^2}$ with parameter values of $\gamma = [10^{-7}, 10^{-6}, \ldots, 10^6, 10^7]$. Details of each dataset are given in Tables 4 and 5. The number of samples, features and classes in each dataset are given as (number of samples × number of features × number of classes). For example, the UCI blood dataset containing 748 samples with feature length 4 and number of classes 2 is given as ($748 \times 4 \times 2$). Table 6 contains 5 publicly available real world classification datasets (Zhang et al., 2019) including two biomedical datasets i.e., Carcinom and Lung, two human face datasets i.e., ORL and Yale and one object recognition datasets i.e., COIL20.

- Biomedical datasets: The Carcinom dataset consists of 174 samples, with each sample represented by a 9182 dimensional feature vector with 11 categories. The Lung dataset consists of 203 instances with 634 dimensional vectors.
- Face image datasets: The ORL face image dataset consists of 400 images and 40 categories, with each image of the size 32×32. The Yale face database consists of 165 face images belonging to 15 different people with each image of the size 32×32.
- Object recognition datasets: COIL20 dataset consists of 20 categories with 1440 sample size, each sample of 1024 dimensions.

For the above real world application datasets, we repeated the 4-fold cross validation 5 times and report the average accuracy and the standard deviation in Table 6. We used one-vs-all strategy in multiclass datasets for all the classification models.

5.1 Synthetic dataset

One of the main properties of LPTWSVM is that it is insensitive to noisy samples in and around the decision boundary which can perturb the hyperplanes and, eventually, the performance of the model. We analyse LPTWSVM's robustness to noise in Fig. 1, where we take synthetic data samples from two Gaussian distributions (the number of samples in

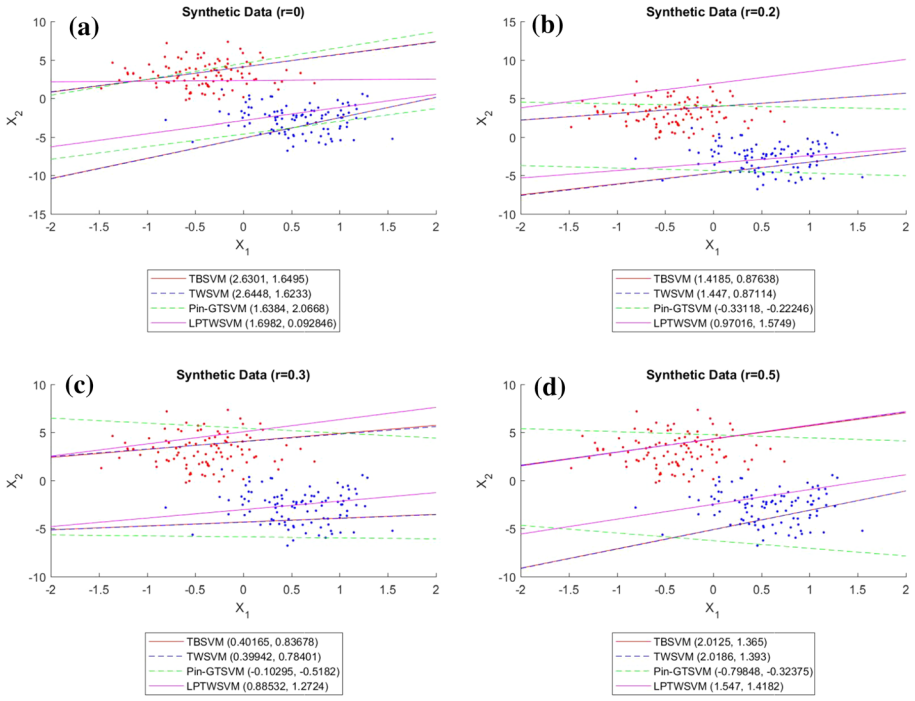

Fig. 1 Figures showing noise insensitivity properties of the proposed LPTWSVM model compared to the baseline methods TBSVM, TWSVM and Pin-GTSVM. In the figures, we have $r = 0$ (noise free), $r = 0.2$, $r = 0.3$ and $r = 0.5$. Here, r is defined as $r = \frac{\text{total number of noise samples}}{\text{total number of samples in the synthetic dataset}}$. The legend below each figure is used to match the separating hyperplanes with the corresponding models. The values in the bracket after each model name are the slopes of the lower and upper hyperplanes of that model, respectively

