Filomat 31:8 (2017), 2195–2210
DOI 10.2298/FIL1708195T

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Smooth Twin Support Vector Machines via Unconstrained Convex Minimization

M. Tanveer[a], K. Shubham[b]

[a] *Discipline of Mathematics, Indian Institute of Technology Indore, Indore 453 552 INDIA*
[b] *Department of Electronics and Communication Engineering, The LNM Institute of Information Technology, Jaipur 302 031 INDIA*

Abstract. Twin support vector machine (TWSVM) exhibits fast training speed with better classification abilities compared with standard SVM. However, it suffers the following drawbacks: (i) the objective functions of TWSVM are comprised of empirical risk and thus may suffer from overfitting and suboptimal solution in some cases. (ii) a convex quadratic programming problems (QPPs) need to be solve, which is relatively complex to implement. To address these problems, we proposed two smoothing approaches for an implicit Lagrangian TWSVM classifiers by formulating a pair of unconstrained minimization problems in dual variables whose solutions will be obtained by solving two systems of linear equations rather than solving two QPPs in TWSVM. Our proposed formulation introduces regularization terms to each objective function with the idea of maximizing the margin. In addition, our proposed formulation becomes well-posed model due to this term, which introduces invertibility in the dual formulation. Moreover, the structural risk minimization principle is implemented in our formulation which embodies the essence of statistical learning theory. The experimental results on several benchmark datasets show better performance of the proposed approach over existing approaches in terms of estimation accuracy with less training time.

1. Introduction

The foundation of support vector machines (SVMs) have been developed by Vapnik and coworkers [4, 8, 40], and are gaining popularity due to many attractive features, and promising empirical performance. This learning strategy introduced by Vapnik and co-workers [3] is a principled and very powerful method in machine learning algorithms. SVM has played an important role in solving problems emerged in pattern recognition and machine learning community over the past decades because of its novel state of art technique. Its applications include a wide spectrum of research areas, ranging from pattern recognition [26], text categorization [14], biomedicine [5] etc. The main idea of SVMs is to find the optimal hyperplane between positive and negative samples such that the margin can be maximized. SVMs have a solid theoretical foundation, rooted in statistical learning theory (SLT) and structural risk minimization (SRM) principle. One of the main challenges for SVM is the large computational complexity of QPP. To address this drawback, many algorithms in the recent past have been reported in [1, 6, 8, 15, 17, 18, 29, 37].

2010 *Mathematics Subject Classification*. Primary 60K05 (mandatory); Secondary 49M15 (optionally)
Keywords. Machine learning; Lagrangian support vector machines; Twin support vector machine; Smoothing techniques; Convex minimization.
Received: August 30, 2014; Revised: 28 December 2014; Accepted: 16 March 2015
Communicated by Predrag Stanimirović
Email addresses: tanveergouri@gmail.com; mtanveer@iiti.ac.in (M. Tanveer), kumarshubham652@gmail.com (K. Shubham)

In the last few years, the research on non-parallel SVMs have been an important and interesting approach where two non-parallel hyperplanes are constructed instead of constructing two parallel hyperplanes in traditional SVMs. Mangasarian and Wild [23] proposed a generalized eigenvalue proximal support vector machine (GEPSVM) which needs to solve generalized eigenvalue problems. Subsequently, Jayadeva et al. [13] proposed the twin support vector machine (TWSVM) in the light of GEPSVM. Different from GEPSVM, TWSVM solves two small SVM-type problems to obtain the hyperplanes. Experimental results show the strong generalization ability of TWSVM over SVM and GEPSVM [13]. Recently, many variants of TWSVM are proposed to reduce the time complexity and keep the effectiveness of TWSVM, see [2, 16, 28, 30–36, 38, 39]. Specifically, Kumar and Gopal [16] enhanced TWSVM using smoothing techniques proposed in [18], to Smooth TWSVM (STWSVM). The main objective of STWSVM is to improve the computational speed of TWSVM such that it can be used for large datasets. Further, Shao et al. [31] proposed twin bounded support vector machines (TBSVM) based on TWSVM. It also constructs two nonparallel hyperplanes by solving two smaller QPPs. But the difference is that the structural risk minimization principle is implemented by adding the regularization term into the primal problems of TBSVM. Two extra parameters introduced in the objective functions are the weights between the regularization term and empirical risk. In order to shorten the training time, SOR technique is applied to TBSVM. Computational results show that TBSVM is not only faster but also shows better generalization. To improve the robustness and sparseness, recently, Tanveer [33–35] proposed novel linear programming formulation of 1-norm twin support vector machine for classification and regression problems, whose solution is obtained, by solving a pair of exterior penalty problems in the dual space as unconstrained optimization problems using Newton-Armijo algorithm.

