Neural Processing Letters (2021) 53:1545–1564
https://doi.org/10.1007/s11063-021-10464-3

Generalized Twin Support Vector Machines

H. Moosaei[1] · S. Ketabchi[2] · M. Razzaghi[2] · M. Tanveer[3]

Accepted: 9 February 2021 / Published online: 6 March 2021

Abstract
In this paper, we propose two efficient approaches of twin support vector machines (TWSVM). The first approach is to reformulate the TWSVM formulation by introducing L_1 and L_∞ norms in the objective functions, and convert into linear programming problems termed as LTWSVM for binary classification. The second approach is to solve the primal TWSVM, and convert into completely unconstrained minimization problem. Since the objective function is convex, piecewise quadratic but not twice differentiable, we present an efficient algorithm using the generalized Newton's method termed as GTWSVM. Computational comparisons of the proposed LTWSVM and GTWSVM on synthetic and several real-world benchmark datasets exhibits significantly better performance with remarkably less computational time in comparison to relevant baseline methods.

Keywords Support vector machines · Twin support vector machines · Linear programming · Unconstrained minimization problem · Generalized Newton-Armijo method

Mathematics Subject Classification 00-01 · 99-00

1 Introduction

Support vector machine (SVM) [6,9,47] is one of the most widely used machine learning models for classification problems. The traditional SVM algorithm works by margin maximisation; deriving two unique parallel supporting hyperplanes such that the distance between the samples of two classes is maximized. SVM has been applied in several real-world problems including brain-computer interface [31], face recognition [11], cancer recognition [46],

✉ H. Moosaei
hmoosaei@gmail.com; moosaei@ub.ac.ir

M. Tanveer
mtanveer@iiti.ac.in
http://www.iiti.ac.in/people/∼mtanveer

1 Department of Mathematics, Faculty of Science, University of Bojnord, Bojnord, Iran

2 Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran

3 Department of Mathematics, Indian Institute of Technology Indore, Indore 453 552, India

electroencephalogram signal classification [38], financial time-series forecasting [7], computational biology [16,34], diagnosis of Alzheimer's disease [43] and heart disease detection [2]. The SVMs which introduced by Vapnik and co-workers [9,48], are a class of highly effective machine learning models for pattern classification and regression. SVMs are based on statistical learning theory and have been applied extensively in relation to binary classification and regression problems [5,15,23,45,47,49,50]. 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 so that a convex quadratic programming problem (QPP) should be solved. Various algorithms have been reported to diminish the complexity of SVM including SVMlight [21], sequential minimal optimization (SMO) [37], proximal SVM (PSVM) [28], and Lagrangian SVM (LSVM) [27].

Mangasarian et al. [29] proposed nonparallel hyperplane classifiers called as the generalized eigenvalue proximal support vector machine (GEPSVM). The GEPSVM is a nonparallel plane classifier that generates two hyperplanes as opposed to SVM that generates one hyperplane. Each of the nonparallel hyperplanes which is generated by GEPSVM is close enough to its own class and far from the other class. Motivated by GEPSVM, Jayadeva et al. [20] proposed twin support vector machine and called TWSVM. This method solves two small-sized quadratic programming problems (QPPs) in the dual space instead of one large QPP in traditional SVM. They showed that their algorithm not only has lower computational time, but also performs better in aspect of classification accuracy than conventional SVM. There has been a lot of research in past decade on TWSVM and its applications [12,13,44]. For instance, the twin bounded support vector machine (TBSVM) which was proposed by Shao et al. [41] is an improvement of the TWSVM. The significant advantage of TBSVM over TWSVM is the structural risk minimization which is implemented by adding a regularization term with the purpose of maximizing the margin. This modification which is a general form of the original TWSVM, increases the performance of classification. As another improvement of TWSVM, Peng [36] was suggested ν-TWSVM by introducing two new parameters for determining the trade off between the support vectors and the margin errors. Motivated by ν-TWSVM, Khemchandani et al. [24] proposed two novel binary classifiers termed as improvements on ν-twin support vector machine: Iν-TWSVM and Iν-TWSVM (Fast). The considerable advantage of Iν-TWSVM over ν-TWSVM was that Iν-TWSVM solved one smaller-sized QPP and one Unconstrained minimization problem (UMP) instead of solving two quadratic programming problems.

In this paper, we propose two efficient approaches of TWSVM. The first approach is to reformulate the TWSVM formulation by introducing L_1 and L_∞ norms in the objective functions, and convert into linear programming problems termed as LTWSVM for binary classification. The second approach is solving the primal TWSVM, and convert into completely unconstrained minimization problem. Since the objective function is convex, piecewise quadratic and only once differentiable, not twice, we present an efficient algorithm using the generalized Newton's method termed as GTWSVM. Computational comparisons of our proposed methods with four existing methods in terms of classification accuracy and learning time have been made on several artificial, UCI, NDC, and clinical datasets. The numerical experiments indicate that our proposed methods outperform the other methods.

The rest the paper is organized as follows. Section 2 gives the formulations of SVM and TWSVM. Section 3 discusses the details of our proposed formulations. In Sect. 4, numerical experiments are conducted and a comparison of their results with other standard methods is drawn. Finally, conclusions are presented in Sect. 5.

Notation. We denote the n-dimensional real space by R^n, and by $A^\top$ we mean the transpose of a matrix A. Next, a_+ replaces negative components of a vector a by zeros. If f is a real valued function defined on the n-dimensional real space R^n, the gradient of f at x is denoted by $\bigtriangledown f(x)$, which is a column vector in R^n and the $n \times n$ Hessian matrix of the second partial derivatives of f at x is denoted by $\bigtriangledown^2 f(x)$. A column vector of one of arbitrary dimensions is indicated by e. For $x \in R^n$, and $1 \leq p < \infty$, we defined the p-norm and the ∞-norm by $\|x\|_p = (\sum_{j=1}^{n} |x_j|^p)^{\frac{1}{p}}$ and $\|x\|_\infty = max_{1\leq j\leq n}|x_j|$ respectively. Matrices A and B denote classes $+1$ and -1, respectively. For $A \in R^{m\times n}$ and $C \in R^{n\times l}$, a kernel $K(A, C)$ is a function that maps $R^{m\times n} \times R^{n\times l}$ into $R^{m\times l}$. The convex hull of a set S is denoted by $co\{S\}$.

2 Background

Here, a short description of the standard SVM and TWSVM formulations are provided. For further descriptions, please refer to [6,20,47].

2.1 Support Vector Machine for Classification

Consider a set of input examples $\{(x_i, y_i)\}$, $(i = 1, 2, \ldots, m)$, where $x_i \in R^n$ are the inputs and $y_i \in \{-1, +1\}$ are the corresponding outputs labels. By including the regularization term $\frac{1}{2}\|w\|^2$ and error variables ξ_i, the optimization problem can be written as [6,47]:

$$\begin{aligned} \min_{(w,b,\xi)} \quad & \frac{1}{2}w^T w + c\sum_{i=1}^{m}\xi_i \\ s.t. \quad & y_i(w^\top x_i + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \quad i = 1, \ldots, m, \end{aligned} \tag{1}$$

where $c > 0$ is the regularization parameter and ξ_i measures the violation of constraint for each x_i. It should be noted that minimizing term $\frac{1}{2}w^T w$ is equal to maximizing the margin between two parallel supporting hyperplanes $w^T x - b = +1$ and $w^T x - b = -1$. Figure 1 illustrates the SVM method graphically.

