Appl Intell (2017) 46:124–134
DOI 10.1007/s10489-016-0809-8

A new approach for training Lagrangian twin support vector machine via unconstrained convex minimization

S. Balasundaram[1] · Deepak Gupta[1] · Subhash Chandra Prasad[1]

Published online: 29 July 2016

Abstract In this paper, a novel unconstrained convex minimization problem formulation for the Lagrangian dual of the recently introduced twin support vector machine (TWSVM) in simpler form is proposed for constructing binary classifiers. Since the objective functions of the modified minimization problems contain non-smooth 'plus' function, we solve them by Newton iterative method either by considering their generalized Hessian matrices or replacing the 'plus' function by a smooth approximation function. Numerical experiments were performed on a number of interesting real-world benchmark data sets. Computational results clearly illustrates the effectiveness and the applicability of the proposed approach as comparable or better generalization performance with faster learning speed is obtained in comparison with SVM, least squares TWSVM (LS-TWSVM) and TWSVM.

Keywords Generalized Hessian approach · Smooth approximation formulation · Twin support vector machine

✉ S. Balasundaram
balajnu@gmail.com; bala_jnu@hotmail.com

Deepak Gupta
deepakjnu85@gmail.com

Subhash Chandra Prasad
subhchandra.jnu@gmail.com; subh_net@yahoo.com

[1] School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, 110067, India

1 Introduction

Support vector machines (SVMs), proposed by Vapnik [25], are computationally powerful machine learning tools applied to classification. They have been successfully applied to problems of practical importance from wider areas like face detection [18], gene prediction [7], and text characterization [10].

The objective of SVM is in determining a separating hyperplane by maximizing the margin between the samples of positive and negative classes and assigning class labels to test samples in accordance with the half-space in which they lie. It is well-known that SVM determines the optimal hyperplane as the solution of a quadratic programming problem (QPP) having linear inequality constraints [3, 25].

As SVM derives a robust, sparse, global solution owning better generalization ability than other machine learning approaches like artificial neural networks, it becomes the state of the art method for classification. However, one of the major challenges of SVM is its high computational cost associated with training which restricts its application to problems with large data sets. To overcome this problem, over the past decade, efficient learning algorithms and models have been proposed in the literature [2, 9, 11, 16, 19, 24]. Recently, Mangasarian and Wild [16] proposed a new SVM model called generalized eigenvalue proximal SVM (GEPSVM) wherein the binary classification is obtained by constructing two non-parallel hyperplanes having the property that samples of each class will be clustered around its corresponding hyperplane. This results in solving two generalized eigenvalue problems. Similar in sprit to GEPSVM, a novel formulation called twin SVM (TWSVM) has been proposed in [9] wherein two non-parallel hyperplanes are

constructed by solving two QPPs of smaller size than solving a single QPP as in the case of the standard SVM. This strategy makes TWSVM work four times faster than the standard SVM and further showing good generalization ability [9, 12]. Due to these advantages, TWSVM becomes one of the most popular methods for classification. For the interesting work on least squares TWSVM (LS-TWSVM) and smooth TWSVM, see [11, 12]. For other extensions to TWSVM, the interested reader is referred to [20, 21, 23].

With the aim of obtaining an efficient TWSVM model, following the novel approach in solving the dual SVM [1, 26], a naïve unconstrained Lagrangian twin SVM (ULTSVM) formulation has been proposed in this paper. Since the objective functions contain a term having non-smooth 'plus' function, the proposed minimization problems are solved either by considering its generalized Hessian [5, 8] or by introducing the smooth approximation function of [13] in place of the non-smooth 'plus' function and then applying Newton-Armijo algorithm [13]. Its convergence and finite termination will follow directly from the results of [13, 14]. Finally, the effectiveness of the proposed ULTSVM problem is demonstrated by performing experiments on a number of interesting real-world datasets and comparing their results with SVM, LS-TWSVM and TWSVM.

Throughout this work, all vectors are assumed as column vectors. The inner product of two vectors $\mathbf{x}, \mathbf{y}$ in the $n-$dimensional real space $\Re^n$ is denoted by: $\mathbf{x}^t\mathbf{y}$, where $\mathbf{x}^t$ is the transpose of $\mathbf{x}$. For any vector $\mathbf{x} = (x_1, ..., x_n)^t \in \Re^n$, the plus function $\mathbf{x}_+$ is defined as: $(\mathbf{x}_+)_i = \max\{0, x_i\}$ and $i = 1, ..., n$. The 2-norm of a vector $\mathbf{x}$ will be denoted by: $||\mathbf{x}||$. We denote the vector of ones of dimension m by $\mathbf{e}$ and the identity matrix of appropriate size by I. If f is a real valued function of the variable $\mathbf{x} = (x_1, ..., x_n)^t \in \Re^n$ then its gradient vector and Hessian matrix are denoted by: $\nabla f = \left(\partial f / \partial x_1, ..., \partial f / \partial x_n\right)^t$ and $\nabla^2 f = \left(\partial^2 f / \partial x_i \partial x_j\right)_{i,j=1,...,n}$ respectively.

The paper is organized as follows. In Section 2, the standard SVM, LS-TWSVM and TWSVM are reviewed. The proposed unconstrained TWSVM problem in its dual form and Newton iterative method of solving it are described in Section 3. Numerical experiments have been performed on a number of real-world datasets and their results have been compared with that of SVM, LS-TWSVM and TWSVM in Section 4 and finally the conclusions and future work are drawn in Section 5.

2 Related work

In this section, we briefly describe the standard SVM for binary classification problems and one of its important variants, the twin support vector machine.

Consider the binary classification problem assuming that the training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$ be given where for the input sample $\mathbf{x}_i \in \Re^n$ let its corresponding class label be $y_i \in \{-1, +1\}$.

2.1 Support vector machine (SVM)

Assume that the input samples are mapped into a higher dimensional feature space via a nonlinear function $\varphi(.)$. Then, an SVM classifier seeks for an optimal hyperplane of the form $\mathbf{w}^t\varphi(\mathbf{x}) + b = 0$ in the feature space, where the bias term $b \in \Re$ and the vector normal to the hyperplane $\mathbf{w}$ are the unknowns which are determined by solving the following QPP [3, 25]

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \tfrac{1}{2}\mathbf{w}^t\mathbf{w} + C\mathbf{e}^t\boldsymbol{\xi}$$

subject to:

$$y_i\ (\mathbf{w}^t\varphi(\mathbf{x}_i) + b\ )\ \geq\ 1\ -\ \xi_i$$

and

$$\xi_i \geq 0 \text{ for } i = 1, 2, \ldots, m. \tag{1}$$

Here, $\boldsymbol{\xi} = (\xi_1, ..., \xi_m)^t$ is the vector of slack variables and $C > 0$ is a parameter.

Usually, problem (1) is solved by minimizing its Wolfe dual obtained to be

$$\min_{\mathbf{u}}\ \tfrac{1}{2}\sum_{i,j=1}^{m} y_i y_j \varphi(\mathbf{x}_i)^t\varphi(\mathbf{x}_j)u_i u_j - \sum_{i=1}^{m} u_i$$

subject to

$$\sum_{i=1}^{m} u_i y_i = 0 \text{ and } 0 \leq u_i \leq C \text{ for } i = 1, 2, \ldots, m, \tag{2}$$

where $\mathbf{u} = (u_1, ..., u_m)^t \in \Re^m$ is the Lagrangian multiplier vector. In this case, the decision function $f(.)$ is taken as

$$f(\mathbf{x}) = sign\left(\sum_{s=1}^{N_{SV}} u_s y_s \varphi(\mathbf{x})^t\varphi(\mathbf{x}_s) + b\right), \tag{3}$$

where N_{SV} is the number of support vectors $\mathbf{x}_s \in \Re^n$ in which $0 < u_s < C$.