both the classes are kept equal to 100). The two distributions are: $x_i, i \in \{i : y_i = 1\} \sim \mathcal{N}(\mu_1, \sum_1)$ and $x_i, i \in \{i : y_i = -1\} \sim \mathcal{N}(\mu_2, \sum_2)$ where $\mu_1 = [0.5, -3]^T, \mu_2 = [-0.5, 3]^T$ and $\sum_1 = \sum_2 = \begin{bmatrix} 0.2 & 0 \\ 0 & 3 \end{bmatrix}$. These samples constitute our synthetic dataset and the above Gaussian distributions can be separated by the Bayes classifier whose separating hyperplane is $f_c(x) = 2.5x(1) - x(2)$. The data samples are now contaminated with noise where the noisy samples are also Gaussian distributed: $x_{noise} \sim \mathcal{N}(\mu_{noise}, \sum_{noise})$ where $\mu_{noise} = [0, 0]^T$ and $\sum_{noise} = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}$. The total number of noisy samples is calculated by the value of r, which is defined as $r = \frac{\text{total number of noise samples}}{\text{total number of samples in the synthetic dataset}}$. These noisy samples are then assigned to either of class $+1$ or class -1 with equal probability. Despite the noise introduced to the synthetic dataset affecting labels in and around the decision boundaries, the Bayes classifier remains unchanged. In the results mentioned in Fig. 1, $\tau = 0.01$ for Pin-GTSVM and LPTWSVM and all penalty parameters (that is, the c constants) for the four models TBSVM, TWSVM, Pin-GTSVM and LPTWSVM are equal to 1. In the figures, it can be observed that as the amount of noise is increased from $r = 0$, $r = 0.2$, $r = 0.3$ to $r = 0.5$, the separating hyperplanes of Pin-GTSVM are affected a lot by the feature noise and their slope values deviate a lot, going from positive to negative values. TBSVM and TWSVM also decrease and then ultimately increase their slope values by big absolute values as noise is increased. On the other hand, the absolute change in the slopes of the hyperplanes of our model LPTWSVM is generally by the smallest amounts out of the four models as noise levels are progressively increased. This comparative robustness in the absolute change of slope values of our model implies the noise insensitivity of LPTWSVM.

5.2 Performance scaling evaluation on NDC datasets

Experiments conducted on progressively large-scale binary classification datasets are set out in Table 2. Here, the NDC Data Generator (Musicant, 1998) is used to assess the computational efficiency of the tested models with respect to linearly-increasing training set sizes (methodological parameters are fixed to be identical across all classifiers: $c_1 = 1, c_2 = 1, \gamma = 1$ and $\tau = 0.5$).

Table 3 shows the execution time results of the non-linear TBSVM, TWSVM, Pin-GTSVM and LPTWSVM. From the table, one can see that the existing models (TBSVM, TWSVM and Pin-GTSVM) rapidly become infeasible—in fact exceeded memory constraints—as the dataset size exceeds 70k. However, the proposed LPTWSVM model still functions, illustrating the efficacy of the proposed approach.

Table 2 Details of NDC datasets

Datasets	Training data	Testing data	Features
NDC-20k	20,000	200	32
NDC-30k	30,000	300	32
NDC-50k	50,000	500	32
NDC-70k	70,000	700	32
NDC-80k	80,000	800	32
NDC-90k	90,000	900	32

Table 3 Comparison of execution times of different models using the Gaussian kernel

Datsets	TBSVM Time(s)	TWSVM Time(s)	Pin-GTSVM Time(s)	LPTWSVM Time(s)
NDC-20k	178.35	162.773	409.553	1067.73
NDC-30k	480.201	433.563	1307.55	2622.16
NDC-50k	1874.07	1513.32	5742.23	5813.61
NDC-70k	4838.37	5774.45	17247.5	14622.2
NDC-80k	*	*	*	15382.8
NDC-90k	*	*	*	19819.1

*Denotes out of memory

5.3 Results comparison and discussion

We analyze the relative performance of the proposed LPTWSVM, TBSVM, TWSVM and Pin-GTSVM models. The experimental results corresponding to the linear and Gaussian kernels are given in Tables 4 and 5, respectively. From Table 4, representing the linear kernel, it may be seen that the proposed LPTWSVM and TWSVM models achieve the best performance in 5 datasets while Pin-GTSVM and TBSVM emerge as overall winners in datasets 1 and 2, respectively. The average prediction accuracy of the proposed LPTWSVM is 82.74%; better than other baseline models. Also, it may be seen that the proposed LPTWSVM achieved lowest average rank among the baseline models.