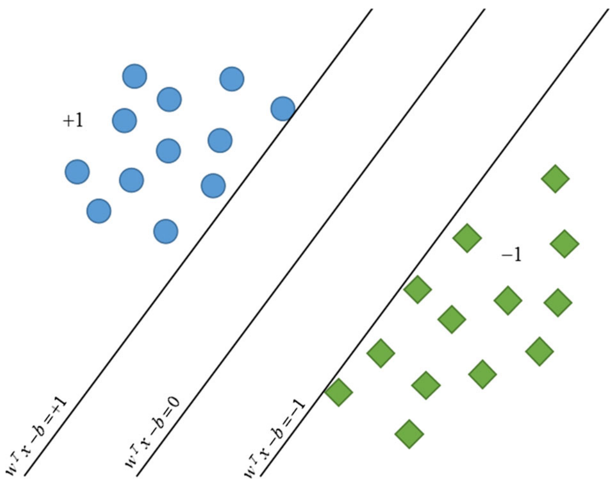

Fig. 1 Geometric representation of SVM method

SVM solves its dual problem instead of primal problem. The Lagrangian dual problem can be derived as follows:

$$
\begin{aligned}
\min_{\alpha} \quad & \sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j x_i x_j - \sum_{i=1}^{m}\alpha_i \\
s.t. \quad & \sum_{i=1}^{m} y_i\alpha_i = 0, \\
& 0 \le \alpha_i \le c, \quad i = 1, \cdots, m,
\end{aligned} \tag{2}
$$

which is convex QPP. The principle of structural risk minimization is carried out in standard SVM. At the same time, the confidential interval term $||w||^2$ and the empirical risk term ξ_i are minimized. For more detail see [10,47].

2.2 Twin Support Vector Machines

Jayadeva et al. [20] presented an efficient nonparallel SVM algorithm called TWSVM for binary classification. Note that TWSVM and standard SVM have the similar formulation [6,47] except that in TWSVM, all the data points do not emerge in the constraints of either problem simultaneously. Furthermore, TWSVM is faster than SVM as it solves two smaller sized QPPs [20].

Suppose that all data points in class +1 are depicted by a matrix $A \in R^{m_1 \times n}$, where the i^{th}row $A_i \in R^n$ and the matrix $B \in R^{m_2 \times n}$ indicate the data points of class -1. The linear TWSVM, unlike SVM, finds a pair of nonparallel hyperplanes as follows:

$$
w_1^T x + b_1 = 0, \quad \text{and} \quad w_2^T x + b_2 = 0, \tag{3}
$$

where $w_1 \in R^n$, $w_2 \in R^n$, $b_1 \in R$ and $b_2 \in R$. Each hyperplane of TWSVM is as close as possible to data points of in one of the two classes and as far as possible from data points of other class (see Fig. 2). Therefore, the formulation of TWSVM can be stated as follows:

$$
\begin{aligned}
\min_{(w_1,b_1,q_1)\in R^{2n+1}} \quad & \|Aw_1 + e_1b_1\|^2 + c_1e_2^T q_1, \\
s.t. \quad & -(Bw_1 + e_2b_1) + q_1 \ge e_2, \\
& q_1 \ge 0,
\end{aligned} \tag{4}
$$

$$
\begin{aligned}
\min_{(w_2,b_2,q_2)\in R^{2n+1}} \quad & \|Bw_2 + e_2b_2\|^2 + c_2e_1^T q_2, \\
s.t. \quad & (Aw_2 + e_1b_2) + q_2 \ge e_1, \\
& q_2 \ge 0,
\end{aligned} \tag{5}
$$

where $c_1, c_2 > 0$ are parameters, e_1, e_2 are vectors of ones of appropriate dimensions, and q_1 and q_2 are slack vectors. It is clear that the purpose of TWSVM is solving two QPPs (4) and (5). Each QPP in the TWSVM pair represents a classic SVM formulation, with the exception that not all data points show up in the constraints of either problem [20].

The corresponding dual problems can be obtained respectively as follows:

$$
\begin{aligned}
\max_{\alpha} \quad & e_2^T\alpha - \frac{1}{2}\alpha^T G(H^T H)^{-1} G^T \alpha, \\
s.t. \quad & 0 \le \alpha \le c_1,
\end{aligned} \tag{6}
$$

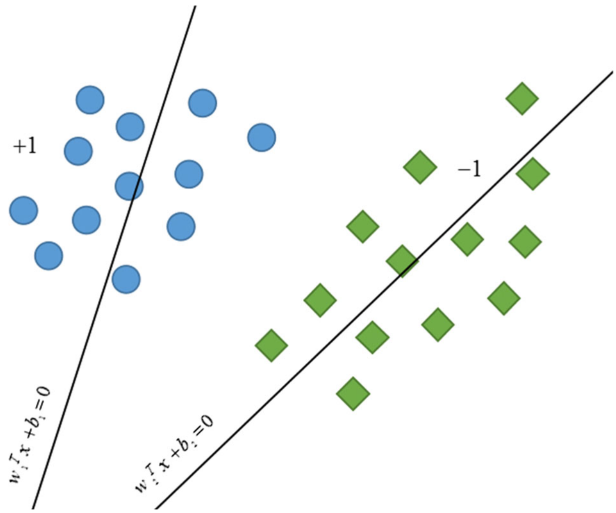

Fig. 2 Geometric interpretation of standard TWSVM

$$\begin{aligned} \max_{\gamma} \quad & e_1^T \gamma - \frac{1}{2}\gamma^T H(G^T G)^{-1} H^T \gamma, \\ s.t. \quad & 0 \le \gamma \le c_2, \end{aligned} \tag{7}$$

where α and γ are Lagrange multipliers, $H = [A\ e_1]$ and $G = [B\ e_2]$.

It can be shown that the two hyperplanes would be find from the solution of (6) and (7) by $[w_1\ \ b_1]^T = -(H^T H)^{-1} G^T \alpha$ and $[w_2\ \ b_2]^T = (G^T G)^{-1} H^T \gamma$.

Although the matrices $G^T G$ or $H^T H$ are positive semidefinite, they may be singular and ill-conditioned, we introduce a regularization term and the inverse matrices $(G^T G)^{-1}$and $(H^T H)^{-1}$ are approximately replaced by $(G^T G + \delta I)^{-1}$and $(H^T H + \delta I)^{-1}$ respectively, where δ and I are a very small positive scalar and identity matrix, respectively.

The nonlinear TWSVM can be expressed as follows:

$$\begin{aligned} \min_{(w_1, b_1, q_1) \in R^{2n+1}} \quad & \|K(A, C^T) w_1 + e_1 b_1\|^2 + c_1 e_2^T q_1, \\ s.t. \quad & -(K(B, C^T) w_1 + e_2 b_1) + q_1 \ge e_2, \\ & q_1 \ge 0. \end{aligned} \tag{8}$$

$$\begin{aligned} \min_{(w_2, b_2, q_2) \in R^{2n+1}} \quad & \|K(B, C^T) B w_2 + e_2 b_2\|^2 + c_2 e_1^T q_2, \\ s.t. \quad & (K(A, C^T) w_2 + e_1 b_2) + q_2 \ge e_1, \\ & q_2 \ge 0, \end{aligned} \tag{9}$$

where $c_1, c_2 > 0$ are parameters, e_1, e_2 are vectors of ones of appropriate dimensions, q_1 and q_2 are slack vectors, $K(\cdot, \cdot)$ is an arbitrary kernel function and $C = [A; B]$.

The dual problems for (8) and (9) are shown in [20] to obtain the following hypersurfaces:

$$K(x^T, C^T) w_1 + b_1 = 0, \quad \text{and} \quad K(x^T, C^T) w_2 + b_2 = 0. \tag{10}$$

Jayadeva et al. [20] showed that the classification performance of TWSVM significantly outperforms that the conventional SVM and GEPSVM on UCI machine learning datasets.

3 Proposed Formulations

In this section, two new approaches are proposed for non-parallel support vector machines for classification problems.

3.1 Linear Programming for Twin Support Vector Machines

In this subsection, a new approach to classification problems have been presented by introducing L_1 and L_∞ norms in the objective functions of TWSVM formulation, and convert into linear programming problems termed as LTWSVM.