By applying the kernel trick, i.e. taking $k(\mathbf{x}, \mathbf{z}) = \varphi(\mathbf{x})^t\varphi(\mathbf{z})$ for $\mathbf{x}, \mathbf{z} \in \Re^n$ in (2) and (3), where $k(.,.)$ is a given kernel function, the explicit construction of the nonlinear mapping $\varphi(.)$ will be avoided. In this work, the Gaussian kernel function of the form

$$k(\mathbf{x}, \mathbf{z}) = \exp(-\mu\ \|\mathbf{x} - \mathbf{z}\|^2\ )$$

is considered, where $\mu > 0$ is a parameter.

2.2 Twin support vector machine (TWSVM)

Assume that the training set consists of m_1 and m_2 number of samples belonging to class $(+1)$ and class (-1) respectively so that $m = m_1 + m_2$. Further, let the samples from class $(+1)$ and class (-1) be represented by matrices $A \in \Re^{m_1 \times n}$ and $B \in \Re^{m_2 \times n}$ respectively.

Similar to SVM, the nonlinear TWSVM problem can be formulated by mapping the training samples into a higher dimensional feature space via a kernel function $k(.,.)$ and performing the linear TWSVM classification in the feature space. More precisely, TWSVM determines two kernel generated surfaces of the form [9]

$$K(\mathbf{x}^t, C^t)\,\mathbf{w}_1 + b_1 = 0 \text{ and } K(\mathbf{x}^t, C^t)\,\mathbf{w}_2 + b_2 = 0 \tag{4}$$

such that each one of them will be as close as possible to samples of one class and also will be at a distance of at least one unit from samples of the other class where $C = [A\,;\,B]$ is an augmented matrix of size $m \times n$ and $K(\mathbf{x}^t, C^t) = (k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_m))$ is a row vector in $\Re^m$. In fact, unlike solving two eigenvalue problems as in the case of GEPSVM [16], the nonparallel kernel surfaces (4) are obtained by solving the following pair of QPPs defined by [9]

$$\min_{(\mathbf{w}_1, b_1, \boldsymbol{\xi}_2) \in \Re^{m+1+m_2}} \tfrac{1}{2} \left\| K(A, C^t)\,\mathbf{w}_1 + \mathbf{e}_1 b_1 \right\|^2 + C_2 \mathbf{e}_2^t \boldsymbol{\xi}_2$$
$$\text{subject to} \quad -(K(B, C^t)\mathbf{w}_1 + \mathbf{e}_2 b_1) + \boldsymbol{\xi}_2 \geq \mathbf{e}_2, \boldsymbol{\xi}_2 \geq 0 \tag{5a}$$

and

$$\min_{(\mathbf{w}_2, b_2, \boldsymbol{\xi}_1) \in \Re^{m+1+m_1}} \tfrac{1}{2} \left\| K(B, C^t)\mathbf{w}_2 + \mathbf{e}_2 b_2 \right\|^2 + C_1\, \mathbf{e}_1^t \boldsymbol{\xi}_1$$
$$\text{subject to} \quad (K(A, C^t)\mathbf{w}_2 + \mathbf{e}_1 b_2) + \boldsymbol{\xi}_1 \geq \mathbf{e}_1, \boldsymbol{\xi}_1 \geq 0 \tag{5b}$$

where $\boldsymbol{\xi}_1 \in \Re^{m_1}$, $\boldsymbol{\xi}_2 \in \Re^{m_2}$ are vectors of slack variables; $C_1, C_2 > 0$ are the regularization parameters and the unknowns are $\mathbf{w}_1, \mathbf{w}_2 \in \Re^m$ and $b_1, b_2 \in \Re$.

In practice, the solution of the above pair of primal problems (5a) and (5b) is obtained by constructing their Wolfe duals and solving them. Finally, for any test sample $\mathbf{x} \in \Re^n$, its class label is assigned according to its proximity to the non-parallel surfaces, i.e.

$$\text{class } i = \arg \min_{k=1,2} |K(\mathbf{x}^t, C^t)\,\mathbf{w}_k + b_k|, \tag{6}$$

where $|K(\mathbf{x}^t, C^t)\,\mathbf{w}_k + b_k|$ is the perpendicular distance from $\mathbf{x} \in \Re^n$ to the hyperplane $K(\mathbf{x}^t, C^t)\,\mathbf{w}_k + b_k$. For more details on TWSVM, see [9].

2.3 Least squares twin support vector machine (LS-TWSVM)

Similar to the study of least squares SVM (LS-SVM) [24], the extension of TWSVM to least squares TWSVM (LS-TWSVM) was proposed in [11] leading to solving a pair of QPPs with equality constraints. More precisely, the nonlinear LS-TWSVM is defined as

$$\min_{(\mathbf{w}_1, b_1, \boldsymbol{\xi}_2) \in \Re^{m+1+m_2}} \tfrac{1}{2} ||K(A, C^t)\mathbf{w}_1 + \mathbf{e}_1 b_1||^2 + \tfrac{C_2}{2} \boldsymbol{\xi}_2^t \boldsymbol{\xi}_2$$
$$\text{subject to} \quad -(K(B, C^t)\mathbf{w}_1 + \mathbf{e}_2 b_1) + \boldsymbol{\xi}_2 = \mathbf{e}_2 \tag{7a}$$

and

$$\min_{(\mathbf{w}_2, b_2, \boldsymbol{\xi}_1) \in \Re^{m+1+m_1}} \tfrac{1}{2} ||K(B, C^t)\,\mathbf{w}_2 + \mathbf{e}_2 b_2||^2 + \tfrac{C_1}{2} \boldsymbol{\xi}_1^t \boldsymbol{\xi}_1$$
$$\text{subject to} \quad (K(A, C^t)\,\mathbf{w}_2 + \mathbf{e}_1 b_2) + \boldsymbol{\xi}_1 = \mathbf{e}_1. \tag{7b}$$

In fact, on substituting the equality constraints into the object functions, the problems (7a) and (7b) become

$$\min_{(\mathbf{w}_1, b_1) \in \Re^{m+1}} \frac{1}{2} ||K(A, C^t)\,\mathbf{w}_1 + \mathbf{e}_1 b_1||^2 + \frac{C_2}{2} ||K(B, C^t)\,\mathbf{w}_1 + \mathbf{e}_2 b_1 + \mathbf{e}_2||^2 \tag{8a}$$

and

$$\min_{(\mathbf{w}_2, b_2) \in \Re^{m+1}} \frac{1}{2} ||K(B, C^t)\,\mathbf{w}_2 + \mathbf{e}_2 b_2||^2 + \frac{C_1}{2} ||\,K(A, C^t)\,\mathbf{w}_2 + \mathbf{e}_1 b_2 - \mathbf{e}_1\,||^2. \tag{8b}$$

In this case, their solutions become

$$\begin{bmatrix} \mathbf{w}_1 \\ b_1 \end{bmatrix} = -(H^t H + \frac{1}{C_2} G^t G)^{-1} H^t\, \mathbf{e}_2 \text{ and}$$
$$\begin{bmatrix} \mathbf{w}_2 \\ b_2 \end{bmatrix} = (G^t G + \frac{1}{C_1} H^t H)^{-1} G^t\, \mathbf{e}_1,$$

where

$$G = [K(A, C^t)\,\mathbf{e}_1] \text{ and } H = [K(B, C^t)\,\mathbf{e}_2] \tag{9}$$

are augmented matrices of size $m_1 \times (m+1)$ and $m_2 \times (m+1)$ respectively. Since LS-TWSVM results in solving two systems of linear equations it is faster in learning than TWSVM. Further, its classification accuracy results are comparable to TWSVM [11].