The Gaussian kernel results presented in Table 5 show that the average prediction accuracy of the proposed LPTWSVM is 85%; again better than the other baseline models. It can be seen from Table 5 that the proposed LPTWSVM approach emerged as overall winner in 5 datasets; a greater number than Pin-GTSVM, TWSVM and TBSVM, which achieved best performance for 4, 3 and 2 datasets, respectively.

Table 6 gives the performance of the classification models on the datasets from different with large features. One can see that the performance of the proposed LPTWSVM model on COIL20, Yale, Lung and Carcinom is better as compared to the baseline models. In COIL20 dataset, the proposed LPTWSVM model achieved 98.73% accuracy which is better than the baseline models. Also, in Yale face dataset the performance of the proposed model is better compared to baseline models. In biological datasets i.e., Lung and Carcinom, the performance of the proposed LPTWSVM is better compared to the baseline models.

5.4 Statistical analysis

We carry out an analysis of the statistical significance of the performance values obtained for the different models via Friedman testing. In Friedman testing, models are ranked for each dataset separately with the best performing model being allocated the lowest rank. The results of this test for the models for linear kernel is given in Table 4 and the Gaussian kernel results are given in Table 5.

Table 4 Classification Accuracy[1] of TBSVM, TWSVM, Pin-GTSVM and LPTWSVM using linear kernel

Datasets	TBSVM	TWSVM	Pin-GTSVM	LPTWSVM
	Accuracy	Accuracy	Accuracy	Accuracy
Dataset size	$c_1, c_3, Time(s)$	$c_1, Time(s)$	$c_1, \tau, Time(s)$	$c_1, c_3, \tau, Time(s)$
Blood	75.89	**78.57**	71.88	75
$(748 \times 4 \times 2)$	$1, 10^4, 0.05653$	$10, 0.14843$	$0.1, 1, 0.21711$	$10^{-2}, 10, 0.2, 0.29742$
Breast-cancer-wisc-diag	95.32	**95.91**	92.98	**95.91**
$(569 \times 30 \times 2)$	$10^2, 10^3, 0.2004$	$10^{-1}, 0.03925$	$10^{-1}, 1, 0.06338$	$10^{-2}, 10, 10^{-2}, 0.3631$
Breast-cancer-wisc	97.62	**98.1**	98.1	95.71
$(699 \times 9 \times 2)$	$10^{-2}, 10^3, 0.40346$	$10^{-1}, 0.05771$	$1, 0.2, 0.09911$	$1, 10, 10^{-2}, 0.63932$
Breast-cancer	70.93	70.93	**76.74**	73.26
$(286 \times 9 \times 2)$	$1, 10^2, 0.05934$	$10, 0.0324$	$10^{-2}, 10^{-2}, 0.01797$	$10^{-3}, 10^{-2}, 10^{-1}, 0.00756$
Conn-bench-sonar-mines-rocks	72.58	67.74	70.97	**74.19**
$(208 \times 60 \times 2)$	$10^2, 10^2, 0.03141$	$10^{-1}, 0.0296$	$1, 0.5, 0.03053$	$10^{-2}, 10^2, 10^{-2}, 0.00794$
Credit-approval	**87.44**	85.99	74.40	86.47
$(690 \times 15 \times 2)$	$1, 10^{-3}, 0.08104$	$10^{-2}, 0.01212$	$10^{-2}, 0.2, 0.06422$	$10^2, 10^4, 10^{-1}, 0.38272$
Heart-hungarian	80.68	**81.82**	81.82	**81.82**
$(294 \times 12 \times 2)$	$10^{-7}, 10^6, 0.00257$	$10^{-5}, 0.00318$	$1, 10^{-2}, 0.02185$	$10^{-7}, 10^6, 10^{-2}, 0.0018$
Ilpd-indian-liver	69.14	**70.86**	70.86	70.29
$(583 \times 9 \times 2)$	$10^7, 10^{-3}, 0.1018$	$10^4, 0.11586$	$10^6, 10^{-2}, 0.03423$	$10^{-2}, 10, 10^{-1}, 0.12589$
Ionosphere	**90.48**	83.81	89.52	**90.48**
$(351 \times 33 \times 2)$	$10^{-5}, 10^2, 0.00286$	$10^{-1}, 0.01815$	$10^{-2}, 0.5, 0.02805$	$10, 10^3, 0.1, 0.14433$
Mammographic	82.29	**84.72**	83.33	83.33
$(961 \times 5 \times 2)$	$10^{-2}, 10, 0.02634$	$10^{-2}, 0.01824$	$1, 10^{-2}, 0.07261$	$10^2, 10^4, 0.01, 0.78302$
Molec-biol-promoter	81.25	68.75	78.13	**84.38**
$(106 \times 57 \times 2)$	$10^5, 10^2, 0.00998$	$10^4, 0.00201$	$10^{-3}, 10^{-1}, 0.02547$	$10^5, 10^4, 10^{-2}, 0.04061$
Parkinsons	82.76	**91.38**	91.38	81.03
$(195 \times 22 \times 2)$	$1, 10^{-4}, 0.01542$	$1, 0.03424$	$10^{-4}, 10^{-1}, 0.02495$	$10^{-1}, 1, 10^{-1}, 0.08794$
Pittsburg-bridges-T-OR-D	87.1	87.1	74.2	**90.32**
$(102 \times 4 \times 2)$	$10^7, 10^{-2}, 0.00737$	$10, 0.02153$	$10^{-7}, 10^{-1}, 0.02503$	$10^{-6}, 10^{-3}, 1, 0.0005$
Statlog-German-credit	**77.67**	76	74.67	76.67
$(1000 \times 24 \times 2)$	$10^{-4}, 10^7, 0.01335$	$1, 0.172$	$10^{-1}, 1, 0.07532$	$10^{-3}, 10^2, 10^{-2}, 0.03945$
Statlog-heart	85.19	83.95	85.19	**86.42**
$(270 \times 13 \times 2)$	$1, 10^6, 0.00251$	$10^{-2}, 0.00485$	$10^{-7}, 10^{-2}, 0.0279$	$10^{-6}, 10^{-1}, 10^{-1}, 0.00152$
Vertebral-column-2clases	75.27	64.52	77.42	**78.49**
$(310 \times 6 \times 2)$	$10^{-1}, 10^{-6}, 0.06243$	$10, 0.09682$	$10^{-4}, 1, 0.03432$	$10, 1, 10^{-2}, 0.18321$
Average Accuracy	81.98	80.63	80.72	**82.74**
Average Rank	2.59	2.5	2.88	**2.03**