3.1.1 Linear Case

For linear case, by replacing L_2 norm with L_1 norm in (4) and (5), the 1-norm TWSVM reformulation can be written as follows:

$$\begin{aligned} \min_{(w_1,b_1,q_1)\in R^{2n+1}} \quad & \|Aw_1 + e_1 b_1\|_1 + c_1 e_2^T q_1, \\ s.t. \quad & -(Bw_1 + e_2 b_1) + q_1 \geq e_2, \\ & q_1 \geq 0, \end{aligned} \tag{11}$$

$$\begin{aligned} \min_{(w_2,b_2,q_2)\in R^{2n+1}} \quad & \|Bw_2 + e_2 b_2\|_1 + c_2 e_1^T q_2, \\ s.t. \quad & (Aw_2 + e_1 b_2) + q_2 \geq e_1, \\ & q_2 \geq 0. \end{aligned} \tag{12}$$

The first term of each objective function $\|Aw_1 + e_1 b_1\|_1$ and $\|Bw_2 + e_2 b_2\|_1$ is to find the hyperplane that are closest to each class. Note that the terms $\|Aw_1 + e_1 b_1\|_1$ and $\|Bw_2 + e_2 b_2\|_1$ are easily converted to linear terms $e_2^T t$ and $e_1^T t$ with the added constraints $-t_1 \leq Aw_1 + e_1 b_1 \leq t_1$ and $-t_2 \leq Bw_2 + e_2 b_2 \leq t_2$ respectively.

One can obtain the following linear programming problems:

$$\begin{aligned} \min_{(w_1,b_1,q_1,t_1)\in R^{3n+1}} \quad & e_2^T t_1 + c_1 e_2^T q_1, \\ s.t. \quad & Aw_1 + e_1 b_1 \leq t_1, \\ & -Aw_1 - e_1 b_1 \leq t_1, \\ & -(Bw_1 + e_2 b_1) + q_1 \geq e_2, \\ & q_1 \geq 0, t_1 \geq 0. \end{aligned} \tag{13}$$

$$\begin{aligned} \min_{(w_2,b_2,q_2,t_2)\in R^{3n+1}} \quad & e_1^T t_2 + c_2 e_1^T q_2, \\ s.t. \quad & Bw_2 + e_2 b_2 \leq t_2, \\ & -Bw_2 - e_2 b_2 \leq t_2, \\ & (Aw_2 + e_1 b_2) + q_2 \geq e_1, \\ & q_2 \geq 0, t_2 \geq 0. \end{aligned} \tag{14}$$

By replacing L_2 norm with L_∞ norm in (4) and (5), the ∞-norm TWSVM reformulation can be written as follows:

$$\begin{aligned}
\min_{(w_1,b_1,q_1)\in R^{2n+1}} \quad & \|Aw_1+e_1b_1\|_\infty + c_1e_2^Tq_1, \\
s.t. \quad & -(Bw_1+e_2b_1)+q_1 \geq e_2, \\
& q_1 \geq 0. \qquad (15)\\
\min_{(w_2,b_2,q_2)\in R^{2n+1}} \quad & \|Bw_2+e_2b_2\|_\infty + c_2e_1^Tq_2, \\
s.t. \quad & (Aw_2+e_1b_2)+q_2 \geq e_1, \\
& q_2 \geq 0. \qquad (16)
\end{aligned}$$

The solution of problems (15) and (16) is similar to the solution of the following linear programming problems:

$$\begin{aligned}
\min_{(w_1,b_1,q_1,t_1)\in R^{2n+2}} \quad & t_1 + c_1e_2^Tq_1, \\
s.t. \quad & -t_1e \leq Aw_1+e_1b_1 \leq t_1e \\
& -(Bw_1+e_2b_1)+q_1 \geq e_2, \\
& q_1 \geq 0, t_1 \geq 0. \qquad (17)\\
\min_{(w_2,b_2,q_2,t_2)\in R^{2n+2}} \quad & t_2 + c_2e_1^Tq_2, \\
s.t. \quad & -t_2e \leq Bw_2+e_2b_2 \leq t_2e \\
& (Aw_2+e_1b_2)+q_2 \geq e_1, \\
& q_2 \geq 0, t_2 \geq 0. \qquad (18)
\end{aligned}$$

3.1.2 Nonlinear Case

For nonlinear version of as LTWSVM, we replace L_2 norm with L_1 norm in (8) and (9), and have the following modeles:

$$\begin{aligned}
\min_{w_1,b_1,q_1} \quad & \|K(A,C^T)w_1+e_1b_1\|_1 + c_1e_2^Tq_1, \\
s.t. \quad & -(K(B,C^T)w_1+e_2b_1)+q_1 \geq e_2, \\
& q_1 \geq 0, \qquad (19)\\
\min_{w_2,b_2,q_2} \quad & \|K(B,C^T)w_2+e_2b_2\|_1 + c_2e_1^Tq_2, \\
s.t. \quad & (K(A,C^T)w_2+e_1b_2)+q_2 \geq e_1, \\
& q_2 \geq 0. \qquad (20)
\end{aligned}$$

Similar to the linear case, the terms $\|K(A,C^T)w_1+e_1b_1\|_1$ and $\|K(B,C^T)w_2+e_2b_2\|_1$ can be converted to linear terms e_2^Tt and e_1^Tt with the added constraints $-t_1 \leq K(A,C^T)w_1+e_1b_1 \leq t_1$ and $-t_2 \leq K(B,C^T)w_2+e_2b_2 \leq t_2$ respectively. So we can drive the following linear programming problems:

$$\begin{aligned}
\min_{w_1,b_1,q_1,t_1} \quad & e_2^Tt_1 + c_1e_2^Tq_1, \\
s.t. \quad & K(A,C^T)w_1+e_1b_1 \leq t_1, \\
& -K(A,C^T)w_1-e_1b_1 \leq t_1,
\end{aligned}$$

$$
\begin{aligned}
& -(K(B, C^T)w_1 + e_2 b_1) + q_1 \geq e_2, \\
& q_1 \geq 0, t_1 \geq 0. \qquad (21) \\
\min_{w_2, b_2, q_2, t_2} \quad & e_1^T t_2 + c_2 e_1^T q_2, \\
s.t. \quad & K(B, C^T)w_2 + e_2 b_2 \leq t_2, \\
& -K(B, C^T)w_2 - e_2 b_2 \leq t_2, \\
& (K(A, C^T)w_2 + e_1 b_2) + q_2 \geq e_1, \\
& q_2 \geq 0, t_2 \geq 0. \qquad (22)
\end{aligned}
$$

Now we replace L_2 norm with L_∞ norm in (8) and (9), the ∞-norm reformulations of TWSVM can be described as follows:

$$
\begin{aligned}
\min_{w_1, b_1, q_1} \quad & \|K(B, C^T)w_1 + e_1 b_1\|_\infty + c_1 e_2^T q_1, \\
s.t. \quad & -(K(B, C^T)w_1 + e_2 b_1) + q_1 \geq e_2, \\
& q_1 \geq 0. \qquad (23) \\
\min_{w_2, b_2, q_2} \quad & \|K(B, C^T)w_2 + e_2 b_2\|_\infty + c_2 e_1^T q_2, \\
s.t. \quad & (K(A, C^T)w_2 + e_1 b_2) + q_2 \geq e_1, \\
& q_2 \geq 0. \qquad (24)
\end{aligned}
$$

The solution of problems (23) and (24) is similar to the solution of the following linear programming problems:

$$
\begin{aligned}
\min_{w_1, b_1, q_1, t_1} \quad & t_1 + c_1 e_2^T q_1, \\
s.t. \quad & -t_1 e \leq K(A, C^T)w_1 + e_1 b_1 \leq t_1 e \\
& -(K(B, C^T)w_1 + e_2 b_1) + q_1 \geq e_2, \\
& q_1 \geq 0, t_1 \geq 0. \qquad (25) \\
\min_{w_2, b_2, q_2, t_2} \quad & t_2 + c_2 e_1^T q_2, \\
s.t. \quad & -t_2 e \leq K(B, C^T)w_2 + e_2 b_2 \leq t_2 e \\
& (K(A, C^T)w_2 + e_1 b_2) + q_2 \geq e_1, \\
& q_2 \geq 0, t_2 \geq 0. \qquad (26)
\end{aligned}
$$

Then we have the following Proposition.

