SMOOTH AUGMENTED LAGRANGIAN METHOD FOR TWIN BOUNDED SUPPORT VECTOR MACHINE

Fatemeh Bazikar and Saeed Ketabchi
Department of Applied Mathematics
Faculty of Mathematical Sciences
University of Guilan, Rasht, Iran

Hossein Moosaei
Department of Mathematics, Faculty of Science
University of Bojnord, Bojnord, Iran

(Communicated by Gerhard-Wilhelm Weber)

Abstract. In this paper, we propose a method for solving the twin bounded support vector machine (TBSVM) for the binary classification. To do so, we use the augmented Lagrangian (AL) optimization method and smoothing technique, to obtain new unconstrained smooth minimization problems for TBSVM classifiers. At first, the augmented Lagrangian method is recruited to convert TBSVM into unconstrained minimization programming problems called as AL-TBSVM. We attempt to solve the primal programming problems of AL-TBSVM by converting them into smooth unconstrained minimization problems. Then, the smooth reformulations of AL-TBSVM, which we called AL-STBSVM, are solved by the well-known Newton’s algorithm. Finally, experimental results on artificial and several University of California Irvine (UCI) benchmark data sets are provided along with the statistical analysis to show the superior performance of our method in terms of classification accuracy and learning speed.

1. Introduction. Support vector machine (SVM) is a powerful tool for data classification and regression has been applied to various real-world problem [1, 2, 39, 15, 19, 27]. In standard SVM, by maximizing the margin between two parallel support hyperplanes, a convex quadratic programming problem (QPP) can be solved. Many improved algorithms such as LIBSVM, PSVM and SMO have been proposed in [5, 31, 34].

Mangasarian et al. [30] suggested nonparallel hyperplane classifiers such as the generalized eigenvalue proximal support vector machine (GEPSVM). GEPSVM is a nonparallel plane classifier that generates two hyperplanes as opposed to SVM that generates one hyperplane. Indeed, GEPSVM is a binary classification method presented to generate two nonparallel hyperplanes such that each plane is close enough to its own class and far from the other class. Inspired by GEPSVM, Jayadeva et al. [16] proposed twin support vector machine (TWSVM), which solves two small-sized quadratic programming problems (QPPs) instead of one large QPP in traditional
They showed that this method has a superior performance in both learning speed and classification accuracy. In the past decade, TWSVM based classifiers have received considerable scholarly attention. Many extensions of TWSVM have been proposed, some of which have been discussed in the survey paper of Tian and Qi [36]. For instance, one extension of TWSVM is the twin bounded support vector machine (TBSVM) proposed by Shao et al. [35]. The main advantage of TBSVM over TWSVM is that the structural risk minimization principle is implemented by introducing the regularization term. In fact, the formulation of TWSVM can be viewed as a special case of TBSVM. Peng [33] proposed a new twin support vector machine (named as ν-TWSVM) for classification, which was a modification of TWSVM. This ν-TWSVM introduces a pair of parameters (ν) to control the bounds of fractions in support vectors and error margins. The parameters ν in the ν-TWSVM have a sound theoretical interpretation compared to the penalty factors in the TWSVM, which control the bounds of fractions for the support vectors and margin errors. Tian et al. [37] proposed improved twin support vector machine (ITWSVM), which is a novel nonparallel hyperplane classifier for binary classification. They introduced different Lagrangian functions for the primal problems in the TWSVM, and get an improved dual formulation of TWSVM. Also, Khemchandani et al. [22] proposed two novel classifiers as improvements on ν-twin support vector machine (Iν-TWSVM and Iν-TWSVM (Fast)), which improve the learning time of TWSVM based classifiers. Similar to ν-TWSVM, Iν-TWSVM determines two nonparallel hyperplanes in a way that they are close to their respective classes and at least ρ away from the other class. The main advantage of Iν-TWSVM over ν-TWSVM is that Iν-TWSVM solves a small-sized quadratic programming problem and an unconstrained minimization problem instead of solving two related quadratic programming problems in ν-TWSVM. After that, Wang et al. [38] proposed improved ν-twin bounded support vector machine (Iν-TBSVM). Recently, Moosaei et al. [25] proposed two efficient approaches of twin support vector machines (TWSVM). They first approach is to reformulate the TWSVM formulation by introducing $L_1$ and $L_\infty$ norms in the objective functions, and convert into linear programming problems termed as LTWSVM for binary classification. And the second approach is to solve the primal TWSVM, and convert into completely unconstrained minimization problem. Also, there have been a number of follow-up studies in the past decade on artificial intelligence and its applications [9, 10, 14, 11, 12].

The augmented Lagrangian (AL) method is highly popular and powerful to solve constraint optimization problems. Recently, Yan and Li proposed an efficient augmented Lagrangian method for SVM [40]. The AL method incorporates constraints into Lagrangian, thereby allowing the use of an unconstrained quadratic optimizer [20, 21]. In optimization community, there have been some progress methods and techniques of dealing with non-smooth problems [6, 7, 24]. Smoothing techniques are extensively used to solve non-smooth programming problems. Chen and Mangasarian introduced a new smooth formulation for the SVM with linear and nonlinear kernel for classification [24]. They converted constrained problem into an unconstrained optimization problem. Since the objective function of this unconstrained optimization problem is not twice differentiable, they employed a smoothing technique to introduce the smooth support vector machine (SSVM). Also, Kumar et al. [23] employed smoothing techniques to TWSVM for binary classification. They
solved the primal TWSVM programming problems by converting them into smooth unconstrained minimization problems.

In this paper, we used the augmented Lagrangian (AL) method to solve the twin bounded support vector machine, obtaining new unconstrained minimization formulations and named AL-TBSVM. Since the objective functions of the AL-TBSVM are not twice differentiable, they cannot be solved by Newton’s method. Therefore, we adopted the smoothing technique for unconstrained optimization problems, converting them into smooth unconstrained optimization problems (known as AL-STBSVM). The smoothing technique involved smoothing the plus function \( x_+ \) using a smoothing function \( p(x; \nu) \) with \( \nu > 0 \) [24]. When the smooth plus function \( p(x; \nu) \) is used instead of plus function, the AL-STBSVM becomes twice differentiable, and therefore solvable with the Newton’s method. We also extended the smooth approach to the nonlinear version AL-TBSVM. Computational comparison of AL-STBSVM with TWSVM, TBSVM and \( \nu \)-TBSVM was made in terms of classification accuracy and computing time for several artificial and UCI data sets in both linear and nonlinear kernels. In addition, the robust and fuzzy version of AL-STBSVM can be consider in future work [28, 29].

The contributions of the paper are as follows:

• Solving the primal problems of TBSVM instead of dual problems.
• Using AL method, which allows solving the primal problems of TBSVM and converts the constrained problems into unconstrained minimization problems.
• We adopted a smoothing technique for unconstrained optimization problems (AL-TBSVM) and converted them into smooth unconstrained optimization problems.
• We also extended the our approach to the nonlinear version of AL-TBSVM.

The rest of the paper is organized as follows: The basic principles of TWSVM and TBSVM are introduced in sections 2 and 3. AL-STBSVM and its theoretical analysis are proposed in section 4. The results of all numerical experiments are shown in section 5 and conclusions are drawn in section 6.

**Notation.** The \( n \)-dimensional real vector space is denoted by \( \mathbb{R}^n \). We consider a binary classification problem of classifying \( m \) training points; \( m_1 \) training points belonging to class +1 and \( m_2 \) training points belonging to class −1 in the \( n \)-dimensional real space \( \mathbb{R}^n \). Matrix \( A \) in \( \mathbb{R}^{m_1 \times n} \) represents the training points of class +1 and matrix \( B \) in \( \mathbb{R}^{m_2 \times n} \) represents the training points of class −1. \( A^\top \) and \( \| \cdot \| \) are notations for the transpose of a matrix \( A \) and the Euclidean norm, respectively. The plus function \( x_+ \) is defined as \( (x_+)_i = \max\{x_i, 0\} \), \( i = 1, \ldots, n \). Let \( f \) be a real valued function on \( \mathbb{R}^n \); its gradient at a point \( x \) is represented by the \( n \)-dimensional column vector \( \nabla f(x) \). The gradient of the function \( f \) with respect to the variable \( x \) is denoted by \( \nabla_x f \). Next, \( x^\top y \) represents the inner product of two \( n \)-dimensional vectors \( x \) and \( y \). \( e_1 \) and \( e_2 \) are column vectors of ones of appropriate dimensions; \( w_1, w_2 \) and \( u_1, u_2 \) are weight vectors in linear and nonlinear state, respectively. \( e_i \) for \( i = 1, 4 \) are penalty parameters and \( \xi_1, \xi_2 \) are vectors of slack variables. \( \alpha_1, \alpha_2, \beta_1, \beta_2 \) are dual variables and vector of Lagrange multipliers. \( \lambda_1, \lambda_2, \mu_1, \mu_2 \) are vectors of Augmented Lagrange multipliers and \( \lambda_1^k, \lambda_2^k, \mu_1^k, \mu_2^k \) are vectors of Augmented Lagrange multipliers in iteration \( k \).