Accuracy[1] denotes the average accuracy and bold denotes best accuracy

Table 5 Classification Accuracy[1] of TBSVM, TWSVM, Pin-GTSVM and LPTWSVM using Gaussian kernel

Datasets	TBSVM	TWSVM	Pin-GTSVM	LPTWSVM
	Accuracy	Accuracy	Accuracy	Accuracy
Dataset size	$c_1, c_3, \gamma, Time(s)$	$c_1, \gamma, Time(s)$	$c_1, \tau, \gamma, Time(s)$	$c_1, c_3, \tau, \gamma, Time(s)$
Blood	78.13	77.68	75.9	**80.36**
$(748 \times 4 \times 2)$	$10^{-1}, 10^{-1}, 10, 0.08478$	$10^{-1}, 10, 0.15009$	$10^{-6}, 0.5, 10^{5}, 0.3264$	$1, 10, 0.5, 1, 0.29148$
Breast-cancer-wisc-diag	**98.83**	96.49	97.07	96.49
$(569 \times 30 \times 2)$	$10^{7}, 10^{2}, 10, 0.09208$	$10^{-1}, 10, 0.04892$	$10^{-7}, 10^{-1}, 10^{2}, 0.06715$	$1, 1, 10^{-1}, 10^{-2}, 0.20238$
Breast-cancer-wisc	98.1	**99.52**	98.57	98.1
$(699 \times 9 \times 2)$	$10^{7}, 10^{7}, 10, 0.28641$	$10, 10^{2}, 0.14659$	$10^{-2}, 0.5, 10^{3}, 0.17899$	$10^{2}, 10^{4}, 10^{-2}, 10^{-3}, 0.32211$
Breast-cancer	72.09	75.58	**75.58**	70.93
$(286 \times 9 \times 2)$	$10^{-7}, 10^{-6}, 10, 0.00811$	$10^{5}, 10, 0.05946$	$1, 10^{-2}, 10^{6}, 0.0576$	$10^{5}, 10^{6}, 10^{-2}, 10^{-3}, 0.14166$
Conn-bench-sonar-mines-rocks	83.87	80.65	82.26	**85.48**
$(208 \times 60 \times 2)$	$10^{-4}, 10^{-3}, 10^{2}, 0.11512$	$10, 10^{2}, 0.04094$	$10^{4}, 10^{-1}, 10^{2}, 0.0321$	$10^{2}, 10, 0.2, 10^{-2}, 0.04921$
Credit-approval	85.51	83.09	84.06	**90.82**
$(690 \times 15 \times 2)$	$10^{-1}, 10^{-7}, 10^{3}, 0.08843$	$10^{-1}, 10, 0.07559$	$10^{-2}, 0.2, 10^{2}, 0.09163$	$10^{-2}, 1, 10^{-1}, 10^{-2}, 0.21094$
Heart-hungarian	79.55	**84.09**	80.68	**84.09**
$(294 \times 12 \times 2)$	$10, 10^{3}, 10, 0.03217$	$1, 10^{2}, 0.02811$	$10^{-5}, 0.5, 10^{2}, 0.02858$	$10, 10^{2}, 10^{-2}, 10^{-2}, 0.08339$
Ilpd-indian-liver	67.43	69.71	**71.43**	67.43
$(583 \times 9 \times 2)$	$10^{7}, 1, 10^{7}, 0.04488$	$10^{-3}, 10^{-1}, 0.04855$	$10^{2}, 1, 1, 0.07741$	$10^{5}, 10^{7}, 0.2, 10^{-4}, 0.16934$