Proposition 1 *Finding two nonparallel hyperplanes for linear and nonlinear binary classification using L_1-norm and L_∞-norm problems is equal to solving a linear programming problem.*

Remark 1 Solving linear programming problems related to L_1 norm and L_∞ norm led to obtain the almost same accuracy and time. Hence, we call both methods LTWSVM.

Therefore, we obtain some modifications of TWSVM that led to linear programming problems (13), (14), (17), (18), (21), (21), (25), and (26) which we call LTWSVM. The advantage of our LTWSVM over TWSVM is clear due to solving a linear programming problems rather than a quadratic programming problem (QPP).

3.2 Twin Support Vector Machines via Unconstrained Convex Minimization

In this subsection, a new approach for solving TWSVM in primal space is proposed and termed as GTWSVM. The proposed method creates two unconstrained minimization problems (UMPs) having their objective functions are not twice differentiable, and thus Newton-Armijo type algorithm is proposed to solve the UMPs.

Let us first modify the model of TWSVM (4). To do this, primal problem (4) of TWSVM changes to problem (27) which uses the square of 2-norm of slack variable q_1 instead of 1-norm q_1 with the same weight.

$$\begin{aligned} \min_{(w_1,b_1,q_1)\in R^{2n+1}} \quad & \|Aw_1 + e_1b_1\|^2 + c_1 q_1^T q_1, \\ s.t. \quad & -(Bw_1 + e_2b_1) + q_1 \geq e_2. \\ & q_1 \geq 0. \end{aligned} \tag{27}$$

From the constraints of the problem (27), we know that

$$\begin{aligned} & q_1 \geq e_2 + Bw_1 + e_2b_1, \\ & q_1 \geq 0. \end{aligned} \tag{28}$$

According to the above relations , any optimal solution of the problem (27) must satisfies in:

$$q_1 = (e_2 + Bw_1 + e_2b_1)_+.$$

Therefore we replace q_1 in (27) with $(e_2 + Bw_1 + e_2b_1)_+$ and convert the QPP (27) into an equivalent UMP as follows:

$$\min_{(w_1,b_1)\in R^{n+1}} \|Aw_1 + e_1b_1\|^2 + c_1\|(e_2 + Bw_1 + e_2b_1)_+\|^2. \tag{29}$$

Similarly, one can obtain the following modified primal problem (5) and can also be reformulated as an unconstrained minimization problem given by (31):

$$\begin{aligned} & \min_{(w_2,b_2)\in R^{n+1}} \|Bw_2 + e_2b_2\|^2 + c_2 q_2^T q_2, \\ s.t. \quad & (Aw_2 + e_1b_2) + q_2 \geq e_1, \\ & q_2 \geq 0, \end{aligned} \tag{30}$$

$$\min_{(w_2,b_2)\in R^{n+1}} \|Bw_2 + e_2b_2\|^2 + c_2\|(e_1 - Aw_2 - e_1b_2)_+\|^2. \tag{31}$$

We can also extend this method to solve nonlinear TWSVM. The unconstrained minimization problem is comparable to the two primal nonlinear TWSVMs (29) and (31) are given respectively as follows:

$$\min_{(w_1,b_1)\in R^{n+1}} \|K(A, C^T)w_1 + e_1b_1\|^2 + c_1\|(e_2 + K(B, C^T)w_1 + e_2b_1)_+\|^2. \tag{32}$$

$$\min_{(w_2,b_2)\in R^{n+1}} \|K(B, C^T)w_2 + e_2b_2\|^2 + c_2\|(e_1 - K(A, C^T)w_2 - e_1b_2)_+\|^2. \tag{33}$$

Since the objective functions of problems (29), (31), (32) and (33) are piecewise, quadratic, convex and only once differentiable, the generalized Newton's method can be applied to solve these UMPs.

We first introduce some properties before solving these problems. The class LC^1 of functions is defined as follows [18]:

Definition 1 A continuously differentiable function $f : R^n \rightarrow R$ is said to be an LC^1 function on an open set A whenever ∇f is locally Lipschitz on A.

From the above definition, Jacobian in Clarke's sense [8] could be defined as follows:

$$\partial^2 f(x) := co\{H \in R^{n\times n} : \exists x^k \rightarrow x \quad with \quad \nabla f \text{ differentiable at } x^k \text{ and } \nabla f^2(x^k) \rightarrow H\},$$

which is the generalized Hessian of f at x.
Now we consider the problem (29), and others (31)–(33) are the same.
The objective function of problem (29) can be written as follows:
$g(u) = \|Mu\|^2 + c_1\|(e_2 - Su)_+\|^2$, where $M = [A \;\; e_1]$, $S = [B \;\; e_2]$, and $u = [w_1, b_1]$.

Theorem 1 *$\nabla g(u)$ is said to be globally Lipschitz continuous and the generalized Hessian of $g(u)$ is*

$$\partial^2 g(u) = 2M^T M + 2c_1 S^T D(z) S,$$

where $D(z)$ indicates the diagonal matrix whose ith diagonal entry z_i is equivalent to 1 if $(e_2 - Su)_i > 0$; z_i is equivalent to 0 if $(e_2 - Su)_i \leq 0$.

The proof of the above theorem are derived in [18].

Since the generalized Hessian matrix can be singular, the following modified Newton's direction is used [4,22,35].

$$-(\partial^2 g(u) + \delta I)^{-1}\nabla g(u),$$

where δ is a small positive number (in our numerical experiments we used $\delta = 10^{-4}$), and I is the identity matrix of appropriate order. Now, we have an iterative process as follows:

$$p_{k+1} = p_k - \lambda_k(\partial^2 g(p_k) + \delta I)^{-1}\nabla g(p_k). \tag{34}$$

The starting vector is $p_0 = 0$. Also, the stopping criterion of this method was as follows: (in our computations, $tol = 10^{-10}$)

$$||p_{k+1} - p_k|| \leq tol.$$

For finding the step size (λ_k), we suggest Armijo rule [3], so we can derive the global finite-step convergence of the modified Newton's method starting from any point. The process of our method is described in Algorithm 1.

Algorithm 1

Require: $\alpha > 1, \tau > 1, \epsilon > 0$ be error tolerance and δ is a small positive number.
Ensure: Choose a $p_0 \in R^n$ and set $k = 0$. $s > 0$ be a constant, $\sigma \in (0, 1)$ and $\mu \in (0, 1)$.
while $\|\nabla g(p_k, \alpha)\|_\infty \geq \epsilon$ **do**
$d_k = -(\partial^2 g(p_k, \alpha) + \delta I)^{-1}\nabla g(p_k, \alpha)$,
$p_{k+1} = p_k + \lambda_k d_k$.
Choose $\lambda_k = max\{s, s\sigma, s\sigma^2, ...\}$ *such that:*
$g(p_k, \alpha) - g(p_k + \lambda_k d_k, \alpha) \geq -\lambda_i \mu \nabla(g(p_k, \alpha))^T d_k$,
$\alpha = \tau\alpha$.
$k = k + 1$.
end while

The finite-step global convergence of this algorithm is proved in [25–27,35].

Theoretically, the main advantage of GTWSVM over TWSVM is that it solves an unconstrained convex problem instead of solving a pair of QPPs.

4 Numerical Experiments

To show the effectiveness of our proposed methods LTWSVM and GTWSVM, numerical experiments were conducted on two artificial datasets, 11 real-world benchmark datasets taken from UCI machine learning repository [14], David Musicant's NDC Data Generator datasets [33], and two clinical datasets. All methods were carried out by MATLAB 2014 running on a PC Intel Pentium Dual Core 2.80 GHz CPU with 4 GB of RAM.

To solve QPPs to achieve the optimal solution of the dual problems, "quadprog.m" function is used. The LPs in LTWSVM were solved by using the " linprog.m" function in Matlab. In our simulations, RBF kernel ($k(x_i, x_j) = exp(\frac{||x_i - x_j||}{\mu})$) was considered as it is commonly utilized and demonstrates excellent generalization performance. The best results are marked in bold. The standard 10-fold crossvalidation [10] was used to estimate the generalized accuracy.

4.1 Baselines

In this subsection, we briefly introduce the baselines which are used into our experiments.