\( L(.) \) is the Lagrangian function and \( L_A(.) \) is Augmented Lagrangian function. For \( A \in \mathbb{R}^{m \times n} \) and \( C \in \mathbb{R}^{n \times l} \), a kernel \( K(A, C) \) is a function that maps \( \mathbb{R}^{m \times n} \times \mathbb{R}^{n \times l} \) into \( \mathbb{R}^{m \times l} \). Particularly, if \( x \) and \( y \) are column vectors in \( \mathbb{R}^n \) then, \( K(x^\top; y) \) is a
real number, $K(x^T; A^T)$ is a row vector in $R^n$ and $K(A; A^T)$ is an $m \times m$ matrix. The identity matrix is denoted by $I$.

2. Twin support vector machine (TWSVM). In this section, we present a brief plan of the twin support vector machine (TWSVM) [16]. TWSVM is a binary classifier that specifies two nonparallel hyperplanes. The two nonparallel hyperplanes of TWSVM are given by $x^T w_1 + b_1 = 0$ and $x^T w_2 + b_2 = 0$, when $w_1, w_2 \in R^n$ and $b_1, b_2 \in R$. The data points belonging to classes 1 and $-1$ are indicated by matrices $A$ and $B$, respectively. Therefore, the size of matrices $A$ and $B$ are $m_1 \times n$ and $m_2 \times n$. The TWSVM classifier is obtained by solving the following pair of quadratic programming problems (QPPs). Figure 1 illustrates the TWSVM graphically.

\[
\begin{align*}
\min_{w_1, b_1, \xi_2} & \quad \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + c_1 e_2^\top \xi_2 \\
\text{s.t.} & \quad -(Bw_1 + e_2 b_1) + \xi_2 \geq e_2, \\
& \quad \xi_2 \geq 0,
\end{align*}
\]  
\[
\begin{align*}
\min_{w_2, b_2, \xi_1} & \quad \frac{1}{2} \|Bw_2 + e_2 b_2\|^2 + c_2 e_1^\top \xi_1 \\
\text{s.t.} & \quad (Aw_2 + e_1 b_2) + \xi_1 \geq e_1, \\
& \quad \xi_1 \geq 0.
\end{align*}
\]

Where $c_1, c_2$ are penalty parameters and $\xi_1 \in R^{m_1}$, $\xi_2 \in R^{m_2}$ are slack vectors $e_1, e_2$ are vectors of ones by dimensions $m_1 \times 1$ and $m_2 \times 1$, respectively. In [16] suggested to solve the dual problems of (1) and (2), to do this the Lagrangian function for the problem (1) can be written as

\[
L(w_1, b_1, \xi_2, \alpha_1, \beta_1) = \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + c_1 e_2^\top \xi_2 - \alpha_1^\top (-Bw_1 - e_2 b_1 - e_2 + \xi_2) - \beta_1^\top \xi_2,
\]

where the Lagrangian multipliers $\alpha_1, \alpha_2 \in R^{m_2}$ and $\beta_1, \beta_2 \in R^{m_2}$. By Karush-Kuhn-Tucker (KKT) conditions, we have

\[
A^\top (Aw_1 + e_1 b_1) + B^\top \alpha_1 = 0,
\]

Figure 1. Illustration of TWSVM
\[ e_1^T (Aw_1 + e_1b_1) + e_2^T \alpha_1 = 0, \quad (5) \]
\[ c_1e_2 - \alpha_1 - \beta_1 = 0, \quad (6) \]
\[ -(Bw_1 + e_2b_1) + \xi_2 \geq e_2, \quad \xi_2 \geq 0, \quad (7) \]
\[ \alpha_1^T (- (Bw_1 + e_2b_1) + \xi_2 - e_2) = 0, \quad \beta_1^T \xi_2 = 0, \quad (8) \]
\[ \alpha_1 \geq 0, \quad \beta_1 \geq 0, \quad (9) \]

since \( \beta_1 \geq 0 \), from (6) we have
\[ 0 \leq \alpha_1 \leq c_1e_2, \quad (10) \]

combining (4) and (5), we have
\[ [A^T e_1^T] [A \ e_1] [w_1 \ b_1]^T + [B^T e_2^T] \alpha_1 = 0. \quad (11) \]

Define \( H = [A \ e_1] \), \( G = [B \ e_2] \) and let \( z_1 = [w_1, b_1]^T \), therefore by these notations, (11) may be rewritten as
\[ H^T H z_1 + G^T \alpha_1 = 0. \]

In other word \( z_1 = - (H^T H)^{-1} G^T \alpha_1 \). By using (3) and above KKT conditions, the Wolfe dual of QPP (1) can be obtained as follows:
\[ \max_{\alpha_1} - \frac{1}{2} \alpha_1^T G(H^T H)^{-1} G^T \alpha_1 + \alpha_1^T e_2 \]
\[ \text{s.t. } 0 \leq \alpha_1 \leq c_1e_2, \quad (12) \]

by using the similar method, we can also get the dual of QPP (2) as follows:
\[ \max_{\alpha_2} - \frac{1}{2} \alpha_2^T H(G^T G)^{-1} H^T \alpha_2 + \alpha_2^T e_1 \]
\[ \text{s.t. } 0 \leq \alpha_2 \leq c_2e_1, \quad (13) \]

In nonlinear state, by using the kernel function \( K(., .) \) [16], the two hyperplanes of TWSVM based on kernel space can be expressed as
\[ K(x^T, D^T) u_1 + b_1 = 0, \quad K(x^T, D^T) u_2 + b_2 = 0, \quad (14) \]

where \( D^T = [A \ B]^T \). So the nonlinear case of QPPs (1) and (2) are as follows
\[ \min_{u_1, b_1, \xi_2} \frac{1}{2} \| K(A, D^T) u_1 + e_1b_1 \|^2 + c_1 e_2^T \xi_2 \]
\[ \text{s.t. } - (K(B, D^T) u_1 + e_2b_1) + \xi_2 \geq e_2, \]
\[ \xi_2 \geq 0, \quad (15) \]

and
\[ \min_{u_2, b_2, \xi_1} \frac{1}{2} \| K(B, D^T) u_2 + e_2b_2 \|^2 + c_2 e_1^T \xi_1 \]
\[ \text{s.t. } (K(A, D^T) u_2 + e_1b_2) + \xi_1 \geq e_1, \]
\[ \xi_1 \geq 0. \quad (16) \]