3 Proposed unconstrained Lagrangian twin SVM (ULTWSVM)

Following the work of [1, 26], a new variant of TWSVM in its dual is proposed in this section as a pair of unconstrained minimization problems whose solutions will be obtained by the Newton iterative method. We discuss our proposed model for both the linear and nonlinear cases.

Instead of assuming the 1-norm of the vector of slack variables $\boldsymbol{\xi}_k$ with weight $C_k > 0$ in (5a) and (5b) where $k = 1, 2$; the square of the 2-norm of $\boldsymbol{\xi}_k$ with weight $\frac{C_k}{2}$ is minimized in our modified TWSVM formulation. In fact, the linear TWSVM in 2-norm solves the pair of QPPs [9]

$$\min_{(\mathbf{w}_1, b_1, \boldsymbol{\xi}_2) \in \Re^{n+1+m_2}} \frac{1}{2}||A\mathbf{w}_1 + \mathbf{e}_1 b_1||^2 + \frac{C_2}{2}\boldsymbol{\xi}_2^t \boldsymbol{\xi}_2$$

subject to

$$-(B\mathbf{w}_1 + \mathbf{e}_2 b_1) + \boldsymbol{\xi}_2 \geq \mathbf{e}_2 \tag{10a}$$

and

$$\min_{(\mathbf{w}_2, b_2, \boldsymbol{\xi}_1) \in \Re^{n+1+m_1}} \frac{1}{2}||B\mathbf{w}_2 + \mathbf{e}_2 b_2||^2 + \frac{C_1}{2}\boldsymbol{\xi}_1^t \boldsymbol{\xi}_1$$

subject to

$$(A\mathbf{w}_2 + \mathbf{e}_1 b_2) + \boldsymbol{\xi}_1 \geq \mathbf{e}_1, \tag{10b}$$

where $\mathbf{w}_1, \mathbf{w}_2 \in \Re^n$ and $b_1, b_2 \in \Re$ are the unknowns.

Note that since the non-negativity constraints of the slack variables will be automatically satisfied at optimality [15], they have been dropped in (10a) and (10b).

By considering Lagrangian functions and using Karush-Kuhn-Tucker (KKT) conditions, the Wolfe duals of (10a) and (10b) can be formulated as QPPs of the following form

$$\min_{0 \leq \mathbf{u}_1 \in \Re^{m_1}} L_1(\mathbf{u}_1) = \frac{1}{2}\mathbf{u}_1^t Q_1 \mathbf{u}_1 - \mathbf{e}_1^t \mathbf{u}_1 \tag{11a}$$

and

$$\min_{0 \leq \mathbf{u}_2 \in \Re^{m_2}} L_2(\mathbf{u}_2) = \frac{1}{2}\mathbf{u}_2^t Q_2 \mathbf{u}_2 - \mathbf{e}_2^t \mathbf{u}_2 \tag{11b}$$

where

$$Q_1 = \left(\frac{I}{C_1} + G(H^t H)^{-1} G^t\right) \text{ and } Q_2 = \left(\frac{I}{C_2} + H(G^t G)^{-1} H^t\right); \tag{12}$$

and, $G = [A\ \mathbf{e}_1]$ and $H = [B\ \mathbf{e}_2]$ are augmented matrices of sizes $m_1 \times (n+1)$ and $m_2 \times (n+1)$ respectively. Here $\mathbf{u}_1 \in \Re^{m_1}$ and $\mathbf{u}_2 \in \Re^{m_2}$ are Lagrange multipliers satisfying the following

$$\begin{bmatrix} \mathbf{w}_1 \\ b_1 \end{bmatrix} = -(G^t G)^{-1} H^t \mathbf{u}_2 \text{ and } \begin{bmatrix} \mathbf{w}_2 \\ b_2 \end{bmatrix} = (H^t H)^{-1} G^t \mathbf{u}_1. \tag{13}$$

Finally, using the solutions of (11a) and (11b), and (13) one can determine the end classifier (6).

The nonlinear TWSVM in 2-norm determines two non-parallel hyperplanes in the feature space of the form (4) by solving the following pair of QPPs:

$$\min_{(\mathbf{w}_1, b_1, \boldsymbol{\xi}_2) \in \Re^{m+1+m_2}} \frac{1}{2}||K(A, C^t)\mathbf{w}_1 + \mathbf{e}_1 b_1||^2 + \frac{C_2}{2}\boldsymbol{\xi}_2^t \boldsymbol{\xi}_2$$

subject to:

$$-(K(B, C^t)\mathbf{w}_1 + \mathbf{e}_2 b_1) + \boldsymbol{\xi}_2 \geq \mathbf{e}_2$$

and

$$\min_{(\mathbf{w}_2, b_2, \boldsymbol{\xi}_1) \in \Re^{m+1+m_1}} \frac{1}{2}||K(B, C^t)\mathbf{w}_2 + \mathbf{e}_2 b_2||^2 + \frac{C_1}{2}\boldsymbol{\xi}_1^t \boldsymbol{\xi}_1$$

subject to:

$$(K(A, C^t)\mathbf{w}_2 + \mathbf{e}_1 b_2) + \boldsymbol{\xi}_1 \geq \mathbf{e}_1$$

where $\boldsymbol{\xi}_1 \in \Re^{m_1}$, $\boldsymbol{\xi}_2 \in \Re^{m_2}$ are slack variable vectors and $C_1, C_2 > 0$ are input parameters.

With the introduction of Lagrange multipliers $\mathbf{u}_1 \in \Re^{m_1}$ and $\mathbf{u}_2 \in \Re^{m_2}$, the duals of the above problems can be derived again of the same form as (11a) and (11b) where the matrices Q_1, Q_2 are defined by (12). However, in this case, the augmented matrices G and H are given by (9). Using their solutions and (13), the kernel generated functions (4) can be easily obtained.

For the purpose of reformulating the pair of duals (11a) and (11b) into a pair of equivalent, unconstrained minimization problems as our proposed problem formulation and obtaining their solutions by iterative methods, one can rewrite them either for the linear or nonlinear case as

$$\min_{0 \leq \mathbf{u}_1 \in \Re^{m_1}} L_1(\mathbf{u}_1) = \frac{1}{2}\mathbf{u}_1^t \left(\frac{I}{C_1} + \mathcal{G}\,\mathcal{G}^t\right)\mathbf{u}_1 - \mathbf{e}_1^t \mathbf{u}_1 \tag{14}$$

and

$$\min_{0 \leq \mathbf{u}_2 \in \Re^{m_2}} L_2(\mathbf{u}_2) = \frac{1}{2}\mathbf{u}_2^t \left(\frac{I}{C_2} + \mathcal{H}\,\mathcal{H}^t\right)\mathbf{u}_2 - \mathbf{e}_2^t \mathbf{u}_2, \tag{15}$$

where $\mathcal{G} = G(H^t H)^{-1} H^t$ and $\mathcal{H} = H(G^t G)^{-1} G^t$ are matrices of sizes $m_1 \times m_2$ and $m_2 \times m_1$ respectively.