- TWSVM [20]: The method obtains two non-parallel hyperplanes by solving two dual QPPs.
- USVM-RFE [39] Motivated by the work on support vector machine based recursive feature elimination (SVM-RFE) [17], this method proposed a universum based technique for feature selection. This provides an improvement over SVM-RFE algorithm by giving prior information about data.
- Iν-TBSVM [24]: This method is motivated by ν-twin support vector machine (ν-TSVM). The significant advantage of Iν-TBSVM over ν-TSVM is that Iν-TBSVM avoids of computing the matrix inverse when solving the dual problems.

4.2 Parameter Selection

We conduct experiments on housing and wdbc datasets to explore the impact of the parameters c_1, c_2 and γ on the generalization performance of our GTWSVM and LTWSVM. The values of the parameters c_1 and c_2 are set as $c_1 = c_2$. One observes from Figs. 3 and 4 that the parameters c_1 and γ have larger effect on the generalization performance.

Therefore,the performances of all algorithms depend largely on the choices of the parameter values. The optimal values for the parameters were determined by the grid search method. In the Iν-TBSVM ,TWSVM, GTWSVM and LTWSVM, we set $c_1 = c_3$, $c_2 = c_4$, and in USVM-RFE $c = c_u$ to reduce the computational time and selected optimal values for $c_1, c_2, c_3, c_4, c, c_u$ from the range of $\{\alpha \times 10^i | i = -10, -9, \ldots, 9, 10\}$ $(\alpha \in (1, 9))$ and the Gaussian kernel parameter μ was chosen from similar range. The parameter ν in Iν-TBSVM is selected from the set $\{0.1, 0.2, \ldots, 0.8, 0.9\}$.

4.3 Artificial Dataset

The first example is generated data randomly in two classes of A and B which are linearly separated from each other. In the first example, 500 points for class A and 200 points for class B are created, and the data are generated randomly in the interval $[-50, 50]$. Figure 5a–c describe the performance of the TWSVM and our LTWSVM, GTWSVM on this example

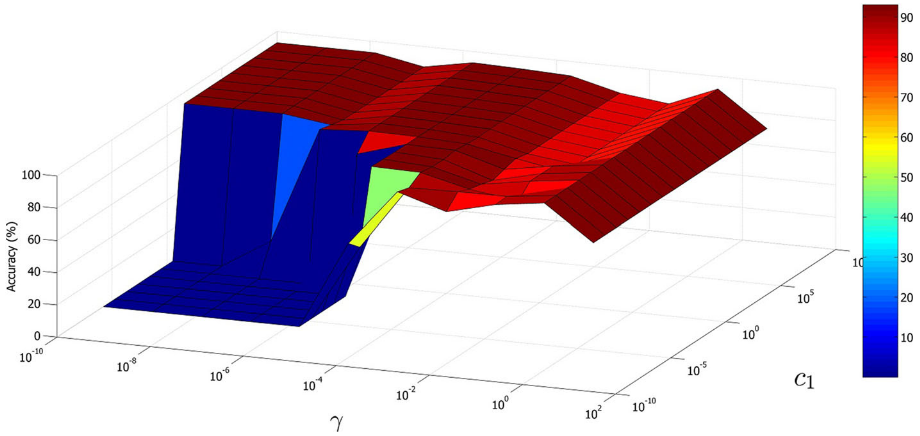

Fig. 3 Effect of the parameters on the generalization performance for housing dataset

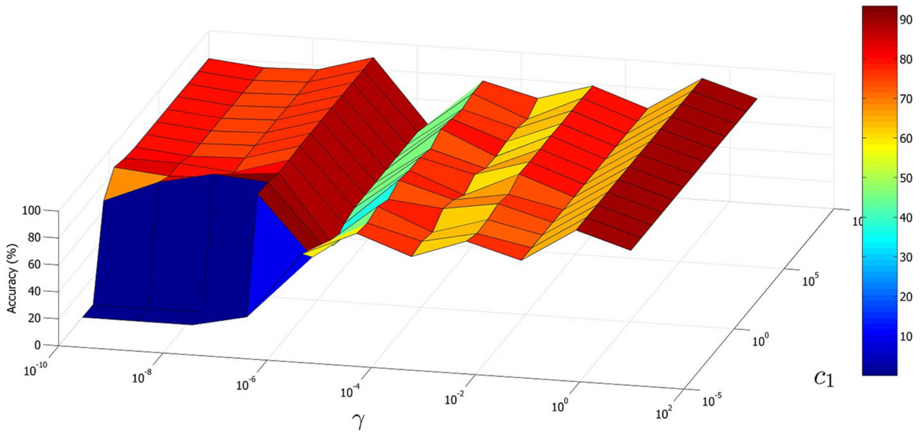

Fig. 4 Effect of the parameters on the generalization performance for WDBC dataset

in which the samples A and B are denoted as purple and blue colors, respectively. As can be seen, the accuracy of TWSVM, LTWSVM and GTWSVM are 94.9 (1375 seconds), 94.9 (260 seconds) and 95.3 (24 seconds) respectively. These results show that GTWSVM obtains a pair of more suitable hyperplanes compared with the other two classifiers and after it, although the accuracies of TWSVM and LTWSVM are the same, however, our LTWSVM is faster than TWSVM.

The second example shows a Ripley's synthetic dataset that is artificially generated [40]. It is a two-dimensional dataset of 250 data samples, of which one half of the samples 125 are assigned to class of A and the other half 125 to class B, and they are not separated linearly. In Fig. 6a–c, the results of linear TWSVM, LTWSVM and GTWSVM on this dataset are shown. It can be seen that our GTWSVM is better for this problem and obtains the better classification results than the LTWSVM and TWSVM. Although the LTWSVM and TWSVM have the same accuracy, the LTWSVM is faster than TWSVM. Figure 6a–c describe the relations between the parameter values and accuracy. The hyperplanes in Fig. 6a–c are shown by black and green colors.

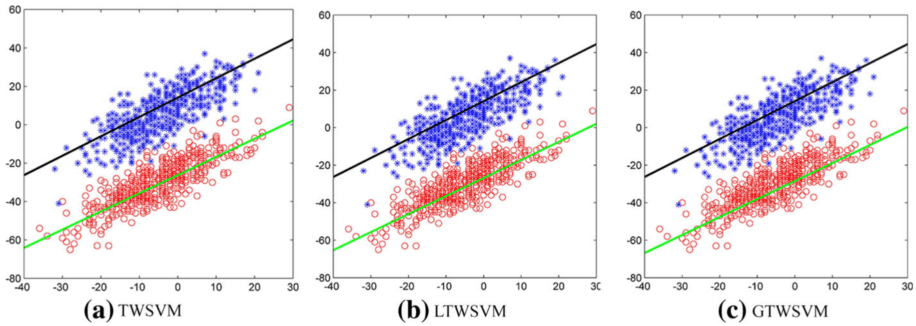

Fig. 5 Classification results of TWSVM, LTWSVM and GTWSVM on generated dataset

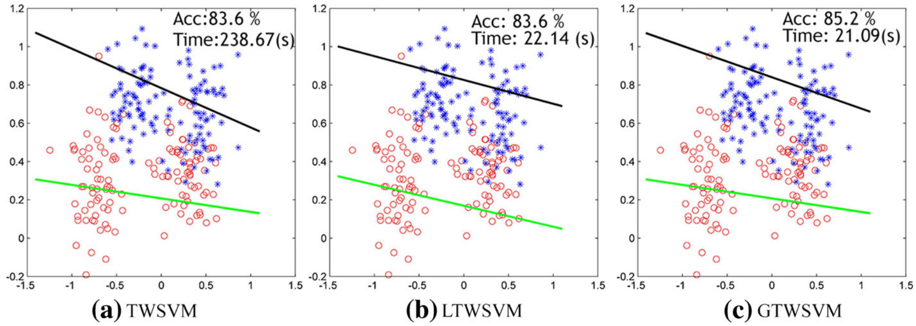

Fig. 6 Classification results of linear methods on Ripley's dataset

4.4 UCI Datasets

For further assessment, we performed numerical experiments on eleven benchmark datasets including Australian, Diabet, German, Sonar, Haberman, Heart, Housevotes, Ionosphere, Spect, Splice and Wdbc taken from the UCI machine learning repository [14] and reported the outcomes of the USVM-RFE, Iν-TBSVM, TWSVM, LTWSVM, and GTWSVM methods on the selected datasets in Tables 1 and 2 in terms of optimal parameters, accuracies and CPU time.