As with TWSVM solves the optimal solution is also the way to solve its dual problem in dual space. In TWSVM we solve constrained quadratic programming problems by using dual problems with MATLAB Quadprog optimization toolbox.
3. **Twin bounded support vector machine (TBSVM).** Twin bounded support vector machine is proposed in [35]. Similar to TWSVM, TBSVM constructs two nonparallel hyperplanes by solving two smaller QPPs. TBSVM finds the two nonparallel hyperplanes \( x^\top w_1 + b_1 = 0 \) and \( x^\top w_2 + b_2 = 0 \). In TBSVM, patterns of class 1 are clustered around the plane \( x^\top w_1 + b_1 = 0 \), and patterns of class \(-1\) cluster around the plane \( x^\top w_2 + b_2 = 0 \). The standard TBSVM, are obtained by maximizing this margin between two nonparallel planes and are equivalent to the following problems:

\[
\begin{align*}
\min_{w_1, b_1} & \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + \frac{c_3}{2} (\|w_1\|^2 + b_1^2) \\
\text{s.t.} & - (Bw_1 + e_2 b_1) \geq e_2, \\
\end{align*}
\]

and

\[
\begin{align*}
\min_{w_2, b_2} & \frac{1}{2} \|Bw_2 + e_2 b_2\|^2 + \frac{c_4}{2} (\|w_2\|^2 + b_2^2) \\
\text{s.t.} & (Aw_2 + e_1 b_2) \geq e_1, \\
\end{align*}
\]

when the two classes are not strictly linearly separable, there will be an error in satisfying the constraints problems (17) and (18) for some patterns and we can modify problems (17) and (18) to

\[
\begin{align*}
\min_{w_1, b_1, \xi_2} & \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + \frac{c_3}{2} (\|w_1\|^2 + b_1^2) + c_1 e_2^\top \xi_2 \\
\text{s.t.} & - (Bw_1 + e_2 b_1) + \xi_2 \geq e_2, \\
& \xi_2 \geq 0, \\
\end{align*}
\]

and

\[
\begin{align*}
\min_{w_2, b_2, \xi_1} & \frac{1}{2} \|Bw_2 + e_2 b_2\|^2 + \frac{c_4}{2} (\|w_2\|^2 + b_2^2) + c_2 e_1^\top \xi_1 \\
\text{s.t.} & (Aw_2 + e_1 b_2) + \xi_1 \geq e_1, \\
& \xi_1 \geq 0. \\
\end{align*}
\]

Where \( \xi_1 \) and \( \xi_2 \) denote the error variables and \( c_i, i = 1, ..., 4 \) are positive parameters. The Lagrangian function of (19) is defined as follows:

\[
L(w_1, b_1, \xi_2, \alpha_1, \beta_1) = \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + \frac{c_3}{2} (\|w_1\|^2 + b_1^2) + c_1 e_2^\top \xi_2 \\
- \alpha_1^\top (-Bw_1 - e_2 b_1 - \xi_2) - \beta_1^\top \xi_2. \\
\]

Where \( \alpha_1, \alpha_2 \) and \( \beta_1, \beta_2 \) are the Lagrangian multipliers. By KKT conditions for (19), we can get

\[
A^\top (Aw_1 + e_1 b_1) + c_3 w_1 + B^\top \alpha_1 = 0, \\
\]

\[
e_1^\top (Aw_1 + e_1 b_1) + e_2^\top \alpha_1 = 0, \\
\]

\[
c_1 e_2 - \alpha_1 - \beta_1 = 0, \\
\]

\[
-(Bw_1 + e_2 b_1) + \xi_2 \geq e_2, \quad \xi_2 \geq 0, \\
\]

\[
\alpha_1^\top (-Bw_1 - e_2 b_1) + \xi_2 - e_2 = 0, \quad \beta_1^\top \xi_2 = 0, \\
\]

\[
\alpha_1 \geq 0, \quad \beta_1 \geq 0, \\
\]

\[
\text{where } A, B, c_1, c_2, e_1, e_2, \xi_2, \alpha_1, \beta_1 \text{ are the matrices and scalars defined as follows:} \\
\]

\[
A = (a_{ij}), \quad B = (b_{ij}), \quad c_1 = (c_{1i}), \quad c_2 = (c_{2i}), \quad e_1 = (e_{1i}), \quad e_2 = (e_{2i}), \quad \xi_2 = (\xi_{2i}), \\
\]

\[
\alpha_1 = (\alpha_{1i}), \quad \beta_1 = (\beta_{1i}). \\
\]

\[
\text{and } w_1 = (w_{1i}), \quad b_1 = (b_{1i}), \quad e_1 = (e_{1i}), \quad e_2 = (e_{2i}), \quad \xi_2 = (\xi_{2i}), \\
\]

\[
\alpha_1 = (\alpha_{1i}), \quad \beta_1 = (\beta_{1i}). \\
\]
since $\beta_1 \geq 0$, from (24) we can get

$$0 \leq \alpha_1 \leq c_1 e_2.$$  \hfill (28)

combining (22) and (23), we can get

$$\left( [A^T \; e_1] [A \; e_1] + c_3 I \right) [w_1 \; b_1]^T + [B^T \; e_2] \alpha_1 = 0.$$  \hfill (29)

Defining $H = [A \; c_1], \ G = [B \; e_2]$ and $z_1 = [w_1, \ b_1]^T$, with these notations, (29) can be rewritten as

$$(H^T H + c_3 I) z_1 + G^T \alpha_1 = 0.$$  \hfill (30)

Equation (30) is equivalent to

$$z_1 = - (H^T H + c_3 I)^{-1} G^T \alpha_1.$$ 

So, the dual problem (21) is present as follows based on the KKT conditions.

$$\begin{align*}
\max_{\alpha_1} & - \frac{1}{2} \alpha_1^T G (H^T H + c_3 I)^{-1} G^T \alpha_1 + \frac{1}{2} \alpha_1^T e_2 \\
\text{s.t.} & \quad 0 \leq \alpha_1 \leq c_1 e_2.
\end{align*}$$  \hfill (31)

Similarly, the dual problem (22) can be obtained as follows:

$$\begin{align*}
\max_{\alpha_2} & - \frac{1}{2} \alpha_2^T H (G^T G + c_4 I)^{-1} H^T \alpha_2 + \frac{1}{2} \alpha_2^T e_1 \\
\text{s.t.} & \quad 0 \leq \alpha_2 \leq c_2 e_1.
\end{align*}$$  \hfill (32)

In nonlinear state, we have the following minimization programming problems

$$\begin{align*}
\min_{u_1, b_1, \xi_2} & \quad \frac{1}{2} \|K (A, D^T) u_1 + e_1 b_1\|^2 + \frac{c_3}{2} (\|u_1\|^2 + b_1^2) + c_1 e_2^T \xi_2 \\
\text{s.t.} & \quad -(K (B, D^T) u_1 + e_2 b_1) + \xi_2 \geq e_2, \\
& \quad \xi_2 \geq 0,
\end{align*}$$  \hfill (33)

and

$$\begin{align*}
\min_{u_2, b_2, \xi_1} & \quad \frac{1}{2} \|K (B, D^T) u_2 + e_2 b_2\|^2 + \frac{c_4}{2} (\|u_2\|^2 + b_2^2) + c_2 e_2^T \xi_1 \\
\text{s.t.} & \quad (K (A, D^T) u_2 + e_1 b_2) + \xi_1 \geq e_1, \\
& \quad \xi_1 \geq 0,
\end{align*}$$  \hfill (34)

where $D^T = [A, \ B]^T$ and $K(\cdot, \cdot)$ is any arbitrary kernel function. So, we define a Lagrangian function of (33) as follows:

$$L(u_1, b_1, \xi_2, \alpha_1, \beta_1) = \frac{1}{2} \|K (A, D^T) u_1 + e_1 b_1\|^2 + \frac{c_3}{2} (\|u_1\|^2 + b_1^2) + c_1 e_2^T \xi_2$$

$$+ c_1 e_2^T \xi_2 - \alpha_1^T (-(K (B, D^T) u_1 - e_2 b_1 - e_2 + \xi_2) - \beta_1^T \xi_2).$$

Similarly, the Lagrangian function of (34) can be obtained. By using the KKT conditions for problems (33) and (34), the dual nonlinear programming problems are obtained as follows:

$$\begin{align*}
\max_{\alpha_1} & \quad - \frac{1}{2} \alpha_1^T S (R^T R + c_3 I)^{-1} S^T \alpha_1 + \alpha_1^T e_2 \\
\text{s.t.} & \quad 0 \leq \alpha_1 \leq c_1 e_2.
\end{align*}$$  \hfill (36)
and
\[
\max_{\alpha_2} -\frac{1}{2} \alpha_2^T R (S^T S + c_4 I)^{-1} R^T \alpha_2 + \alpha_2^T e_1
\]
\[
\text{s.t.} \quad 0 \leq \alpha_2 \leq c_2 e_1.
\]

Where \( R = [K (A, D^T) \ e_1], S = [K (B, D^T) \ e_2] \) and \( \alpha_1, \alpha_2, \beta_1 \) and \( \beta_2 \) are the Lagrangian multipliers. We solve constrained quadratic programming problems TBSVM by using dual problems with MATLAB Quadprog optimization toolbox. Thus, for nonlinear state TBSVM once the solution \((u_1, b_1)\) of problem (33) is obtained from the solution of (36).