Ionosphere	**96.19**	94.29	93.33	93.33
$(351 \times 33 \times 2)$	$1, 10^{-1}, 10, 0.0313$	$10^{5}, 10, 0.03816$	$10, 10^{-2}, 10, 0.03784$	$10^{6}, 10^{-1}, 0.5, 10^{-1}, 0.09088$
Mammographic	82.99	**85.76**	84.03	81.6
$(961 \times 5 \times 2)$	$1, 10^{-3}, 10^{2}, 0.28458$	$10^{-6}, 10^{2}, 0.11175$	$10^{-3}, 0.2, 10^{3}, 0.21413$	$10, 10^{2}, 10^{-1}, 10^{-1}, 0.50026$
Molec-biol-promoter	84.38	78.13	84.38	**90.63**
$(106 \times 57 \times 2)$	$10^{-4}, 10^{-3}, 10^{3}, 0.00344$	$10^{-3}, 10^{2}, 0.00268$	$10^{-5}, 0.2, 10^{4}, 0.0321$	$10^{5}, 10^{-1}, 10^{-1}, 10^{-2}, 0.02168$
Parkinsons	94.83	87.93	89.66	**96.55**
$(195 \times 22 \times 2)$	$10, 10^{-3}, 10, 0.03194$	$10^{7}, 1, 0.00484$	$10^{2}, 0.5, 1, 0.01753$	$10^{5}, 1, 0.5, 10^{-1}, 0.05707$

Table 5 (continued)

Datasets	TBSVM	TWSVM	Pin-GTSVM	LPTWSVM
	Accuracy	Accuracy	Accuracy	Accuracy
Dataset size	$c_1, c_3, \gamma, Time(s)$	$c_1, \gamma, Time(s)$	$c_1, \tau, \gamma, Time(s)$	$c_1, c_3, \tau, \gamma, Time(s)$
Pittsburg-bridges-T-OR-D	80.65	87.1	**90.32**	83.87
$(102 \times 4 \times 2)$	$10^{-5}, 10, 10, 0.00373$	$10^{5}, 10, 0.00938$	$10^{3}, 1, 10, 0.02504$	$10^{2}, 10^{2}, 0.2, 10^{-1}, 0.02163$
Statlog-German-credit	69	**76.67**	72.33	71.33
$(1000 \times 24 \times 2)$	$10^{-2}, 10^{-7}, 10, 0.12822$	$1, 10^{2}, 0.24544$	$10^{-1}, 0.5, 10^{6}, 0.21836$	$10^{-1}, 10, 0.2, 10^{-2}, 0.48482$
Statlog-heart	**90.12**	82.72	86.42	**90.12**
$(270 \times 13 \times 2)$	$10^{-6}, 10^{-6}, 10^{5}, 0.01283$	$10^{-3}, 10, 0.00675$	$10^{-1}, 0.2, 10^{5}, 0.03897$	$10^{-1}, 1, 10^{-2}, 10^{-2}, 0.10612$
Vertebral-column-2clases	78.49	79.57	**83.87**	79.57
$(310 \times 6 \times 2)$	$1, 10^{-3}, 10, 0.10668$	$1, 1, 0.02786$	$10^{-7}, 10^{-2}, 10, 0.02829$	$10, 10, 10^{-2}, 10^{-1}, 0.06889$
Average Accuracy	83.76	83.69	84.37	**85.04**
Average Rank	2.66	2.59	**2.31**	2.44