As shown in Table 1, the propose to GTWSVM and LTWSVM have superior generalization performance compared TWSVM, USVM-RFE, and Iν-TBSVM on most of the datasets considered. Further, we drew a comparison between the training time for these five algorithms. As Table 1 suggests, our proposed GTWSVM need lower training time than the other methods. The mean accuracy of all the datasets for all the five algorithms have shown in Fig. 7. We see that GTWSVM has the highest average classification accuracy followed by LTWSVM, TWSVM, USVM-RFE, and Iν-TBSVM. Similarly, Fig. 8 shows the average of training time of all datasets for all the five algorithms, one can see that the rank of training speed of GTWSVM, TWSVM, Iν-TBSVM, LTWSVM, and USVM-RFE is one, two, three, four, and five respectively. To improve the training time of LTWSVM, we can use of CPLEX for solving the linear programming problem.

To further examine the effectiveness of the proposed algorithms, we conduct experiments nonlinearly on five benchmark UCI datasets. One can see the conclusion on nonlinear datasets in Table 2 similar to linear case.

Table 1 Comparison of linear USVM-RFE, Iν-TBSVM, TWSVM, LTWSVM, and GTWSVM on UCI datasets

Dataset	USVM-RFE		Iν-TBSVM		TWSVM		LTWSVM		GTWSVM	
	$c = c_u$ e	Acc (%) Time (s)	c_1=c_2 ν	Acc (%) Time (s)	$c_1 = c_2$	Acc (%) Time (s)	$c_1 = c_2$	Acc (%) Time (s)	$c_1 = c_2$	Acc (%) Time (s)
Australian	10^{-4}	76.4±4.6	0.125	68.3±12.3	10^{-5}	80.3±10.9	3.4	**85.5±33.9**	0.29	84.0±17.7
690 × 14	10^{-2}	13.63	0.3	2.32		0.41		1.07		**0.21**
Diabet	10^{-4}	69.7±33.5	10^{-5}	65.2±4.0	0.0002	70.6±0.8	3.35	66.7±33.4	10^{-10}	**72.5±17.9**
768× 8	1	16.56	0.2	2.59		0.52		1.17		**0.06**
German	10^{-4}	71.9±24.8	10^{-5}	61.1±7.6	10^{-10}	67.3±3.3	3.4	70 ± 19.5	0.29	**74.5 ±3.9**
1000×24	100	27.12	0.3	3.69		1.13		3.15		**0.09**
Sonar	10^{-4}	72.4±13.3	0.5	65.8±3.7	0.3	74.9±1.6	0.29	73.0 ± 37.2	0.29	**76.4±2.0**
208×60	10^{-2}	1.20	0.5	**0.11**		0.19		0.86		**0.11**
Haberman	10^{-5}	65.2 ±24.4	8	73.2±13.5	10^{-5}	68.4±10.6	37.9	**73.5 ± 37.5**	10^{-10}	41.9 ±5.1
306×3	10^{-4}	2.54	0.9	0.34		0.15		0.37		**0.06**
Heart	10^{-5}	73.2±11.9	0.0625	65.2±9.9	0.002	66.3±3.2	3.4	**74.4±32.5**	3.35	73.7± 9.3
270×14	10^{-4}	2.21	0.2	0.32		0.14		0.41		**0.04**
Housevotes	10^{-5}	83.1±5.6	10^{-5}	61.6 ±9.6	10^{-10}	95.6± 0.0	3.35	95.9±48.8	0.02	**95.9 ±1.6**
435×16	10^{3}	7.36	0.2	2.18		0.60		0.96		**0.08**
Ionosphere	10^{-5}	81.1±6.7	0.5	83.7±10.1	10^{-5}	74.6±0.1	0.29	82.3± 41.6	0.02	**92.0±5.0**
351×34	10^{2}	3.74	0.5	0.22		0.16		1.03		**0.09**
Spect	10^{-4}	67.4±2.0	1	67.8 ±7.7	0.02	68.6±1.3	3.35	**69.3±31.3**	0.29	69.0±3.4
237×22	10^{-2}	2.14	0.6	**0.39**		1.35		6.78		1.38
Splice	10^{-4}	73.2±2.4	0.5	79.8±1.6	10^{8}	**78.9±2.7**	4.2×10^{2}	66.3±33.9	10^{-10}	78.4±7.1
1000×60	10^{-2}	27.33	0.5	2.89		0.82		10^{3}		**0.22**
Wdbc	10^{-2}	88.4±21.3	4	90.5±4.1	38	93.5±1.9	3.35	96.13 ±48.52	0.29	**97.0±17.7**
569×30	10^{-3}	9.07	0.8	1.92		0.28		1.84		**0.09**

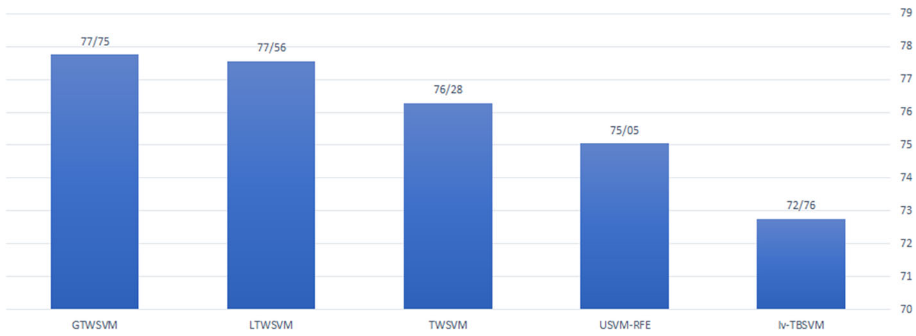

Fig. 7 Average accuracy of linear models on UCI datasets

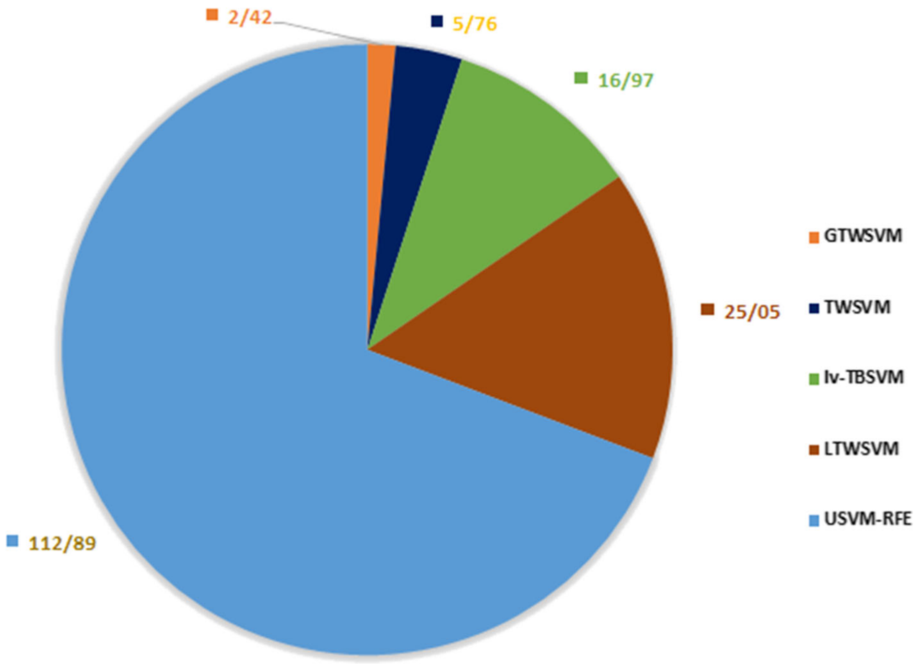

Fig. 8 Sum of CPU time of linear models on UCI datasets

4.5 NDC Datasets

The NDC data sets are generated by using David Musicants NDC Data Generator [33] and is normally distributed. Moosaei et al. introduced an extended version of it with arbitrary number of samples, futures and classes [32]. We further experimented on nine NDC datasets to obtain a vivid representation of how all of these algorithms are scaled in terms of the number of data points. The description of NDC datasets are given in Table 3. In all of these examples, the original data are normalized with mean 0 and standard deviation equals to 1.