4. Augmented Lagrangian method and smooth technique for solving TBSVM (AL-STBSVM). In this section, we improved the primal problems of TBSVM by using the augmented Lagrangian method and smooth technique for solving TBSVM. The AL method is also known as the method of multipliers and solves the constrained optimization problem. The general idea behind the AL method is to incorporate some or all of the constraints into the objective function \([3, 4, 32]\). Using 2-norm slack variables \( \xi_1 \) and \( \xi_2 \) in objective functions of (19) and (20) lead to the following minimization problems.

\[
\min_{w_1, b_1, \xi_1, \xi_2} \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + \frac{c_3}{2} (\|w_1\|^2 + b_1^2) + \frac{c_1}{2} \|\xi_2\|^2
\]
\[
\text{s.t.} \quad -(Bw_1 + e_2 b_1) + \xi_2 \geq e_2,
\]
\[
\xi_2 \geq 0,
\]

and

\[
\min_{w_2, b_2, \xi_1} \frac{1}{2} \|Bw_2 + e_2 b_2\|^2 + \frac{c_4}{2} (\|w_2\|^2 + b_2^2) + \frac{c_2}{2} \|\xi_1\|^2
\]
\[
\text{s.t.} \quad (Aw_2 + e_1 b_2) + \xi_1 \geq e_1,
\]
\[
\xi_1 \geq 0.
\]

In the AL method, we add a quadratic penalty to the Lagrangian function. For all \( \rho, r > 0, \lambda_1, \lambda_2 \in R^{m_2} \) and \( \mu_1, \mu_2 \in R^{m_1} \), the AL function problems (38) and (39), respectively will be given by

\[
L_A (w_1, b_1, \xi_1, \lambda_1, \lambda_2) = \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + \frac{c_3}{2} (\|w_1\|^2 + b_1^2) + \frac{c_1}{2} \|\xi_2\|^2
\]
\[
+ \frac{\rho}{2} \|(e_2 - \xi_2 + (Bw_1 + e_2 b_1) + \frac{\lambda_1}{\rho} + \frac{\xi_2}{\rho})\|_2^2
\]
\[
+ \frac{\rho}{2} \|(e_1 - \xi_1 - (Aw_2 + e_1 b_2) + \frac{\mu_1}{r} + \frac{\xi_1}{r})\|_2^2,
\]

and

\[
L_A (w_2, b_2, \xi_1, \mu_1, \mu_2) = \frac{1}{2} \|Bw_2 + e_2 b_2\|^2 + \frac{c_4}{2} (\|w_2\|^2 + b_2^2) + \frac{c_2}{2} \|\xi_1\|^2
\]
\[
+ \frac{r}{2} \|(e_1 - \xi_1 - (Aw_2 + e_1 b_2) + \frac{\mu_1}{r} + \frac{\xi_1}{r})\|_2^2
\]
\[
+ \frac{r}{2} \|(e_2 - \xi_2 + (Bw_1 + e_2 b_1) + \frac{\lambda_2}{r} + \frac{\xi_2}{r})\|_2^2.
\]

By using the AL method for the inequality-constrained programming problems (38) and (39), we obtain the following subproblems:

\[
\min_{w_1, b_1, \xi_2} L_A (w_1, b_1, \xi_2, \lambda_1^k, \lambda_2^k) = \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + \frac{c_3}{2} (\|w_1\|^2 + b_1^2) + \frac{c_1}{2} \|\xi_2\|^2
\]
\[
+ \frac{\rho}{2} \|(e_2 - \xi_2 + (Bw_1 + e_2 b_1) + \frac{\lambda_2^k}{\rho} + \frac{\xi_2}{\rho})\|_2^2
\]
\[
+ \frac{\rho}{2} \|(e_1 - \xi_1 - (Aw_2 + e_1 b_2) + \frac{\mu_1}{r} + \frac{\xi_1}{r})\|_2^2.
\]
and

\[
\min_{w_2,b_2,\xi_1} L_A(w_2, b_2, \xi_1, \mu^k_1, \mu^k_2) = \frac{1}{2} \| B w_2 + b_2 \|^2 + \frac{c_1}{2} (\| w_2 \|^2 + b_2^2) + \frac{c_2}{2} \| \xi_1 \|^2 \tag{43}
\]

\[
\quad + \frac{r}{2} ((e_1 - \xi_1 - (A w_2 + e_1 b_2)) + \frac{\mu^k_1}{r}) + (\| (-\xi_1) + \frac{\mu^k_2}{r} \|)^2,
\]

and we use the following formula to update the Lagrangian multipliers:

\[
\lambda^{k+1}_1 = (\lambda^k_1 + \rho (e_2 - \xi_2 + (B w_1 + e_2 b_1)))_+,
\]

\[
\lambda^{k+1}_2 = (\lambda^k_2 + \rho (-\xi_2))_+, \tag{44}
\]

and

\[
\mu^{k+1}_1 = (\mu^k_1 + r (e_1 - \xi_1 - (A w_2 + e_1 b_2)))_+,
\]

\[
\mu^{k+1}_2 = (\mu^k_2 + r (-\xi_1))_+. \tag{45}
\]

This formula follows from the fact that the derivative of \( L_A(w_1, b_1, \xi_1, \lambda_1, \lambda_2) \) and \( L_A(w_2, b_2, \xi_1, \mu_1, \mu_2) \) in (40) and (41) with respect to \( (w_1, b_1, \xi_2) \) and \( (w_2, b_2, \xi_1) \) should be close to zero at the approximate solution \( (w^k_1, b^k_1, \xi^k_2) \) and \( (w^k_2, b^k_2, \xi^k_1) \). By using (40) and (41), we obtain

\[
\nabla L_A(w^k_1, b^k_1, \xi^k_2, \lambda^k_1, \lambda^k_2) \approx 0, \tag{46}
\]

\[
\nabla L_A(w^k_2, b^k_2, \xi^k_1, \mu^k_1, \mu^k_2) \approx 0.
\]

The update formula (42) and (43) follow by comparing this formula to the KKT condition, which we can state as

\[
\nabla L_A(w^k_1, b^k_1, \xi^k_2, \lambda^k_1, \lambda^k_2) = 0, \quad \tag{47}
\]

\[
\nabla L_A(w^k_1, b^k_1, \xi^k_2, \mu^k_1, \mu^k_2) = 0.
\]

The objective functions subproblems (42) and (43) are not twice differentiable, so at first we propose using the smoothing techniques for the subproblems and converting them into twice differentiable problems.

Applying the smoothing techniques [24] and replacing \( \lambda_+ \) by a smooth approximation, we transform these problems to a twice continuously differentiable problems. Chen and Mangasarian [6] introduced a family of smoothing functions, which is built as follows. Let \( S : R \rightarrow [0, \infty) \) be a piecewise continuous density function satisfying

\[
\int_{-\infty}^{\infty} S(s) ds = 1, \quad \int_{-\infty}^{\infty} |s| S(s) ds < \infty. \tag{48}
\]

It is obvious that the derivative of plus function is a step function, that is, \( (x)_+ = \int_{-\infty}^{x} \delta(t) dt \), where the step function \( \delta(x) \) is defined 1 if \( x > 0 \) and equals 0 if \( x \leq 0 \). Therefore, a smoothing approximation function of the plus function is defined by

\[
p(x, \nu) = \int_{-\infty}^{x} \Psi(t, \nu) dt, \tag{49}
\]

where \( \Psi(x, \nu) \) is smoothing approximation function of step function and is defined as

\[
\Psi(x, \nu) = \int_{-\infty}^{x} \nu S(\nu t) dt, \quad S(s) = \frac{e^{-s}}{(1 + e^{-s})^2}. \tag{50}
\]
specific cases of these approaches are obtained as follows:
\[ p(x, \nu) = x + \frac{1}{\nu} \log(1 + e^{-\nu x}) \approx (x)_{+}, \quad \nu > 0. \]  
(52)
The function \( p(x, \nu) \) with a smoothing parameter \( \nu \) is used here to replace the plus function of problems (42) and (43) to obtain a smooth reformulation of the subproblems (42) and (43) as follow:
\[ \min_{w_1,b_1,\xi_2} L_A(w_1,b_1,\xi_2,\lambda_1,\lambda_2) = \frac{1}{2} \|Aw_1 + e_1b_1\|^2 + \frac{c_1}{2} (\|w_1\|^2 + b_1^2) + \frac{c_2}{2} \|\xi_2\|^2 \]  
(53)
and
\[ \min_{w_2,b_2,\xi_1} L_A(w_2,b_2,\xi_1,\mu_1,\mu_2) = \frac{1}{2} \|Bw_2 + e_2b_2\|^2 + \frac{c_1}{2} (\|w_2\|^2 + b_2^2) + \frac{c_2}{2} \|\xi_1\|^2 \]  
(54)
For solving the subproblems (53) and (54) we proposed Newton’s method [18]. The following algorithm describes our AL method for solving subproblems (53) and (54). The algorithm proceeds by minimizing the AL function at each iteration and updating Lagrangian multipliers between iterations.