Now, let us consider the minimization problem (14). It can be equivalently written as

$$\min_{\substack{0 \leq \mathbf{u}_1 \in \Re^{m_1} \\ \mathbf{v}_2 \in \Re^{m_2}}} \left(\frac{1}{2C_1}\mathbf{u}_1^t \mathbf{u}_1 - \mathbf{e}_1^t \mathbf{u}_1\right) + \frac{1}{2}\mathbf{v}_2^t \mathbf{v}_2$$

subject to

$$\mathbf{v}_2 = \mathcal{G}^t \mathbf{u}_1 \in \Re^{m_2}.$$

By introducing the Lagrangian multiplier $\mathbf{z}_2 \in \Re^{m_2}$, the dual of the above problem can be expressed as

$$\begin{aligned} &\max_{\mathbf{z}_2 \in \Re^{m_2}} \{ \min_{\substack{0 \leq \mathbf{u}_1 \in \Re^{m_1} \\ \mathbf{v}_2 \in \Re^{m_2}}} [\left(\frac{1}{2C_1}\mathbf{u}_1^t \mathbf{u}_1 - \mathbf{e}_1^t \mathbf{u}_1\right) \\ &\quad + \frac{1}{2}\mathbf{v}_2^t \mathbf{v}_2 + \mathbf{z}_2^t(\mathcal{G}^t \mathbf{u}_1 - \mathbf{v}_2)] \} \\ &= \max_{\mathbf{z}_2 \in \Re^{m_2}} \{ \min_{0 \leq \mathbf{u}_1 \in \Re^{m_1}} [\left(\frac{1}{2C_1}\mathbf{u}_1^t \mathbf{u}_1 - \mathbf{e}_1^t \mathbf{u}_1 + \mathbf{z}_2^t \mathcal{G}^t \mathbf{u}_1\right)] \\ &\quad + \min_{\mathbf{v}_2 \in \Re^{m_2}} (\frac{1}{2}\mathbf{v}_2^t \mathbf{v}_2 - \mathbf{z}_2^t \mathbf{v}_2) \}. \end{aligned} \tag{16}$$

However, it can be easily verified analytically that [22]

$$\min_{0 \leq \mathbf{u}_1 \in \Re^{m_1}} [\left(\frac{1}{2C_1}\mathbf{u}_1^t \mathbf{u}_1 - \mathbf{e}_1^t \mathbf{u}_1 + \mathbf{z}_2^t \mathcal{G}^t \mathbf{u}_1\right)] = -\frac{C_1}{2}||\,(\mathbf{e}_1 - \mathcal{G}\,\mathbf{z}_2)_+||^2 \tag{17}$$

and is attained at

$$\mathbf{u}_1 = C_1\,(\mathbf{e}_1 - \mathcal{G}\,\mathbf{z}_2)_+\,.$$

Similarly,

$$\min_{\mathbf{v}_2 \in \Re^{m_2}} \left(\frac{1}{2}\mathbf{v}_2^t\mathbf{v}_2 - \mathbf{z}_2^t\mathbf{v}_2\right) = -\frac{1}{2}\mathbf{z}_2^t\mathbf{z}_2 \tag{18}$$

when $\mathbf{v}_2 = \mathbf{z}_2$.

Applying the results of (17) and (18) in (16), the dual problem (14) can be written as an unconstrained, strongly convex minimization problem in its simpler form

$$\min_{\mathbf{z}_2 \in \Re^{m_2}} \tilde{L}_1(\mathbf{z}_2) = \frac{1}{2}\mathbf{z}_2^t\mathbf{z}_2 + \frac{C_1}{2}||\,(\mathbf{e}_1 - \mathcal{G}\,\mathbf{z}_2)_+||^2. \tag{19}$$

By following the above procedure, the dual problem (15) can be equivalently written as a strongly convex, minimization problem of the form

$$\min_{\mathbf{z}_1 \in \Re^{m_1}} \tilde{L}_2(\mathbf{z}_1) = \frac{1}{2}\mathbf{z}_1^t\mathbf{z}_1 + \frac{C_2}{2}||\,(\mathbf{e}_2 - \mathcal{H}\,\mathbf{z}_1)_+||^2. \tag{20}$$

Since, the objective functions $\tilde{L}_1(.)$, $\tilde{L}_2(.)$ of our proposed LTWSVM are strongly convex and therefore each problems (19), (20) will have a unique solution. From the above discussion, one can easily conclude that if $\tilde{\mathbf{z}}_1$ and $\tilde{\mathbf{z}}_2$ are the unique solutions of (20) and (19) respectively and let $\tilde{\mathbf{u}}_1 = C_1\,(\mathbf{e}_1 - \mathcal{G}\,\tilde{\mathbf{z}}_2)_+$ and $\tilde{\mathbf{u}}_2 = C_2\,(\mathbf{e}_2 - \mathcal{H}\,\tilde{\mathbf{z}}_1)_+$ then $\tilde{\mathbf{u}}_1$ and $\tilde{\mathbf{u}}_2$ become solutions of the dual problems (14) and (15) respectively.

The above approach provides an alternative pair of unconstrained optimization problems instead of the pair of constrained optimization problems (14), (15).

Remark 1 For the derivation of the proposed problem formulations (19) and (20), it is required that the inverse of the matrices (H^tH) and (G^tG) of order $(m+1)$ should exist. However, since the matrix (H^tH), similarly (G^tG), is positive semi-definite and therefore its inverse may or may not exist. Following [9], a regularization term $\sigma_H\,I$ may be introduced where $\sigma_H > 0$ is a very small number so that the matrix $(\sigma_H\,I + H^tH)$ becomes positive definite whose inverse can be computed using SMW identity [6], i.e.

$$(\sigma_H\,I + H^tH)^{-1} = \frac{1}{\sigma_H}\,(I - H^t(\sigma_H\,I + HH^t)^{-1}H),$$

having the advantage that it is sufficient to compute $(\sigma_H\,I + H\,H^t)^{-1}$ of order m_2 only. Similarly, introducing a regularization term $\sigma_G\,I$ where $\sigma_G > 0$ is very small, one can compute $(\sigma_G\,I + G^tG)^{-1} = \frac{1}{\sigma_G}\,(I - G^t(\sigma_G\,I + GG^t)^{-1}G)$ in which the matrix $(\sigma_G\,I + GG^t)^{-1}$ is of order m_1 only.

Finally, for solving (19) and (20), it is proposed to obtain their critical points, i.e. finding the roots of the following system of nonlinear equations by using the Newton iterative method

$$\nabla\tilde{L}_1(\mathbf{z}_2) = \mathbf{z}_2 - C_1\,\mathcal{G}^t(\mathbf{e}_1 - \mathcal{G}\,\mathbf{z}_2)_+ = 0 \text{ and}$$
$$\nabla\tilde{L}_2(\mathbf{z}_1) = \mathbf{z}_1 - C_2\mathcal{H}^t(\mathbf{e}_2 - \mathcal{H}\,\mathbf{z}_1)_+ = 0. \tag{21}$$

As each of the above equations of (21) contains the non-differentiable 'plus' function, it is proposed to solve them by using the Newton iterative method with Armijo step size using the generalized Hessian approach of [5] and smoothing technique detailed in [13].

Assume that $k = 1, 2$. Using generalized derivative, a generalized Hessian matrix of $\tilde{L}_k(.)$ can be obtained [8] as

$$\nabla^2\tilde{L}_1(\mathbf{z}_2) = I + C_1\mathcal{G}^t\,diag(\,(\mathbf{e}_1 - \mathcal{G}\,\mathbf{z}_2)_*)\,\mathcal{G} \text{ and}$$
$$\nabla^2\tilde{L}_2(\mathbf{z}_1) = I + C_2\mathcal{H}^t\,diag((\mathbf{e}_2 - \mathcal{H}\,\mathbf{z}_1)_*)\,\mathcal{H}.$$

Clearly $\nabla^2\tilde{L}_k(.)$ is symmetric and positive-definite, and therefore the solution of each system of nonlinear (21) can be computed using fast Newton iterative algorithm with Armijo stepsize whose proof of convergence and finite termination will follow from [14].