Accuracy[1] denotes the average accuracy and bold denotes best accuracy

Table 6 Classification Accuracy of TBSVM, TWSVM, Pin-GTSVM, SPTWSVM and LPTWSVM on real world application datasets

Domain	Dataset	*Samples* × *Features* × *Class*	TWSVM *Acc.* ± *Std.*	TBSVM *Acc.* ± *Std.*	Pin-GTSVM *Acc.* ± *Std.*	SPTWSVM *Acc.* ± *Std.*	LPTWSVM *Acc.* ± *Std.*
Object	COIL20	1440 × 1024 × 20	97.13 ± 0.0088	97.94 ± 0.0079	71.37 ± 0.0285	98.61 ± 0.0036	98.73 ± 0.0178
Face	ORL	400 × 1024 × 40	88.25 ± 0.0431	88.25 ± 0.0431	87.05 ± 0.0296	87.05 ± 0.0296	88.02 ± 0.0345
Face	Yale	165 × 1024 × 15	67.86 ± 0.0594	67.86 ± 0.0594	74.78 ± 0.0591	74.9 ± 0.058	75.38 ± 0.0444
Biology	Lung	203 × 3312 × 5	91.82 ± 0.041	91.82 ± 0.041	91.81 ± 0.0422	91.22 ± 0.0409	92.57 ± 0.1727
Biology	Carcinom	174 × 9182 × 11	90.66 ± 0.0521	90.66 ± 0.0521	91.35 ± 0.0542	90.32 ± 0.0528	91.72 ± 0.0854

5.4.1 Linear results

Under the null hypothesis, the Friedman statistics are distributed according to χ_F^2 with $(k-1)$ degrees of freedom as follows (Demšar, 2006):

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

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

where $R_j = \frac{1}{N}\sum_j r_i^j$ and r_i^j denotes the rank of the j^{th} algorithm on the i^{th} dataset out of k algorithms and N datasets with $(k-1)$ and $(k-1)(N-1)$ degrees of freedom. Here, we are comparing the four algorithms on 16 datasets i.e. $k=4$ and $N=16$. Also, the average ranks of TBSVM, TWSVM, Pin-GTSVM and the proposed LPTWSVM are 2.59375, 2.5, 2.875 and 2.03125, respectively. Therefore,

$$\begin{aligned}\chi_F^2 =& \frac{12\times 16}{4(4+1)}\left[2.59375^2 + 2.5^2 + 2.875^2 + 2.03125^2\right.\\ &\left. -\frac{4\times 5\times 5}{4}\right] = 3.5437,\\ F_F =& \frac{(16-1)\times 3.5437}{16\times(4-1)-3.5437} = 1.1956.\end{aligned}$$

For $k=4, N=16$, the critical values of $F(3, 45)$ for $\alpha = 0.05$ is 2.815. Thus, Friedman test fails to detect the significant difference among the models. However, one can see that the proposed LPTWSVM model achieved better average accuracy compared to the existing models. Also, the average rank of the proposed LPTWSVM model is better compared to baseline models.

5.4.2 Non-linear results

Under the null hypothesis, the Friedman statistics are distributed according to χ_F^2 with $(k-1)$ degrees of freedom as follows (Demšar, 2006):

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right], \tag{48}$$

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}. \tag{49}$$

Similar to linear case, here also $k=4$ and $N=16$. The average ranks of the TBSVM, TWSVM, Pin-GTSVM and the proposed LPTWSVM with Gaussian kernel are 2.65625, 2.59375, 2.3125 and 2.4375, respectively. Therefore,

$$\chi_F^2 = \frac{12 \times 16}{4(4+1)}\left[2.65625^2 + 2.59375^2 + 2.3125^2 + 2.4375^2 - \frac{4 \times 5 \times 5}{4}\right] = 0.6941,$$

$$F_F = \frac{(16-1) \times 20.1852}{16 \times (4-1) - 20.1852} = 0.2201.$$

For $k = 4, N = 16$, the critical values of $F(3, 45)$ with $\alpha = 0.05$ is 2.815. Thus, Friedman test fails to detect the significant difference among the models. However, one can see that the proposed LPTWSVM model achieved better average accuracy compared to the existing models. Also, the average rank of the proposed LPTWSVM model is better compared to TBSVM and TWSVM models.