Algorithm 1
1. Let \( \lambda_i, \mu_i, \rho, r > 0 \), tolerance \( \epsilon_1, \epsilon_2 > 0 \) and initial point \( z_i^1 = (w_i^1, b_i^1, \xi_i^1) \) for \( i = 1, 2 \). Initialize \( k := 1 \).
2. Find \( z_k = (w_k^1, b_k^1, \xi_k^1) \) as an approximate solution of the subproblems (53) and (54), initial at \( z_k^1 \) and terminating when
\[ \|\nabla_z L_A(z, \lambda_1^k, \lambda_2^k)\|_{\infty} > \epsilon_1 \text{ and } \|\nabla_z L_A(z, \mu_1^k, \mu_2^k)\|_{\infty} > \epsilon_2, \]
if final convergence test satisfied stop with approximate solution \( z_k \).
3. Update Lagrangian multipliers using (44) and (45) to obtain \( \lambda_i^{k+1} \) and \( \mu_i^{k+1} \) for \( i = 1, 2 \).
4. Set \( k = k + 1 \) and initial point for the next iteration to \( z_i^{k+1} = z_k \) and go to step 2.

We want to show that solutions of problems (42) and (43) can be obtained by solving their smooth reformulations (53) and (54) as \( \nu \) approaches infinity. As proved by Lee et al. [24], for \( x \in R \) and \( |x| < w \) we have
\[ p^2(x, \nu) - (x)^2 \leq \left( \frac{\log 2}{\nu} \right)^2 + \frac{2w}{\nu} \log 2, \]  
(55)
where \( p(x, \nu) \) is the \( p \) function of (52) with smoothing parameter. In addition, according to Theorem 2.2 and Theorem 6 in [24, 20], there exists a unique solution \( z_i^* \) of (42) and a unique solution \( z_i^* \) of (53) and for all \( \nu > 0 \), \( P = [B_e_2 - I_F] \) (\( I_F \) is matrix of ones by dimension \( m_2 \times m_2 \)) and \( (w_1, b_1, \xi_2) = z_i \in R^{n+m_2+1} \) we have
\[ \|z_i^* - z_i^*\|^2 \leq \mu \left( \frac{2\log 2}{\nu} \right)^2 + 8Mm_2\mu \log 2, \]  
(56)
where $M$ is defined as follows:

$$M = \max_{1 \leq i \leq m_2} | (Pz + e_2)_i | .$$  \hspace{1cm} (57)

Considering the fact that the objective functions (53) and (54) are twice differentiable. Thus, to solve smooth quadratic programming problems (53) and (54), we can use the Newton’s method.

In nonlinear state, where $K(.,.)$ is any arbitrary kernel function and $D^\top = [A \ B]^\top$, problems (53) and (54) can be written as follows:

$$\min_{u_1, b_1, \xi_2} \hat{L}_A(u_1, b_1, \xi_2, \lambda_1^k, \lambda_2^k, \nu) = \frac{1}{2} \|K(A, D^\top)u_1 + e_1b_1\|^2 + \frac{c_3}{2}(\|u_1\|^2 + b_1^2)$$

$$+ \frac{c_1}{2}\|\xi_2\|^2 + \frac{\rho}{2}\|p(e_2 - \xi_2 + K(B, D^\top)u_1 + e_2b_1) + \lambda_1^k, \nu\|^2$$

$$+ \frac{\rho}{2}\|p(-\xi_2 + \lambda_2^k, \nu)\|^2,$$

and

$$\min_{u_2, b_2, \xi_1} \hat{L}_A(u_2, b_2, \xi_1, \mu_1^k, \mu_2^k, \nu) = \frac{1}{2} \|K(B, D^\top)u_2 + e_2b_2\|^2 + \frac{c_4}{2}(\|u_2\|^2 + b_2^2)$$

$$+ \frac{c_2}{2}\|\xi_1\|^2 + \frac{r}{2}\|p(e_1 - \xi_1 - (K(A, D^\top)u_2 + e_1b_2) + \mu_1^k, \nu\|^2$$

$$+ \frac{r}{2}\|p(-\xi_1 + \mu_2^k, \nu)\|^2.$$  \hspace{1cm} (58)

The smooth formulations (60) and (61) with nonlinear kernel retain the twice differentiability and thus we can apply Newton’s algorithm to solve them.

5. Numerical experiments. The numerical results of the algorithms are presented in this section. This idea is discussed over two different performances, i.e. accuracy and learning speed. To verify the efficiency of our proposed method, we decided to compare AL-STBSVM method with $I\nu$-TBSVM, TBSVM and TWSVM on a generated data set and several UCI benchmark data sets [26]. The experiments were performed in MATLAB software (R2016a) on a PC with this system configuration: Intel(R) core(TM) i7 CPU 2.20 G Hz with 4GB of RAM, and Windows 7 operating system. Also, the ten-fold cross-validation was used to evaluate the performance of the classifier and estimate its accuracy. Cross-validation follows these steps:

- Data sets are divided into ten disjoint subsets of equal size.
- Classifier is trained on all subsets except for one.
- By testing an error on the excluded subset, the validation error is computed.
- This process is repeated for ten trials.

The performance of these algorithms largely relies on the choice of parameters. To accelerate the model selection, we tuned a set comprising 20% of randomly-selected training samples to choose optimal parameters.

5.1. Generated data set. In this subsection, some arbitrary points are randomly specified in two classes of $A$ and $B$, which are almost separated from each other linearly based on given MATLAB code in the ”Appendix A” (here, we created 250 points for class $A$ and 250 points for class $B$). Figure 2 shows the learning results of linear TWSVM, TBSVM, $I\nu$-TBSVM and linear AL-STBSVM. By applying the
proposed method to the generated data set, the accuracy of our AL-STBSVM, \( \nu \)-TBSVM, TBSVM and TWSVM was estimated at 98.81%, 81.56%, 98.41% and 97.89% respectively. Moreover, the AL-STBSVM method is faster rather than \( \nu \)-TBSVM, TBSVM and TWSVM methods.

5.2. UCI data sets. In this subsection, we use several UCI data sets for numerical experiments\cite{26}.

5.2.1. Parameter setting. The performance of these algorithms is largely dependent on the choice of parameters. The optimal value for parameters \( c_i \) and \( \nu \), for \( i = 1, \ldots, 4 \) in all methods were selected from the range of \( c_i = \{2^j | j = -10, -9, -8, \ldots, 8, 9, 10\} \) and \( \nu \in \{0.1, \ldots, 0.9\} \). In this experimental part, we used the Gaussian Kernel (i.e. \( K(x, y) = \exp(-\gamma ||x - y||^2) \), \( \gamma > 0 \)), for all data and Gaussian kernel parameter \( \gamma \) was selected from \( \{2^i | i = -10, -9, -8, \ldots, 8, 9, 10\} \).