The smooth approximation approach is a very popular method [13] used for solving optimization problems, especially QPPs of SVM and SVR, whose objective functions are not twice differentiable. In this work, as another approach, we employ a smoothing technique to make the objective functions $\tilde{L}_k(.)$ sufficiently smooth. More precisely, since the plus function $(x)_+$ appearing in (19) and (20) is not differentiable, it will be replaced by the smooth approximation function $p(x, \alpha)$ with smooth parameter $\alpha > 0$, defined as [13]

$$p(x, \alpha) = x + \frac{1}{\alpha}\log(1 + \exp(-\alpha x)).$$

With the above approximation, the smooth reformulation of (19), for example, will become

$$\min_{\mathbf{z}_2 \in \Re^{m_2}} \frac{1}{2}\mathbf{z}_2^t\mathbf{z}_2 + \frac{C_1}{2}||\;p(\mathbf{e}_1 - \mathcal{G}\,\mathbf{z}_2, \alpha)\;||^2, \tag{22}$$

where $p(\mathbf{e}_1 - \mathcal{G}\,\mathbf{z}_2, \alpha)$ is a vector in $\Re^{m_1}$ whose i-th component can be written as $(\,p(\,\mathbf{e}_1 - \mathcal{G}\,\mathbf{z}_2, \alpha)\,)_i = p(1 - \mathcal{G}_i\,\mathbf{z}_2, \alpha)$ and $\mathcal{G}_i$ is the i-th row of the matrix $\mathcal{G}$.

With the advantage of twice differentiability of the objective function (22), one can apply the Newton-Armijo algorithm for solving it.

Remark 2 The Hessian corresponding to the smooth approximation problem (22) can be computed to be a matrix of order m_2 as

$$I + C_1\,\mathcal{G}^t(\,diag\left(\frac{1}{1 + \exp(\alpha\,(\mathcal{G}\,z_2 - \mathbf{e}_1))}\right)\,)\,\mathcal{G}\,.$$

Following the work of [13] it can be easily shown that the Newton method with Armijo stepsize for (22) converges to its unique solution with quadratic convergence.

The smoothing technique can be extended in a similar manner to the modified dual problem (20) leading to its smooth reformulation

$$\min_{\mathbf{z}_1 \in \Re^{m_1}} \frac{1}{2}\mathbf{z}_1^t \mathbf{z}_1 + \frac{C_2}{2} || \, p(\mathbf{e}_2 - \mathcal{H}\,\mathbf{z}_1, \alpha) \, ||^2$$

whose solution can be obtained by applying the Newton-Armijo algorithm.

Remark 3 For simplicity reasons, we apply the Newton algorithm without Armijo stepsize to solve the pair of problems (19) and (20) numerically by either generalized derivative or smooth approximation approach.

Remark 4 Throughout in this work, solving (21) by Newton method using generalized Hessian and smooth approximation will be denoted by NLWTSVM and SLTWSVM respectively.

4 Experimental results

To analyze the generalization performance and the computational efficiency of our proposed ULTWSVM formulation solved by NLTWSVM and SLTWSVM training algorithms, experiments were performed on 17 bench mark data sets from UCI repository [17] and their results were compared with SVM, LS-TWSVM and TWSVM. All the classifiers were implemented on a PC running on Windows XP OS with 64 bit, 3.20 GHz Intel®core™2 Duo processor having 8 GB of RAM under MATLAB R2008a environment. The standard SVM was solved by MOSEK optimization toolbox for MATLAB available at http://www.mosek.com and, however, no external optimizer was used for solving LS-TWSVM, TWSVM, NLTWSVM and SLTWSVM.

In the implementation of NLTWSVM and SLTWSVM, the values of the termination criteria *tol* and *itmax* were set to 0.001 and 10 respectively. The regularization parameters σ_H, $\sigma_G > 0$ were chosen to be 10^{-5}. Since smooth function approximation with parameter $\alpha = 5$ has shown successful results [13], we assumed $\alpha = 5$ in the implementation of SLTWSVM.

All the datasets were normalized so that each feature value lies in [0, 1]. The optimal parameter values of C_1, C_2 and μ were obtained by performing 10-fold cross validation on the training set by varying their values from the sets $\{10^{-5}, \ldots, 10^5\}$ and $\{2^{-5}, \ldots, 2^5\}$ respectively. With these optimal values, the classification prediction on the test set was computed by dividing the whole dataset randomly into 10 equal parts of which one of them was taken for testing and the remaining parts for training. Finally, the average test accuracy was taken as the measure of prediction.

In Tables 1, 2, 3 and 4 we have shown the accuracy results, along with the optimal parameter values and training time in seconds, by all the classifiers and their averaged ranks on accuracy values for the linear and Gaussian kernels. We notice immediately from Tables 1 and 3 that nonlinear classifiers perform better than their corresponding linear classifiers in terms of accuracy but not in terms of learning time. From Table 1, one can observe that the best performance in terms of accuracy was shown more number of times by SVM than the rest of the classifiers. However, for the case of Gaussian kernel, we observe from Table 3 that NLTWSVM shows the best performance in comparison with the rest of the learning algorithms considered.

To further analyze statistically the performance of the proposed LTWSVM and SLTWSVM classifiers with SVM, LS-TWSVM and TWSVM, as it was suggested in Demsar [4], we perform a non-parametric Friedman test with the corresponding post hoc tests. For this purpose, the average ranks of all the classifiers in terms of prediction accuracy for the linear and Gaussian kernels were computed and listed in Tables 2 and 4 respectively. From the tables we notice that the average ranks of SVM and TWSVM are the least for the linear and Gaussian kernels respectively.

Under the null hypothesis that all the five algorithms on the seventeen data sets considered are equivalent, we compute the Friedman statistics [4] for the linear kernel as shown below

$$\chi_F^2 = \frac{12 \times 17}{5 \times 6}\left[(2.4118^2 + 2.8824^2 + 3.5294^2 + 2.9706^2 + 3.2059^2) - \frac{5 \times 6^2}{4}\right] \approx 4.6512,$$

$$F_F = \frac{16 \times 4.6512}{17 \times 4 - 4.6512} \approx 1.1748,$$

where F_F is distributed according to F-distribution with $(4, 4 \times 16) = (4, 64)$degrees of freedom. The critical value of $F(4, 64)$ for the level of significance $\alpha = 0.05$ is 2.5153. Since the F_F value on RMSE, i.e. 1.1748, is smaller than the critical value 2.5153 for $\alpha = 0.05$, there is no significant difference between the five algorithms. Also, from Table 1 we can observe that LS-TWSVM takes the least training time and it is followed, in general, by ULTWSVM solved using the algorithms NLTWSVM and SLTWSVM.