6 Conclusions

In this paper, we have proposed a novel classification model, LPTWSVM. In contrast to the TWSVM, the proposed model is tunably-insensitive to feature noise while exhibiting greater stability under resampling. Furthermore, a structural risk minimization principle is directly implemented within the proposed LPTWSVM model to ensure better generalization. Numerical experiments conducted on standard benchmark datasets with respect to both linear as well as non-linear implementations show the validity of the proposed LPTWSVM approach, for which the classification performance is similar or better than the baseline methods. We further performed experiments on progressively-increased NDC dataset sizes to demonstrate the effectiveness of the proposed LPTWSVM model on large-scale datasets. Finally, we note that in the regularised LPTWSVM, additional parameters need to be tuned via cross-validation; future work will focus on the appropriate mechanisms for automatic selection of these parameters.

Acknowledgements This work was supported by Science and Engineering Research Board (SERB) as Early Career Research Award 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.

Author Contributions M. Tanveer: Conceptualization, Methodology, Formal analysis, Investigation, Resources, Writing—review & editing, Supervision, Funding acquisition. A. Tiwari: Conceptualization, Methodology, Validation, Resources. R. Choudhary: Conceptualization, Methodology, Validation, Resources, Writing—original draft, Writing—review & editing, Visualization. M.A. Ganaie: Conceptualization, Methodology, Validation, Resources, Writing—original draft, Writing—review & editing, Visualization.

Availability of data and material The datasets are the benchmark datasets available online (Data Source available in manuscript.)

Code availability The codes will be available at https://github.com/mtanveer1.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethics approval The study is original and has not been submitted to any other journal/conference.

References

Borgwardt, K. M. (2011). Kernel methods in bioinformatics. In *Handbook of statistical bioinformatics* (pp. 317–334). Springer.

Cao, L.-J., & Tay, F. E. H. (2003). Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on Neural Networks, 14*(6), 1506–1518.

Chapelle, O., Vapnik, V., Bousquet, O., & Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning, 46*(1–3), 131–159.

Chen, X., Yang, J., Ye, Q., & Liang, J. (2011). Recursive projection twin support vector machine via within-class variance minimization. *Pattern Recognition, 44*(10–11), 2643–2655.

Cheong, S., Oh, S. H., & Lee, S.-Y. (2004). Support vector machines with binary tree architecture for multi-class classification. *Neural Information Processing Letters and Reviews, 2*(3), 47–51.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1–30.

Déniz, O., Castrillon, M., & Hernández, M. (2003). Face recognition using independent component analysis and support vector machines. *Pattern Recognition Letters, 24*(13), 2153–2157.

Dheeru, D. & Karra Taniskidou, E. (2017). UCI machine learning repository [Online]. Available: http://archive.ics.uci.edu/ml

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research, 15*(1), 3133–3181.

Fung, G. M., & Mangasarian, O. L. (2005). Multicategory proximal support vector machine classifiers. *Machine Learning, 59*(1–2), 77–97.

Gao, S., Ye, Q., & Ye, N. (2011). 1-Norm least squares twin support vector machines. *Neurocomputing, 74*(17), 3590–3597.

González-Castano, F. J., García-Palomares, U. M., & Meyer, R. R. (2004). Projection support vector machine generators. *Machine Learning, 54*(1), 33–44.

Huang, X., Shi, L., & Suykens, J. A. (2014). Support vector machine classifier with pinball loss. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 36*(5), 984–997.

Jayadeva, Khemchandani, R. & Chandra, S. (2007). Twin support vector machines for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(5), 905–910.

Kumar, M. A., & Gopal, M. (2008). Application of smoothing technique on twin support vector machines. *Pattern Recognition Letters, 29*(13), 1842–1848.

Kumar, M. A., & Gopal, M. (2009). Least squares twin support vector machines for pattern classification. *Expert Systems with Applications, 36*(4), 7535–7543.

Kumar, M. A., Khemchandani, R., Gopal, M., & Chandra, S. (2010). Knowledge based least squares twin support vector machines. *Information Sciences, 180*(23), 4606–4618.

Madzarov, G., Gjorgjevikj, D., & Chorbev, I. (2009). A multi-class SVM classifier utilizing binary decision tree. *Informatica, 33*(2)

Mangasarian, O. L., & Wild, E. W. (2006). Multisurface proximal support vector machine classification via generalized eigenvalues. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(1), 69–74.