5.2.2. Results comparisons for UCI data sets. Table 1 summarizes the characteristics of these data sets, and Tables 2 and 3 compares the computing time and accuracy for all four methods in both linear and nonlinear states. The numerical results derived from the linear classifier for AL-STBSVM, \( \nu \)-TBSVM, TBSVM and TWSVM are reported in Table 2. Table 3 compares the performance of the AL-STBSVM classifier with that of \( \nu \)-TBSVM, TBSVM and TWSVM methods using the Gaussian kernel. In Tables 2 and 3, the highest accuracy and optimum time are
shown in bold numbers. From the tables it can be seen that in aspect of accuracy, the linear and nonlinear proposed methods outperform three other methods, ie. \( I_\nu \)-TBSVM, TBSVM and TWSVM in most cases. In Table 2, the mean accuracy of AL-STBSVM, for all the data sets, is 83.31% as compared to 74.11%, 80.53% and 79.08% for \( I_\nu \)-TBSVM, TBSVM and TWSVM respectively. AL-STBSVM is able to achieve the maximum classification accuracy for most of the UCI data sets through linear classifier. Similarly, the classification accuracy for UCI data sets is reported in Table 3 for all four methods with nonlinear kernel. The AL-STBSVM has the highest average accuracy among all mentioned methods as well. UCI data sets are not structurally separated and it makes a lot of sense that the nonlinear state performance better on this data. The accuracy of the proposed method is better in Table 2 and Table 3 than that of the three others. Because the structure of nonlinear classifiers often performs better data classification in all methods. So, in this proposed method, the performance is better in nonlinear classifiers compared with linear.

Table 1. Descriptions of the data sets from the UCI repository.

| Data set    | # Cases | # Features | # Classes | Source |
|-------------|---------|------------|-----------|--------|
| Sonar       | 208     | 60         | 2         | UCI    |
| Cancer      | 699     | 9          | 2         | UCI    |
| Diabet      | 768     | 8          | 2         | UCI    |
| Wdbc        | 569     | 30         | 2         | UCI    |
| Ionosphere  | 351     | 34         | 2         | UCI    |
| Australian  | 690     | 14         | 2         | UCI    |
| Heart       | 270     | 14         | 2         | UCI    |
| Haberman    | 306     | 3          | 2         | UCI    |
| German      | 1000    | 24         | 2         | UCI    |
| House Votes | 435     | 16         | 2         | UCI    |
| Spect       | 237     | 22         | 2         | UCI    |
| Splice      | 1000    | 60         | 2         | UCI    |
| Lung-cancer | 32      | 56         | 2         | UCI    |
| F-diagnosis | 100     | 9          | 2         | UCI    |
| Breast-cancer | 116   | 9          | 2         | UCI    |
| Bupa        | 345     | 6          | 2         | UCI    |
| Pima        | 768     | 9          | 2         | UCI    |
| Housing     | 506     | 14         | 2         | UCI    |

5.3. Statistical analysis. Tables 2 and 3 illustrate the comparison of the test accuracy and learning speed of TWSVM, TBSVM, \( I_\nu \)-TBSVM and AL-STBSVM in linear and nonlinear states. One can observe from Tables 2 and 3 that the proposed AL-STBSVM obtained enhanced test accuracies when compared with TWSVM, TBSVM and \( I_\nu \)-TBSVM for most of the UCI benchmark data sets. To analyze the performance of all the methods on several UCI benchmark data sets statistically, as it was suggested in \[2, 8, 17\], we assumed Friedman test with the corresponding post hoc tests which are considered to be a simple, nonparametric yet safe test. For this, the average ranks of all the methods on accuracies were computed and listed in Tables 4 and 5. This test ranks methods for individual data sets, with the best performing method receiving the first rank, and the other methods are ranked accordingly as shown in Tables 4 and 5. In case two methods perform similarly, an
| Data set         | TWSVM  | TBSVM  | Iν-TBSVM | AL-STBSVM |
|------------------|--------|--------|----------|-----------|
| size             | c1 = c2 | c1 = c4, c1 = c2 | c1 = c2, ν | c1 = c4, c1 = c2 |
| Sonar 208×60     | 76.88, 0.73 | 77.45, 1.53 | 70.29, 1.53 | **79.14, 0.52** |
| Cancer 699×9     | 96.13, 1.63 | **96.28, 2.63** | 90.13, 2.28 | 96.15, 1.99 |
| Diabet 768×8     | 69.13, 1.90 | **73.58, 2.68** | 62.24, 3.39 | 71.74, 27.35 |
| Wdbc 569×30      | 93.66, 1.6 | 94.74, 2.37 | 90.87, 2.0 | **95.62, 1.61** |
| Ionosphere 351×34| 84.89, 0.86 | **86.30, 1.71** | 84.03, 1.67 | 85.21, 0.62 |
| Australian 690×14| 83.05, 1.62 | **84.05, 2.68** | 67.66, 3.30 | 83.90, 2.93 |
| Heart 270×14     | 84.44, 0.98 | 84.44, 1.57 | 83.70, 1.64 | **85.19, 0.63** |
| Haberman 306×3   | 74.51, 0.86 | 75.91, 1.64 | 72.91, 1.66 | **77.05, 0.55** |
| German 1000×24   | 73.9, 1.78 | **75.4, 3.94** | 61.8, 6.59 | 75.1, 11.42 |
| House Votes 435×16| 95.62, 0.85 | 95.85, 1.90 | 93.11, 1.67 | **96.08, 0.65** |
| Spect 237×22     | 68.60, 0.75 | 70.44, 1.59 | **73.57, 1.6** | 71.24, 0.55 |
| Splice 1000×60   | 75.40, 2.44 | **79.70, 3.77** | 78.20, 5.63 | 79.18, 10.19 |
| Lung-cancer 32×56| 82.5, 0.62 | 82.51, 1.40 | 85.1, 1.42 | **85.83, 0.42** |
| F-diagnosis 100×9| 76.49, 0.62 | **82.34, 1.40** | 68.35, 1.48 | 75.01, 0.49 |
| Breast-cancer 116×9| 72.52, 0.61 | 71.84, 1.38 | 57.04, 1.52 | **73.81, 0.45** |
| Bupa 345×6       | 64.35, 0.73 | 65.23, 1.69 | **69.29, 1.72** | 66.68, 0.58 |
| Pima 768×9       | 71.36, 0.86 | 73.30, 1.64 | 64.72, 2.65 | **73.62, 2.04** |
| Housing 506×14   | 80.08, 0.96 | 80.24, 2.89 | 61.11, 2.01 | **93.09, 1.5** |
| Avg.acc          | 79.08 | 80.53 | 74.11 | **83.31** |

**Table 2.** Comparison of linear TWSVM, TBSVM, Iν-TBSVM and proposed model (AL-STBSVM) on UCI benchmark data sets.
Table 3. Comparison of nonlinear TWSVM, TBSVM, $\nu$-TBSVM and proposed model (AL-STBSVM) on UCI benchmark data sets.