For the Gaussian kernel, we notice from Table 3 that the number of times the best accuracy obtained by SVM, LS-TWSVM, TWSVM, NLTWSVM and SLTWSVM become 3, 4, 3, 5 and 3 respectively. This gives an indication of the effectiveness of the proposed problem formulation solved by NLTWSVM. To analyze statistically the comparative performance of all the algorithms, the Friedman

Table 1 Performance comparison of our proposed methods NLTWSVM and SLTWSVM with LS-TWSVM, TWSVM and SVM on real world datasets. Linear kernel was employed. Time is for training in seconds

Datasets (Total size)	SVM (C) (time)	LS -TWSVM (C1=C2) (time)	TWSVM (C1=C2) (time)	NLTWSVM (C1=C2) (time)	SLTWSVM (C1=C2) (time)
Australian Credit	85.65 ± 5.22	86.08 ± 5.51	84.78 ± 4.00	86.52 ± 3.68	80.29 ± 3.36
(690×14)	(10^2)	(10^0)	(10^0)	(10^1)	(10^{-1})
	(10.8669)	(0.0792)	(0.9606)	(0.1887)	(0.1202)
Breast-cancer	96.79 ± 2.80	96.23 ± 4.54	95.93 ± 5.19	95.93 ± 5.19	95.93 ± 5.19
(683×9)	(10^0)	(10^1)	(10^0)	(10^0)	(10^0)
	(10.4799)	(0.0761)	(0.7339)	(0.1353)	(0.2786)
Cleveland	83.85 ± 6.98	83.66 ± 7.10	84.52 ± 6.52	84.51 ± 8.06	84.51 ± 8.06
(297×13)	(10^2)	(10^0)	(10^0)	(10^{-1})	(10^{-1})
	(1.9312)	(0.0102)	(0.2190)	(0.0123)	(0.0144)
Haberman	73.47 ± 10.05	75.80 ± 9.15	74.12 ± 8.94	74.45 ± 8.69	74.45 ± 8.69
(306×3)	(10^{-5})	(10^{-1})	(10^{-5})	(10^{-2})	(10^{-2})
	(2.0308)	(0.0108)	(0.2569)	(0.0133)	(0.0132)
Ionosphere	87.48 ± 9.23	87.22 ± 9.27	83.78 ± 13.17	74.94 ± 17.47	80.07 ± 14.35
(351×33)	(10^1)	(10^{-1})	(10^{-2})	(10^{-2})	(10^{-2})
	(2.7552)	(0.0164)	(0.098)	(0.0214)	(0.0181)
Tic-Tac-Toe	65.35 ± 6.71	65.41 ± 6.70	65.35 ± 6.71	68.38 ± 5.96	68.38 ± 5.96
(958×9)	(10^{-5})	(10^1)	(10^0)	(10^0)	(10^0)
	(20.5388)	(0.1685)	(0.4072)	(0.2060)	(0.3903)
Transfusion	76.24 ± 15.54	77.06 ± 15.48	76.24 ± 15.54	77.17 ± 14.5	77.17 ± 14.5
(748×4)	(10^{-5})	(10^0)	(10^0)	(10^0)	(10^0)
	(12.3275)	(0.0878)	(1.6812)	(0.1291)	(0.1703)
Votes	95.86 ± 3	95.90 ± 4.52	95.87 ± 3.52	95.87 ± 3.52	95.87 ± 3.52
(435×16)	(10^0)	(10^{-1})	(10^{-5})	(10^{-5})	(10^{-5})
	(4.1913)	(0.0244)	(0.3925)	(0.0187)	(0.0158)
WDBC	97.71 ± 1.67	96.49 ± 3.09	94.20 ± 5.49	94.56 ± 4.17	94.56 ± 4.17
(569×30)	(10^0)	(10^{-1})	(10^0)	(10^0)	(10^0)
	(7.2651)	(0.0492)	(0.4436)	(0.1007)	(0.2399)
WPBC	79.92 ± 8.37	81.00 ± 8.75	76.37 ± 10.46	74.82 ± 12.86	66.92 ± 13.76
(194×33)	(10^4)	(10^0)	(10^1)	(10^1)	(10^1)
	(0.8419)	(0.0035)	(0.2337)	(0.0124)	(0.0265)
CMC	74.96 ± 4.93	75.06 ± 4.76	74.96 ± 4.93	74.96 ± 4.93	74.96 ± 4.93
(1473×9)	(10^{-5})	(10^0)	(10^0)	(10^0)	(10^3)
	(49.8832)	(0.4853)	(0.9598)	(0.7930)	(0.9491)
German	76.40 ± 5.87	75.60 ± 6.39	72.20 ± 3.61	76.20 ± 6.01	76.20 ± 6.01
(1000×24)	(10^0)	(10^0)	(10^{-1})	(10^0)	(10^0)
	(23.035)	(0.1853)	(0.9073)	(0.2655)	(0.4550)
Heart-statlog	84.07 ± 4.29	83.33 ± 3.59	83.70 ± 3.58	83.70 ± 4.35	83.70 ± 6.1
(270×13)	(10^{-1})	(10^0)	(10^0)	(10^0)	(10^3)
	(1.5945)	(0.0070)	(0.2043)	(0.0172)	(0.0587)
Sonar	77.43 ± 7.06	73.33 ± 12.33	74.02 ± 10.37	75.93 ± 10.38	75.93 ± 10.38
(208×60)	(10^{-1})	(10^0)	(10^0)	(10^0)	(10^0)
	(0.9600)	(0.0041)	(0.1310)	(0.0143)	(0.0291)
Bupa Liver	66.28 ± 11.95	64.57 ± 14.51	62.28 ± 22.44	57.42 ± 6.38	57.42 ± 6.38
(345×6)	(10^3)	(10^0)	(10^2)	(10^{-1})	(10^{-1})
	(1.5513)	(0.0140)	(0.2175)	(0.0277)	(0.4412)

Table 1 (continued)

Datasets (Total size)	SVM (C) (time)	LS -TWSVM (C1=C2) (time)	TWSVM (C1=C2) (time)	NLTWSVM (C1=C2) (time)	SLTWSVM (C1=C2) (time)
Pima Indians	77.53 ± 5.26	77.14 ± 5.51	77.53 ± 6.06	77.27 ± 5.51	77.27 ± 5.51
(768×8)	(10^4)	(10^0)	(10^0)	(10^0)	(10^0)
	(7.9935)	(0.0940)	(1.2209)	(0.2194)	(1.0437)
Splice	84.93 ± 2.76	84.37 ± 2.63	84.40 ± 2.24	84.46 ± 2.52	84.46 ± 2.52
(3175×60)	(10^{-1})	(10^0)	(10^0)	(10^0)	(10^0)
	(169.3800)	(3.6182)	(39.8413)	(19.8909)	(40.2089)

Table 2 Average ranks of SVM, LS-TWSVM, TWSVM, NLTWSVM and SLTWSVM with linear kernel on accuracy values

Datasets	SVM	LS-TWSVM	TWSVM	NLTWSVM	SLTWSVM
Australian Credit	3	2	4	1	5
Breast-cancer	1	2	4	4	4
Cleveland	4	5	1	2.5	2.5
Haberman	5	1	4	2.5	2.5
Ionosphere	1	2	3	5	4
Tic-Tac-Toe	4.5	3	4.5	1.5	1.5
Transfusion	4.5	3	4.5	1.5	1.5
Votes	5	1	3	3	3
WDBC	1	2	5	3.5	3.5
WPBC	2	1	3	4	5
CMC	3.5	1	3.5	3.5	3.5
German	1	4	5	2.5	2.5
Heart-statlog	1	5	3	3	3
Sonar	1	5	4	2.5	2.5
Bupa Liver	1	2	3	4.5	4.5
Pima Indians	1.5	5	1.5	3.5	3.5
Splice	1	5	4	2.5	2.5
Average Rank	2.4118	2.8824	3.5294	2.9706	3.2059

Table 3 Performance comparison of our proposed methods NLTWSVM and SLTWSVM with LS-TWSVM, TWSVM and SVM on real world datasets. Gaussian kernel was employed. Time is for training in seconds