Table 4 depicts a comparison of CPU time and accuracy for linear TWSVM, LTWSVM and GTWSVM. We see from Table 4 that our GTWSVM outperforms TWSVM on all the datasets. It is important to note that our GTWSVM was able to perform multiple orders of

Table 2 Comparison of USVM-RFE, Iν-TBSVM, TWSVM, LTWSVM, and GTWSVM using nonlinear Gaussian kernel on UCI datasets

Dataset	USVM-RFE		Iν-TBSVM		TWSVM		LTWSVM		GTWSVM	
	$c_1 = c_2$ & v μ	Acc (%) Time (s)	$c_1 = c_2$ μ	Acc (%) Time (s)	$c_1 = c_2$ μ	Acc (%) Time (s)	$c_1 = c_2$ μ	Acc (%) Time (s)	$c_1 = c_2$ μ	Acc (%) Time(s)
Sonar	10^2, 0.9	81.4±8.2	0.6	74.5±2.8	1.4×10^{-7}	88±0.9	10^{-10}	84.7±17.2	4.3×10^2	**88.9±7.2**
	0.001	1.4	1	**0.1**	3.4	0.2	4.3×10^2	1.4	10^{-10}	0.3
Haberman	10^{-2}, 0.3	67.1±12.4	0.2	75.5±15.3	0.002	**77.5± 3.9**	1.62×10^{-6}	74.9±15.4	1.6×10^{-6}	77.2 ± 18.2
	0.1	2.9	0.01	**0.3**	1.8×10^{-5}	0.3	0.03	1.9	1.4×10^{-7}	5.8
Ionosphere	10^0, 0.1	84.4±3.3	0.1	92.2±6.2	10^{-10}	93.7±1.1	0.3	92.3±18.3	0.03	**94.9 ±14.8**
0.004	3.9	1	**0.4**	0.29	0.43	1.27×10^{-8}	14.99	0.002	8.33	
Wdbc	10^{-1}, 0.1	88.1±10.1	0.3	93.3±13.9	1.2×10^{-8}	94.0±0.2	10^{-10}	93.3±11.0	3.6	**96.5±13.4**
	0.0001	10.4	0.001	**0.8**	1.4×10^{-7}	1.1	2.3×10^{-7}	1.15	1.6×10^{-6}	25.9
Housing	10^{-1}, 0.7	83.9±7.9	0.1	93.1±2.0	0.002	**94.3±0.9**	37.9	93.1±18.1	7.8×10^7	93.7 ±14.2
	0.25	8.3	1	**0.6**	0.02	0.8	10^{-10}	4.3	0.3	22.7

Table 3 Description of NDC datasets

Dataset	Training data	Test data	Features
NDC500	500	50	32
NDC550	550	55	32
NDC1k	1000	100	32
NDC2k	2000	200	32
NDC3k	3000	300	32
NDC5k	5000	500	32
NDC8k	8000	800	32
NDC10k	10, 000	1000	32
NDC50k	50, 000	5000	32

Table 4 Comparisons of linear TWSVM, LTWSVM and GTWSVM on NDC datasets

Dataset	TWSVM		LTWSVM		GTWSVM	
	$c_1 = c_2$	Acc (%) Time (s)	$c_1 = c_2$	Acc (%) Time (s)	$c_1 = c_2$	Acc (%) Time (s)
NDC500	0.002	83.63±1.20	3.35	**84.90** ±**42.97**	3.35	84.72±1.12
		262.74		257.21		**34.12**
NDC1k	1×10^{-10}	83.63±0.45	3.35	84.18±42.62	0.29	**84.81±0.89**
		797.48		953.11		**32.08**
NDC2k	0.002	84.68±0.34	3.35	**85.90±42.84**	0.29	85.27±1.24
		5781		6514.2		**29.90**
NDC3k	0.02	85.54±0.29	3.35	**86.27±32.28**	0.29	85.93±1.29
		1.49×10^4		12032.8		**141.26**
NDC5k	–, –	OTM	– , –	OTM[a]	0.29	**85.47±1.38**
	–	–	–	–		**113.88**
NDC8k	–, –	OTM	– , –	OTM	0.29	**85.64±1.27**
	–	–	–	–		**160.25**
NDC10k	–, –	OTM	– , –	OTM	0.29	**86.01±0.93**
	–	–	–	–		**192.9**
NDC50k	–, –	OTM	– , –	OTM	0.29	**86.01±1.05**
	–	–	–	–		**850.69**

[a] Out of Memory

magnitude faster than TWSVM and LTWSVM. Table 5 shows the comparison of CPU time and accuracy of TWSVM, LTWSVM and GTWSVM on two NDC datasets with nonlinear kernel. We see that our LTWSVM on NDC500 is better than TWSVM, and our GTWSVM is significantly faster than both TWSVM and LTWSVM. Further, we see that GTWSVM has shown better generalization performance on NDC550 dataset.

4.6 Some Applications

Given the impressive performance of our GTWSVM and LTWSVM, we further experimented on two expression profiles for colon tumor and lung cancer. Lung cancer dataset was adopted

Table 5 Comparisons of TWSVM, LTWSVM and GTWSVM using nonlinear kernel on NDC datasets

Dataset	TWSVM		LTWSVM		GTWSVM	
	$c_1 = c_2$ μ	Acc (%) Time (s)	$c_1 = c_2$ μ	Acc (%) Time (s)	$c_1 = c_2$ μ	Acc (%) Time (s)
NDC500	1.83×10^{-5}	92.36±0.48	10^{-10}	**94.18±11.12**	0.29	62±2.44
	10^{-10}	1.56×10^{5}	10^{-10}	13197	10^{-10}	**1.79**
NDC550	8.85×10^{8}	94±0.39	10^{-10}	92.36±11.93	0.02	**96.18±12.08**
	10^{-10}	$\mathbf{1.48}\times10^{4}$	0.002	1.53×10^{5}	10^{-10}	1.12×10^{5}

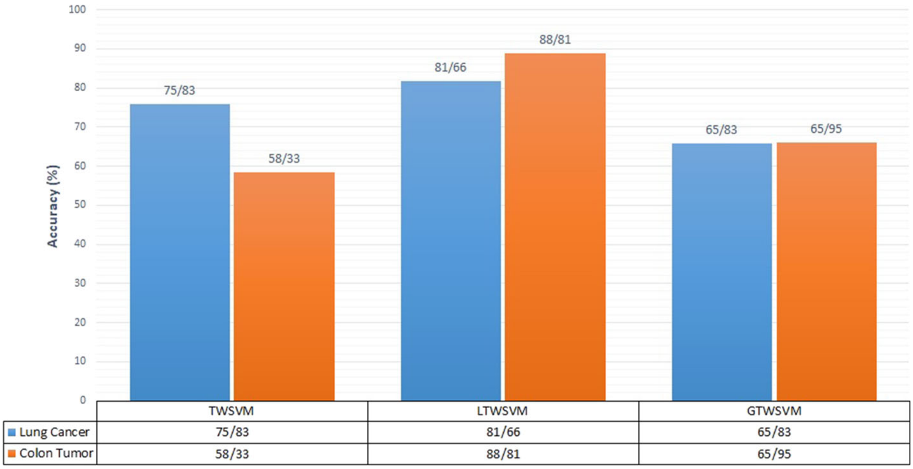

Fig. 9 Comparison of lung cancer and colon tumor datasets for TWSVM, LTWSVM and GTWSVM algorithms

by Hong and Yang [19] to exhibit the strength of the optimal discriminant plane even in settings that are ill posed. Hong and Yang [19] used KNN method in the resulting plane and achieved an accuracy of 77%. Colon tumor dataset consists of 62 samples gathered from colon-cancer patients. Of this figure, 40 tumor biopsies are taken from tumors (labeled as "negative") and 22 from normal (labeled as "positive") healthy parts of the colons of the same patients. 2000 out of around 6500 genes were chosen [1]. When we apply our GTWSVM and LTWSVM on lung cancer dataset, we achieved accuracies 81.66% and 65.83% for LTWSVM and GTWSVM respectively. For colon tumor dataset, our GTWSVM and LTWSVM show 88.81% and 65.95% respectively. However, TWSVM achieved 75.83% and 58.33%, for lung cancer and colon tumor respectively. Figure 9 compare the accuracy between our proposed GTWSVM and LTWSVM, and TWSVM. One can observe from Fig. 9 that our LTWSVM outperforms other two algorithms.