| Dataset        | TWSVM | TBSVM | $\nu$-TBSVM | AL-STBSVM |
|----------------|-------|-------|-------------|-----------|
| size           | Acc(%) ,Time(s) | Acc(%) ,Time(s) | Acc(%) ,Time(s) | Acc(%) ,Time(s) |
| $c_1 = c_2, \gamma$ | $c_3 = c_4, c_1 = c_2, \gamma$ | $c_3 = c_4, c_1 = c_2, \gamma$ |
| Sonar 208×60   | 84.53, 0.80 | 86.14, 1.62 | 83.27, 1.67 | **87.54, 0.84** |
| Cancer 699×9   | 96.42, 4.71 | 96.42, 6.59 | 96.02, **2.90** | **96.85, 3.85** |
| Diabet 768×8   | 65.49, 10.18 | 66.28, 8.21 | 65.11, **5.33** | **69.14, 21.28** |
| Wdbc 569×30    | 94.89, 1.42 | 94.54, 2.15 | 87.17, 1.88 | **94.89, 2.96** |
| Australian 351×34 | 73.85, 1.28 | 73.52, 2.01 | 73.19, 1.88 | 73.52, 2.67 |
| Heart 270×14   | 82.59, 0.92 | **83.33, 1.86** | 81.48, 1.76 | **83.33, 1.58** |
| Haberman 306×3 | 70.1, 14.69 | 70.2, 14.34 | 70, **7.33** | **71.5, 35.33** |
| House Votes 435×16 | 92.64, **1.22** | 93.55, 2.20 | 91.70, 1.78 | **94.71, 4.12** |
| Spect 237×22   | 71.89, **0.86** | 73.76, 1.66 | 68.90, 1.66 | **74.17, 1.45** |
| Splice 1000×60 | 76.71, 15.98 | 75.20, 16.58 | 74.41, **14.66** | **87.20, 42.10** |
| Lung-cancer 32×56 | 80.83, 0.67 | **85.83, 1.41** | 84.16, 1.51 | **85, 0.35** |
| F-diagnosis 100×9 | 88.14, 0.64 | 88.14, 1.42 | 87.16, 1.59 | **89.25, 0.44** |
| Breast-cancer 116×9 | 55.16, 0.65 | 55.16, 1.48 | 55.15, 1.56 | **60.31, 0.48** |
| Bupa 345×6     | 64.37, **1.19** | 64.35, 2.02 | 65.80, 1.94 | **66.36, 2.62** |
| Pima 768×9     | 65.63, 9.66 | 66.15, 8.08 | 65.12, **4.74** | **69, 20.07** |
| Housing 506×14 | **93.09, 2.12** | **93.09, 3.65** | **93.09, 4.30** | **93.09, 7.01** |
| Avg.acc        | 76.37 | 76.86 | 77.16 | **80.46** |
average rank is assigned to both. Let \( r_i^j \) be the rank of the \( j \)-th classifiers on the \( i \)-th data set. This test compares the average rank of the methods \( R_j = \frac{1}{N} \sum_i r_i^j \). The Friedman test statistic is calculated as follows:

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

where \( k \) is the number of algorithms and \( N \) is the number of data sets. Iman and Davenport [13] showed that Friedman \( \chi^2_F \) is too conservative and proposed a better statistic as follows:

\[
F_f = \frac{(N-1)\chi^2_F}{N(k-1)-\chi^2_F},
\]

which is distributed according to the \( F \)-distribution with \(((k - 1), (k - 1)(N - 1))\) degrees of freedom. According to these Tables, bold values show the best average rank of each method, and the AL-STBSVM method has the highest Friedman score (average rank) among all the algorithms for both linear and nonlinear states.

Table 4. Rank of accuracy linear classifiers on UCI benchmark data sets.

| Data set     | TWSVM | TBSVM | \( \nu \)-TBSVM | AL-STBSVM |
|--------------|-------|-------|-----------------|-----------|
| Sonar        | 3     | 2     | 4               | 1         |
| Cancer       | 3     | 1     | 4               | 2         |
| Diabet       | 3     | 1     | 4               | 2         |
| Wdbc         | 3     | 2     | 4               | 1         |
| Ionosphere   | 3     | 1     | 4               | 2         |
| Australian   | 3     | 1     | 4               | 2         |
| Heart        | 2.5   | 2.5   | 4               | 1         |
| Haberman     | 3     | 2     | 4               | 1         |
| German       | 3     | 1     | 4               | 2         |
| House Votes  | 3     | 2     | 4               | 1         |
| Spect        | 4     | 3     | 1               | 2         |
| Splice       | 4     | 1     | 3               | 2         |
| Lung-cancer  | 3.5   | 3.5   | 2               | 1         |
| F-diagnosis  | 2     | 1     | 4               | 3         |
| Breast-cancer | 2    | 3     | 4               | 1         |
| Bupa         | 4     | 3     | 1               | 2         |
| Pima         | 3     | 2     | 4               | 1         |
| Housing      | 3     | 2     | 4               | 1         |
| Average rank | 3.06  | 1.89  | 3.5             | 1.56      |

\( \chi^2_F \) and \( F_f \) are respectively obtained as follows using Table 4:

\[
\chi^2_F = \frac{12 \times 18}{4(4+1)} [(3.06)^2 + (1.89)^2 + (3.5)^2 + (1.56)^2 - \frac{4(4+1)^2}{4}] = 27.41,
\]
Similarly, $\chi^2_F = 25.62$ and $F_f = 15.23$ were obtained using Table 5. Given three methods and 18 UCI data sets for the linear and nonlinear states, $F_f$ is distributed according to the $F$-distribution with $((k - 1), (k - 1)(N - 1)) = (3, 51)$ degrees of freedom. The critical values of $F(3, 51)$ are respectively $F(3, 51) = 2.79$ and $F(3, 51) = 4.19$ for $\alpha = 0.05$ and $\alpha = 0.01$. Tables 4, 5 and the critical value of $F(3, 51)$ when $\alpha = 0.05$ and $\alpha = 0.01$ show a significant difference between the performance of the methods. The performance of the proposed method is therefore obviously better than that of the three others in terms of accuracy as shown in Tables 4 and 5.

**Table 5.** Rank of accuracy nonlinear classifiers on UCI benchmark data sets.

| Data set      | TWSVM | TBSVM | 4-\nu-TBSVM | AL-STBSVM |
|---------------|-------|-------|-------------|-----------|
| Sonar         | 3     | 2     | 4           | 1         |
| Cancer        | 2.5   | 2.5   | 4           | 1         |
| Diabet        | 3     | 2     | 4           | 1         |
| Wdbc          | 3.5   | 3.5   | 1           | 2         |
| Ionosphere    | 1.5   | 3     | 4           | 1.5       |
| Australian    | 2     | 4     | 3           | 1         |
| Heart         | 3     | 1.5   | 4           | 1.5       |
| Haberman      | 1     | 2.5   | 4           | 2.5       |
| German        | 3     | 2     | 4           | 1         |
| House Votes   | 3     | 2     | 4           | 1         |
| Spect         | 3     | 2     | 4           | 1         |
| Splice        | 2     | 3     | 4           | 1         |
| Lung-cancer   | 4     | 1     | 3           | 2         |
| F-diagnosis   | 2.5   | 2.5   | 4           | 1         |
| Breast-cancer | 2.5   | 2.5   | 4           | 1         |
| Bupa          | 3     | 4     | 2           | 1         |
| Pima          | 2     | 3     | 4           | 1         |
| Housing       | 2.5   | 2.5   | 2.5         | 2.5       |
| Average rank  | 2.61  | 2.53  | 3.53        | **1.33**  |

6. **Conclusions.** In this paper, for the first time, we used the AL method to solve the TBSVM and obtained a new unconstrained minimization problem formulation. That is, we proposed an unconstrained AL method as well as a smoothing technique for solving the constrained TBSVM. In AL-STBSVM, we solved two primal problems using Newton’s algorithm instead of two dual problems in TBSVM. Also,
AL-STBSVM was extended to nonlinear separation surfaces using the nonlinear Gaussian kernel technique. The experimental results obtained from several UCI data sets as well as the generated data set showed that AL-STBSVM outperformed $I_{\nu}$-TBSVM, TBSVM and TWSVM methods in both linear and nonlinear states. We have used UCI data set which are real data's. UCI data sets are not structurally separated and it makes a lot of sense that the nonlinear state performance better on this data. Because the structure of nonlinear classifiers often performs better data classification in all methods. So, in this proposed method, the performance is better in nonlinear classifiers compared with linear. A classic statistical analysis method was presented that could be used to analyze the accuracy of linear classifiers on UCI benchmark data sets. Using Friedman test, the performance of AL-TBSVM method was compared with other methods. As can be seen, AL-STBSVM method has an excellent performance in both linear and nonlinear states. There are several directions of research that could be pursued in future studies. For example, since AL-STBSVM has been shown to be only effective for a two-class problem to date, AL-STBSVM could be extended to a multi-class problem, which is a promising field for investigating the application of a multi-class AL-STBSVM in the real world.