Musicant, D. (1998). Normally distributed clustered datasets, Computer Sciences Department, University of Wisconsin, Madison. http://www.cs.wisc.edu/dmi/svm/ndc

Noble, W. S. (2004). Support vector machine applications in computational biology. *Kernel Methods in Computational Biology, 71*, 92.

Peng, X. (2010). TSVR: An efficient twin support vector machine for regression. *Neural Networks, 23*(3), 365–372.

Qi, Z., Tian, Y., & Shi, Y. (2013). Robust twin support vector machine for pattern classification. *Pattern Recognition, 46*(1), 305–316.

Richhariya, B., & Tanveer, M. (2018). EEG signal classification using universum support vector machine. *Expert Systems with Applications, 106*, 169–182.

Richhariya, B., & Tanveer, M. (2020). A reduced universum twin support vector machine for class imbalance learning. *Pattern Recognition, 102*, 107150.

Shao, Y.-H., Chen, W.-J., Huang, W.-B., Yang, Z.-M., & Deng, N.-Y. (2013). The best separating decision tree twin support vector machine for multi-class classification. *Procedia Computer Science, 17*, 1032–1038.

Shao, Y.-H., Zhang, C.-H., Wang, X.-B., & Deng, N.-Y. (2011). Improvements on twin support vector machines. *IEEE Transactions on Neural Networks, 22*(6), 962–968.

Sharma, S., Rastogi, R., & Chandra, S. (2021). Large-scale twin parametric support vector machine using pinball loss function. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51*(2), 987–1003.
Singla, M., Ghosh, D., Shukla, K. & Pedrycz, W. (2020). Robust twin support vector regression based on rescaled hinge loss. *Pattern Recognition,* 107395
Tanveer, M. (2015). Robust and sparse linear programming twin support vector machines. *Cognitive Computation, 7*(1), 137–149.
Tanveer, M., Khan, M. A., & Ho, S.-S. (2016a). Robust energy-based least squares twin support vector machines. *Applied Intelligence, 45*(1), 174–186.
Tanveer, M., Mangal, M., Ahmad, I., & Shao, Y.-H. (2016b). One norm linear programming support vector regression. *Neurocomputing, 173*, 1508–1518.
Tanveer, M., Rajani, T., Rastogi, R., & Shao, Y. (2021). Comprehensive review on twin support vector machines. arXiv preprint arXiv:2105.00336
Tanveer, M., Sharma, A., & Suganthan, P. N. (2019a). General twin support vector machine with pinball loss function. *Information Sciences, 494*, 311–327.
Tanveer, M., Tiwari, A., Choudhary, R., & Jalan, S. (2019b). Sparse pinball twin support vector machines. *Applied Soft Computing, 78*, 164–175.
Tian, Y., & Ping, Y. (2014). Large-scale linear nonparallel support vector machine solver. *Neural Networks, 50*, 166–174.
Trafalis, T. B. & Ince, H. (2000). Support vector machine for regression and applications to financial forecasting. In *Proceedings of the IEEE-INNS-ENNS international joint conference on neural networks, 2000. IJCNN 2000* (Vol. 6. pp. 348–353). IEEE.
Valentini, G., Muselli, M., & Ruffino, F. (2004). Cancer recognition with bagged ensembles of support vector machines. *Neurocomputing, 56*, 461–466.
Van Gestel, T., Suykens, J. A., Baesens, B., Viaene, S., Vanthienen, J., Dedene, G., De Moor, B., & Vandewalle, J. (2004). Benchmarking least squares support vector machine classifiers. *Machine Learning, 54*(1), 5–32.
Vapnik, V. (1998). *Statistical learning theory. 1998* (Vol. 3). Wiley.
Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks, 10*(5), 988–999.
Vapnik, V. (2013). *The nature of statistical learning theory*. Springer.
Wang, H., Xu, Y., & Zhou, Z. (2020). Twin-parametric margin support vector machine with truncated pinball loss. *Neural Computing and Applications,* 1–18.
Xu, Y., & Wang, L. (2014). K-nearest neighbor-based weighted twin support vector regression. *Applied Intelligence, 41*(1), 299–309.
Yan, H., Ye, Q.-L., & Yu, D.-J. (2019). Efficient and robust twsvm classification via a minimum l1-norm distance metric criterion. *Machine Learning,* 1–26.
Zhang, Y., Wu, J., Cai, Z., Du, B., & Philip, S. Y. (2019). An unsupervised parameter learning model for RVFL neural network. *Neural Networks, 112*, 85–97.