5 Conclusion

In this paper, we proposed a linear programming problem for non-parallel support vector machines by introducing L_1 and L_∞ norms in the primal problems of TWSVM and solve it using interior point method. In addition, a new UMP is proposed for TWSVM and solved by generalized Newton method. Computational comparisons of our LTWSVM and GTWSVM with the other methods on synthetic and several real-world benchmark datasets indicate that our proposed methods have comparable classification accuracy and remarkably less CPU time.

Compliance with Ethical Standards

Conflict of interest The authors have no conflict of interests to declare.

References

1. Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc Natl Acad Sci 96:6745–6750
2. Arabasadi Z, Alizadehsani R, Roshanzamir M, Moosaei H, Yarifard AA (2017) Computer aided decision making for heart disease detection using hybrid neural network-genetic algorithm. Comput Methods Prog Biomed 141:19–26
3. Armijo L (1996) Minimization of functions having lipschitz continuous first partial derivatives. Pacific J Math 16:1–3
4. Bazaraa MS, Sherali HD, Shetty CM (2013) Nonlinear programming: theory and algorithms. Wiley, Hoboken
5. Bazikar F, Ketabchi S, Moosaei H (2020) Dc programming and dca for parametric-margin ν-support vector machine. Appl Intell 50:1–12
6. Burges CJ (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Discov 2:121–167
7. Cao L (2003) Support vector machines experts for time series forecasting. Neurocomputing 51:321–339
8. Clarke F (1990) Optimization and Nonsmooth Analysis, Society for Industrial and Aplied Mathematics
9. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297
10. Deng N, Tian Y, Zhang C (2012) Support vector machines: optimization based theory, algorithms, and extensions. CRC Press, Boca Raton
11. Déniz O, Castrillon M, Hernández M (2003) Face recognition using independent component analysis and support vector machines. Pattern Recogn Lett 24:2153–2157
12. Ding S, Shi S, Jia W (2019) Research on fingerprint classification based on twin support vector machine. IET Image Process 14:231–235
13. Ding S, Zhang N, Zhang X, Wu F (2017) Twin support vector machine: theory, algorithm and applications. Neural Comput Appl 28:3119–3130
14. Dua D, Graff C (2019) Uci machine learning repository, 2017, http://archive.ics.uci.edu/ml, 37
15. Fung GM, Mangasarian OL (2005) Multicategory proximal support vector machine classifiers. Mach Learn 59:77–97
16. Georgiev PG, Theis FJ (2009) Optimization techniques for date representions with biomedical applications. In: Handbook of optimization in medicine, Springer, pp 1–38
17. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46:389–422
18. Hiriart-Urruty JB, Strodiot JJ, Nguyen VH (1984) Generalized hessian matrix and second-order optimality conditions for problems withc 1, 1 data. Appl Math Optim 11:43–56
19. Hong ZQ, Yang JY (1991) Optimal discriminant plane for a small number of samples and design method of classifier on the plane. Pattern Recogn 24:317–324
20. Khemchandani Jayadeva R, Chandra S (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29:905–910
21. Joachims T (1999) Making large-scale svm learning practical. advances in kernel methods-support vector learning, http://svmlight.joachims.org/
22. Ketabchi S, Moosaei H (2012) Minimum norm solution to the absolute value equation in the convex case. J Optim Theory Appl 154:1080–1087
23. Ketabchi S, Moosaei H, Razzaghi M, Pardalos PM (2019) An improvement on parametric ν -support vector algorithm for classification. Ann Oper Res 276:155–168

24. Khemchandani R, Saigal P, Chandra S (2016) Improvements on ν-twin support vector machine. Neural Netw 79:97–107
25. Mangasarian OL (1994) Nonlinear programming, SIAM
26. Mangasarian OL (2002) A finite newton method for classification. Optim Methods Softw 17:913–929
27. Mangasarian OL, Musicant DR (2001) Lagrangian support vector machines. J Mach Learn Res 1:161–177
28. Mangasarian OL, Wild EW (2001) Proximal support vector machine classifiers. In: Proceedings KDD-2001: knowledge discovery and data mining, Citeseer
29. Mangasarian OL, Wild EW (2005) Multisurface proximal support vector machine classification via generalized eigenvalues. IEEE Trans Pattern Anal Mach Intell 28:69–74
30. Mangasarian OL, Wild EW (2006) Multisurface proximal support vector machine classification via generalized eigenvalues. IEEE Trans Pattern Anal Mach Intell 28:69–74
31. Molina GNG, Ebrahimi T, Vesin JM (2003) Joint time-frequency-space classification of eeg in a brain-computer interface application. EURASIP J Adv Signal Process 2003:253269
32. Moosaei H, Musicant D, Khosravi S, Hladík M (2020) MC-NDC: multi-class normally distributed clustered datasets. Carleton College, University of Bojnord. https://github.com/dmusican/ndc
33. Musicant D (1998) NDC: normally distributed clustered datasets
34. Noble WS et al (2004) Support vector machine applications in computational biology. Kernel Methods Comput Biol 71:92
35. Pardalos PM, Ketabchi S, Moosaei H (2014) Minimum norm solution to the positive semidefinite linear complementarity problem. Optimization 63:359–369
36. Peng X (2010) A ν-twin support vector machine (ν-tsvm) classifier and its geometric algorithms. Inf Sci 180:3863–3875
37. Platt J (1999) Fast training of svms using sequential minimal optimization, advances in kernel methods-support vector learning
38. Richhariya B, Tanveer M (2018) Eeg signal classification using universum support vector machine. Exp Syste Appl 106:169–182
39. Richhariya B, Tanveer M, Rashid A, Initiative ADN et al (2020) Diagnosis of alzheimer's disease using universum support vector machine based recursive feature elimination (usvm-rfe). Biomed Signal Process Control 59:101903
40. Ripley B (1996) Pattern recognition and neural networks datasets collection
41. Shao YH, Zhang CH, Wang XB, Deng NY (2011) Improvements on twin support vector machines. IEEE Trans Neural Netw 22:962–968
42. Tanveer M, Khan MA, Ho SS (2016) Robust energy-based least squares twin support vector machines. Appl Intell 45:174–186
43. Tanveer M, Richhariya B, Khan R, Rashid A, Khanna P, Prasad M, Lin C (2020) Machine learning techniques for the diagnosis of alzheimer's disease: a review. ACM Trans Multimedia Comput Commun Appl (TOMM) 16:1–35
44. Tian Y, Qi Z (2004) Review on: twin support vector machines. Ann Data Sci 1:253–277
45. Trafalis TB, 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. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, vol 6, IEEE, pp 348–353
46. Valentini G, Muselli M, Ruffino F (2004) Cancer recognition with bagged ensembles of support vector machines. Neurocomputing 56:461–466
47. Vapnik V (2013) The nature of statistical learning theory. Springer, Berlin
48. Vapnik VN, Chervonenkis AJ (1974) Theory of pattern recognition. Nauka, Moscow
49. Vn V (1998) Statistical learning theory. Wiley, New York
50. Zhang Z, Ding S, Sun Y (2020) A support vector regression model hybridized with chaotic krill herd algorithm and empirical mode decomposition for regression task. Neurocomputing 410:185–201