Appendix A. Matlab Code.

```matlab
% Generate random M,N.
% Input: m1,m2 n; Output: M N
pl=inline('(abs(x)+x)/2');
M=rand(m1,n); M=100*(M-0.5*spones(M));
M(:,2)=M(:,1)+1*ones(m1,1)+100*rand(m1,1)+100*rand(m1,1);
N=rand(m2,n); N=100*(N-0.5*spones(N));
N(:,2)=N(:,1)-1*ones(m2,1)-100*rand(m2,1)-100*rand(m2,1);
uu=5*rand(3,n); uu1=uu;uu1(:,2)= uu1(:,1)+1*ones(3,1);
uu2=uu;uu2(:,2)= uu2(:,1)-1*ones(3,1);
M=[M;uu1;10 0]; N=[N;uu2;30 -20];
xM=[-50:40*rand: 50];yM=xM+1;
xN=[-50:20*rand:50];yN=xN-1;
plot(M(:,2),M(:,1),’oblack’,N(:,2),N(:,1),’*bl’);
axis square
format short ;[m1 m2 n toc],[max(M(:,1)) min(N(:,1))]
```

REFERENCES

[1] Z. Arabasadi, R. Alizadehsani, M. Roshanzamir, H. Moosaei and A. Yarifard, Computer aided decision making for heart disease detection using hybrid neural network-Genetic algorithm, *Computer Methods and Programs in Biomedicine*, **141** (2017), 19–26.

[2] F. Bazikar, S. Ketabchi and H. Moosaei, DC programming and DCA for parametric-margin $\nu$—support vector machine, *Applied Intelligence*, (2020), 1–12.

[3] D. P. Bertsekas, *Nonlinear Programming*, Belmont, 1995.

[4] E. G. Birgin and J. M. Martínez, *Practical Augmented Lagrangian Methods for Constrained Optimization*, Society for Industrial and Applied Mathematics, 2014.

[5] C. C. Chang and C. J. Lin, LIBSVM: A library for support vector machines, *ACM transactions on intelligent systems and technology (TIST)*, **2** (2011), 1–27.

[6] C. Chen and O. L. Mangasarian, A class of smoothing functions for nonlinear and mixed complementarity problems, *Computational Optimization and Applications*, **5** (1996), 97–138.

[7] C. Chen and O. L. Mangasarian, Smoothing methods for convex inequalities and linear complementarity problems, *Mathematical Programming*, **71** (1995), 51–69.

[8] J. Demsar, Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research*, (2006), 1–30.
SMOOTH AUGMENTED LAGRANGIAN METHOD

[9] A. Goli, E. Moeini, A. M. Shafiee, M. Zamani and E. Touti, Application of improved artificial intelligence with runner-root meta-heuristic algorithm for dairy products industry: a case study, *International Journal on Artificial Intelligence Tools*, **29** (2020), 1–30.

[10] A. Goli, H. K. Zare, R. T. Moghaddam and A. Sadeghieh, An improved artificial intelligence based on gray wolf optimization and cultural algorithm to predict demand for dairy products: a case study, *IJIMAI*, **5** (2019), 15–22.

[11] A. Goli, H. K. Zare, R. Tavakkoli-Moghaddam and A. Sadeghieh, Application of robust optimization for a product portfolio problem using an invasive weed optimization algorithm, *Numerical Algebra, Control & Optimization*, **9** (2019), 187–209.

[12] A. Goli, H. K. Zare, R. Tavakkoli-Moghaddam and A. Sadeghieh, Hybrid artificial intelligence and robust optimization for a multi-objective product portfolio problem Case study: The dairy products industry, *Computers & Industrial Engineering*, **9** (2019), ID: 106090.

[13] R. L. Iman and J. M. Davenport, Approximations of the critical region of the Whitney statistic, *Communications in Statistics-Theory and Methods*, **9** (1980), 571–595.

[14] S. Jafarian-Namin, A. Goli, M. Qolipour, M. Mostafaeipour and A. M. Golmohammadi, Forecasting the wind power generation using Box-Jenkins and hybrid artificial intelligence, *International Journal of Energy Sector Management*, **13** (2019), 1038–1062.

[15] H. Javadi, H. Moosaei and D. Ciouzzo, Learning wireless sensor networks for source localization, *Sensors*, **19** (2019), 635.

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

[17] H. Ju, Q. Hou and L. Jing, Fuzzy and interval-valued fuzzy nonparallel support vector machine, *Journal of Intelligent and Fuzzy Systems*, **36** (2019), 2677–2690.

[18] S. Ketabchi and H. Moosaei, Minimum norm solution to the absolute value equation in the convex case, *Journal of Optimization Theory and Applications*, **154** (2012), 1080–1087.

[19] S. Ketabchi, H. Moosaei, M. Razzaghi and P. Pardalos, An improvement on parametric $\nu$-support vector algorithm for classification, *Ann. Oper. Res.*, **276** (2019), 155–168.

[20] S. Ketabchi and M. Behboodi-Kahoo, Smoothing techniques and augmented Lagrangian method for recourse problem of two-stage stochastic linear programming, *Journal of Applied Mathematics*, (2013), Article ID: 735916.

[21] S. Ketabchi and M. Behboodi-Kahoo, Augmented Lagrangian method for recourse problem of two-stage stochastic linear programming, *Kybernetika*, **49** (2013), 157–198.

[22] R. Khemchandani, P. Saigal and S. Chandra, Improvements on $\nu$-twin support vector machine, *Neural Networks*, **79** (2016), 97–107.

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

[24] Y. J. Lee and O. L. Mangasarian, SSVM: A smooth support vector machine for classification, *Computational Optimization and Applications*, **20** (2001), 5–22.

[25] H. Moosaei, S. Ketabchi, M. Razzaghi, and M. Tanveer, Generalized twin support vector machines, *Neural Processing Letters*, **53** (2021), 1545–1564.

[26] M. Lichman, *UCI Machine Learning Repository*, Irvine, CA: University of California, School of Information and Computer Science, 2013.

[27] W. Noble, S. William and others, Support vector machine applications in computational biology, *Kernel Methods in Computational Biology*, **71** (2004), 92.

[28] R. Lotfi, Y. Z. Mehrjerdi, M. S. Pishvaee, A. Sadeghieh and G. W. Weber, A robust optimization model for sustainable and resilient closed-loop supply chain network design considering conditional value at risk, *Numerical Algebra, Control and Optimization*, **11** (2021), 221–253.

[29] R. Lotfi, Z. Yadegari, S. H. Hosseini, A. H. Khameh, E. B. Tirkolaee and G. W. Weber, A robust time-cost-quality-energy-environment trade-off with resource-constrained in project management: A case study for a bridge construction project, *Journal of Industrial and Management Optimization*, Online.

[30] O. L. Mangasarian and E. W. Wild, Multisurface proximal support vector machine classification via generalized eigenvalues, *IEEE Trans. Pattern Anal. Mach. Intell.*, **28** (2005), 69–74.

[31] O. L. Mangasarian and E. W. Wild, Proximal support vector machine classifiers, In *Proceedings KDD-2001: Knowledge Discovery and Data Mining*, 2001.

[32] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer-Verlag, New York, 1999.
[33] X. Peng, A $\nu$-twin support vector machine ($\nu$-TSVM) classifier and its geometric algorithms, Information Sciences, 180 (2010), 3863–3875.

[34] C. Platt, Fast training of support vector machines using sequential minimal optimization, in Advances in Kernel Methods, (1999), 185–208.

[35] Y. H. Shao, C. H. Zhang, X. B. Wang and N. Y. Deng, Improvements on twin support vector machines, IEEE Transactions on Neural Networks, 22 (2011), 962–968.

[36] Y. Tian and Z. Qi, Review on: twin support vector machines, Annals of Data Science, 1 (2014), 253–277.

[37] Y. Tian, X. Ju, Z. Qi and Y. Shi, Improved twin support vector machine, Science China Mathematics, 57 (2014), 417–432.

[38] H. Wang, Z. Zhou and Y. Xu, An improved $\nu$-twin bounded support vector machine, Appl. Intell., 48 (2018), 1041–1053.

[39] Y. Wang, T. Wang and J. Bu, Color image segmentation using pixel wise support vector machine classification, Pattern Recognition, 44 (2011), 777–787.

[40] Y. Yan and Q. Li, An efficient augmented Lagrangian method for support vector machine, Optimization Methods and Software, 35 (2020), 855–883.

Received September 2020; 1st revision December 2020; Final revision June 2021; Early access July 2021.

E-mail address: f.bazikar@gmail.com, f_bazikar@phd.guilan.ac.ir
E-mail address: sketabchi@guilan.ac.ir
E-mail address: hmoosaei@gmail.com, moosaei@ub.ac.ir