Datasets (Total size)	SVM (C,μ) (time)	LS-TWSVM (C,μ) (time)	TWSVM (C1=C2,μ) (time)	NLTWSVM (C1=C2,μ) (time)	SLTWSVM (C1=C2,μ) (time)
Australian Credit	86.23±6.00	86.08± 5.55	87.25 ± 4.26	87.25± 4.26	87.25 ± 4.26
(690×14)	$(10^{-1}, 2^2)$	$(10^0, 2^3)$	$(10^{-5}, 2^3)$	$(10^{-5}, 2^3)$	$(10^{-5}, 2^3)$
	(12.1111)	(0.6154)	(1.9165)	(0.9894)	(1.0456)
Breast-cancer	97.08 ± 2.55	97.24 ± 2.10	97.37 ± 1.65	97.38 ± 3.33	97.38 ± 3.33
(683×9)	$(10^{-1}, 2^{-1})$	$(10^{-1}, 2^5)$	$(10^0, 2^3)$	$(10^{-3}, 2^1)$	$(10^{-3}, 2^1)$
	(11.9403)	(0.5914)	(1.5393)	(1.0080)	(1.0339)
Cleveland	84.49 ± 8.08	83.66 ± 7.10	84.51 ± 7.10	85.51 ± 7.74	85.51 ± 7.74
(297×13)	$(10^1, 2^3)$	$(10^{-1}, 2^5)$	$(10^0, 2^4)$	$(10^0, 2^5)$	$(10^0, 2^5)$
	(2.1994)	(0.0982)	(0.3680)	(0.1768)	(0.1828)
Haberman	74.49 ± 10.17	76.12 ± 8.76	75.13 ± 7.97	75.78 ± 9.51	75.78 ± 9.51
(306×3)	$(10^3, 2^0)$	$(10^0, 2^1)$	$(10^{-5}, 2^5)$	$(10^0, 2^2)$	$(10^0, 2^2)$
	(2.3345)	(0.1003)	(0.4317)	(0.1812)	(0.1893)

Table 3 (continued)

Datasets (Total size)	SVM (C,μ) (time)	LS-TWSVM (C,μ) (time)	TWSVM (C1=C2,μ) (time)	NLTWSVM (C1=C2,μ) (time)	SLTWSVM (C1=C2,μ) (time)
Ionosphere	92.03 ± 5.97	93.05 ± 4.39	94.87 ± 2.95	94.02 ± 4.14	93.48 ± 5.63
(351×33)	($10^0, 2^0$)	($10^{-5}, 2^{-2}$)	($10^{-1}, 2^2$)	($10^0, 2^2$)	($10^{-4}, 2^{-1}$)
	(3.1302)	(0.1437)	(0.4844)	(0.2767)	(0.2653)
Tic-Tac-Toe	99.16 ± 0.96	98.33 ± 5.27	99.48 ± 1.01	99.58 ± 1.01	99.27 ± 1.40
(958×9)	($10^2, 2^0$)	($10^{-1}, 2^3$)	($10^{-1}, 2^0$)	($10^4, 2^0$)	($10^0, 2^0$)
	(23.5799)	(1.2450)	(2.9997)	(2.8091)	(3.2542)
Transfusion	78.50 ± 13.05	79.46 ± 11.67	79.44 ± 11.99	79.83 ± 11.46	76.64 ± 14.64
(748×4)	($10^0, 2^{-3}$)	($10^0, 2^0$)	($10^{-1}, 2^0$)	($10^0, 2^{-1}$)	($10^1, 2^{-1}$)
	(14.133)	(0.6950)	(1.9290)	(1.2318)	(1.7508)
Votes	96.09 ± 2.85	96.13 ± 3.03	96.56 ± 3.90	96.79 ± 3.07	96.34 ± 3.74
(435×16)	($10^2, 2^4$)	($10^0, 2^2$)	($10^0, 2^4$)	($10^1, 2^5$)	($10^1, 2^5$)
	(4.7922)	(0.2212)	(0.6054)	(0.4049)	(0.5094)
WDBC	98.07 ± 1.29	97.01 ± 2.62	98.24 ± 1.17	98.25 ± 1.43	98.42 ± 1.00
(569×30)	($10^1, 2^{-1}$)	($10^0, 2^0$)	($10^0, 2^1$)	($10^1, 2^1$)	($10^0, 2^1$)
	(8.2767)	(0.4203)	(1.2894)	(0.8300)	(1.1044)
WPBC	82.47 ± 8.95	79.50 ± 10.12	74.79 ± 8.32	78.84 ± 9.31	79.92 ± 8.48
(194×33)	($10^2, 2^1$)	($10^0, 2^2$)	($10^{-2}, 2^0$)	($10^1, 2^4$)	($10^1, 2^3$)
	(0.9514)	(0.0407)	(0.1816)	(0.0753)	(0.0943)
CMC	75.02 ± 4.73	75.27 ± 4.62	75.02 ± 4.89	74.96 ± 4.93	75.09 ± 4.81
(1473×9)	($10^4, 2^2$)	($10^5, 2^{-5}$)	($10^2, 2^1$)	($10^0, 2^4$)	($10^3, 2^2$)
	(58.4523)	(4.9207)	(10.5294)	(5.6026)	(9.3109)
German	76.90 ± 5.82	75.90 ± 4.28	75.70 ± 4.62	76.50 ± 4.77	76.50 ± 4.77
(1000×24)	($10^2, 2^4$)	($10^0, 2^3$)	($10^{-1}, 2^4$)	($10^{-1}, 2^4$)	($10^{-1}, 2^4$)
	(26.1972)	(1.3786)	(3.1892)	(2.3681)	(2.4934)
Heart-statlog	84.07 ± 4.95	84.07 ± 6.06	84.81 ± 5.37	84.44 ± 4.55	84.44 ± 4.55
(270×13)	($10^1, 2^3$)	($10^{-1}, 2^5$)	($10^0, 2^4$)	($10^{-1}, 2^2$)	($10^{-1}, 2^2$)
	(1.8238)	(0.0806)	(0.3211)	(0.1451)	(0.1457)
Sonar	90.40 ± 5.48	90.95 ± 6.12	90.40 ± 5.00	88.02 ± 6.78	90.40 ± 5.00
(208×60)	($10^1, 2^0$)	($10^2, 2^0$)	($10^{-4}, 2^0$)	($10^{-4}, 2^0$)	($10^2, 2^0$)
	(1.1063)	(0.0530)	(0.1706)	(0.1032)	(0.1296)
Bupa Liver	70.57 ± 8.62	68.00 ± 9.21	68.28 ± 8.45	67.42 ± 5.25	67.42 ± 5.25
(345×6)	($10^5, 2^3$)	($10^0, 2^1$)	($10^{-1}, 2^0$)	($10^0, 2^1$)	($10^0, 2^1$)
	(1.7476)	(0.1339)	(0.5223)	(0.1729)	(0.5177)
Pima Indians	77.79 ± 4.87	77.92 ± 6.15	77.66 ± 5.00	77.14 ± 5.44	77.14 ± 5.44
(768×8)	($10^3, 2^3$)	($10^0, 2^1$)	($10^0, 2^3$)	($10^0, 2^1$)	($10^0, 2^1$)
	(8.8799)	(0.7512)	(2.7521)	(1.1933)	(1.8972)
Splice	91.50 ± 1.15	91.57 ± 1.43	92.07 ± 1.44	91.47 ± 1.57	91.47 ± 1.57
(3175×60)	($10^1, 2^1$)	($10^2, 2^1$)	($10^{-5}, 2^1$)	($10^{-1}, 2^1$)	($10^{-1}, 2^1$)
	(178.36)	(25.3465)	(73.3826)	(52.1420)	(57.5960)

statistic under the null hypothesis that all the algorithms are equivalent can be computed as

$$\chi_F^2 = \frac{12 \times 17}{5 \times 6}\left[(3.4706^2 + 3.2059^2 + 2.7353^2 + 2.7647^2 + 2.8235^2) - \frac{5 \times 6^2}{4}\right] \approx 2.8601,$$

$$F_F = \frac{16 \times 2.8601}{17 \times 4 - 2.8601} \approx 0.7025.$$

Since the F_F value on RMSE, i.e. 0.7025, is again smaller than the critical value 2.5153 for $\alpha = 0.05$, there is no significant difference between the five algorithms, i.e. we conclude that none of the methods are statistically better

Table 4 Average ranks of SVM, LS-TWSVM, TWSVM, NLTWSVM and SLTWSVM with Gaussian kernel on accuracy values

Datasets	SVM	LS-TWSVM	TWSVM	NLTWSVM	SLTWSVM
Australian Credit	4	5	2	2	2
Breast-cancer	5	4	3	1.5	1.5
Cleveland	4	5	3	1.5	1.5
Haberman	5	1	4	2.5	2.5
Ionosphere	5	4	1	2	3
Tic-Tac-Toe	4	5	2	1	3
Transfusion	4	2	3	1	5
Votes	5	4	2	1	3
WDBC	4	5	3	2	1
WPBC	1	3	5	4	2
CMC	3.5	1	3.5	5	2
German	1	4	5	2.5	2.5
Heart-statlog	4.5	4.5	1	2.5	2.5
Sonar	3	1	3	5	3
Bupa Liver	1	3	2	4.5	4.5
Pima Indians	2	1	3	4.5	4.5
Splice	3	2	1	4.5	4.5
Average Rank	3.4706	3.2059	2.7353	2.7647	2.8235

than the rest. Finally, regarding the computational learning speed, NLTWSVM and SLTWSVM show faster learning speed than SVM and TWSVM except for one data set. The overall superiority of the proposed novel formulation solved by the two iterative algorithms clearly illustrates its effectiveness and applicability.

5 Conclusion and future work

By reformulating the pair of Lagrangian dual problems of the twin support vector machine, a novel equivalent problem formulation was proposed in this work as a problem of solving a pair of unconstrained minimization problems. Since the objective functions contain the non-smooth 'plus' function, their solutions were obtained by Newton iterative method using the well-known generalized Hessian and smooth approaches. The efficiency of the proposed model in terms of classification accuracy and learning time was demonstrated by performing numerical experiments and comparing their results with SVM, least squares twin SVM and twin SVM. In summary, a simple problem formulation, very simple MATLAB coding and the computational efficiency clearly illustrate the effectiveness and the applicability of our proposed model. The future work will be on the application of semi-smooth approach of [26] for solving this novel problem formulation.

Acknowledgments The authors are thankful to the anonymous reviewers for their comments.

References

1. Balasundaram S, Gupta D (2014) Training Lagrangian twin support vector regression via unconstrained convex minimization. Knowl-Based Syst 59:85–96
2. Cortes C, Vapnik VN (1995) Support vector networks. Mach Learn 20:273–297
3. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel based learning method. Cambridge University Press, Cambridge
4. Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
5. Fung G, Mangasarian OL (2003) Finite Newton method for Lagrangian support vector machine. Neurocomputing 55:39–55
6. Golub GH, Van Loan CF (1996) Matrix computations, 3rd ed., The Johns Hopkins University Press
7. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machine. Mach Learn 46:389–422
8. Hiriart-Urruty J-B, Strodiot JJ, Nguyen VH (1984) Generalized Hessian matrix and second-order optimality conditions for problems with $C^{1,1}$ data. Appl Math Optim 11:43–56
9. Jayadeva KR, Chandra S (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
10. Joachims T, Ndellec C, Rouveriol (1998) Text categorization with support vector machines: learning with many relevant features. In: European conference on machine learning, no.10, Chemnitz, Germany, pp 137–142
11. Kumar MA, Gopal M (2009) Least squares twin support vector machines for pattern classification. Expert Syst Appl 36:7535–7543
12. Kumar MA, Gopal M (2008) Application of smoothing technique on twin support vector machines. Pattern Recogn Lett 29:1842–1848
13. Lee YJ, Mangasarian OL (2001) SSVM: A smooth support vector machine for classification. Comput Optim Appl 20(1):5–22

14. Mangasarian OL (2002) A finite Newton method for classification. Optimization Methods and Software 17:913–929
15. Mangasarian OL, Musicant DR (2001) Lagrangian support vector machines. J Mach Learn Res 1:161–177
16. Mangasarian OL, Wild EW (2006) Multisurface proximal support vector classification via generalized eigenvalues. IEEE Trans Pattern Anal Mach Intell 28(1):69–74
17. Murphy PM, Aha DW (1992) UCI Repository of machine learning databases. University of California, Irvine. http://www.ics.uci.edu/~mlearn
18. Osuna E, Freund R, Girosi F (1997) Training support vector machines: an application to face detection. In: Proceedings of Computer Vision and Pattern Recognition, pp 130–136
19. Platt J (1999) Fast training of support vector machines using sequential minimal optimization. In: Scholkopf B, Burges CJC, Smola AJ (Ed.), Advances in kernel methods- support vector learning, MIT press, Cambridge, MA, pp 185–208
20. Peng X (2011) TPMSVM: A novel twin parametric-margin support vector machine for pattern recognition. Pattern Recogn 44(10-11):2678–2692
21. Peng X (2010) TSVR: An efficient twin support vector machine for regression. Neural Netw 23(3):365–372
22. Rockafellar RT (1974) Conjugate duality and optimization. SIAM, Philadelphia
23. Shao Y, Zhang C, Wang X, Deng N (2011) Improvements on twin support vector machines. IEEE Trans Neural Netw 22(6):962–968
24. Suykens JAK, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Process Lett 9(3):293–300
25. Vapnik VN (2000) The nature of statistical learning theory, 2nd ed. Springer, New York
26. Zhou S, Liu H, Zhou L, Ye F (2007) Semi-smooth Newton support vector machine. Pattern Recogn Lett 28:2054–2062

S. Balasundaram is a Professor of Jawaharlal Nehru University, India. He received his Ph.D. from Indian Institute of Technology, Delhi in 1983. From 1983-85 he was a post doctoral fellow in INRIA, Rocquencourt, France. He joined as an Assistant Professor in Jawaharlal Nehru University in 1986. His main research includes support vector machine and extreme learning machine methods for classification and regression problems, fuzzy regression and applied optimization.

Deepak Gupta is an Assistant Professor of National Institute of Technology, Arunachal Pradesh, India. He received his Ph.D. from Jawaharlal Nehru University, New Delhi in 2015. He also received his Masters of Computer Applications and Master of Technology degree in Computer Science and Technology from Jawaharlal Nehru University in 2009 and 2011 respectively. His research interests include support vector machines, extreme learning machines and other data mining techniques.

Subhash Chandra Prasad is currently pursuing Ph.D. in Computer Science from School of Computer and Systems Sciences, Jawaharlal Nehru University(JNU), New Delhi, India. He received his Master of Technology degree in Computer Science and Technology from JNU, in 2015. His main research includes support vector machine and extreme learning machine methods for classification and regression problems and other data mining techniques.