Motivated by the works of [11, 18, 21, 27, 31, 33, 34], we proposed in this paper two smoothing approaches for an implicit Lagrangian twin support vector machine (TWSVM) classifiers by formulating a pair of unconstrained minimization problems (UMPs) in dual variables whose solutions will be obtained using finite Newton method. Our formulation possesses the following attractive advantages:

- Unlike TWSVM and STWSVM, our proposed formulation introduces regularization terms to each objective function with the idea of maximizing the margin. In addition, our proposed formulation becomes well-posed model due to adding extra regularization term which introduces invertibility in the dual formulation.
- Note that the 2-norm of the slack variables is minimized in our formulation instead of 1-norm as in TWSVM and TBSVM, to make the objective functions strongly convex. It implies the existence of global optimal solution.
- Unlike TWSVM and STWSVM, the structural risk minimization principle is implemented in our formulation which embodies the essence of statistical learning theory.
- Two smoothing techniques are proposed whose solution is obtained by solving two systems of linear equations rather than solving two QPPs in TWSVM.
- The experimental results on several benchmark datasets exhibit excellent performance of our formulation over existing approaches in terms of estimation accuracy with less training time.

The paper is organized as follows: we briefly review the TWSVM formulation in Section 2. Section 3 describe the details of our proposed method. Numerical experiments have been performed on a number of interesting synthetic and real-world benchmark datasets and their results are compared with other SVMs in Section 4, finally we conclude our work in Section 5.

2. Brief Review of Twin Support Vector Machines

The main idea of TWSVM [13] is to generate two nonparallel hyperplanes instead of a single hyperplane in the standard SVM to classify samples. The two nonparallel hyperplanes are obtained by solving two smaller sized QPPs compared with a single large QPP solved by the standard SVM. This makes TWSVM faster than standard SVM.

Suppose that all the data samples in class +1 are denoted by a matrix $A \in R^{m_1 \times n}$ and the matrix $B \in R^{m_2 \times n}$ represent the data samples of class -1. For the linear case, the TWSVM seeks a pair of non-parallel hyperplanes

$$f_1(x) = w_1^t x + b_1 \quad \text{and} \quad f_2(x) = w_2^t x + b_2, \tag{1}$$

such that each hyperplane is proximal to the data points of one class and far from the data points of other class, where $w_1 \in R^n, w_2 \in R^n, b_1 \in R$ and $b_2 \in R$. The formulation of TWSVM can be written as follows:

$$\begin{aligned} \min_{(w_1,b_1)\in R^{n+1}} \quad & \frac{1}{2}\|Aw_1 + e_2 b_1\|^2 + C_1\|\xi_1\| \\ s.t. \quad & -(Bw_1 + e_1 b_1) + \xi_1 \geq e_1,\ \xi_1 \geq 0, \end{aligned} \tag{2}$$

$$\begin{aligned} \min_{(w_2,b_2)\in R^{n+1}} \quad & \frac{1}{2}\|Bw_2 + e_1 b_2\|^2 + C_2\|\xi_2\| \\ s.t. \quad & (Aw_2 + e_2 b_2) + \xi_2 \geq e_2,\ \xi_2 \geq 0, \end{aligned} \tag{3}$$

where C_1, C_2 are positive parameters, ξ_1, ξ_2 are slack variables and e_1, e_2 are vectors of one of appropriate dimensions. It is evident that the idea in TWSVM is to solve two QPPs (2) and (3), each of the QPPs in the TWSVM pair is a typical SVM formulation, except that not all data points appear in the constraints of either problem [13].

By introducing the Lagrangian vectors α and γ, the Wolfe duals of (2) and (3) are

$$\begin{aligned} \min_{\alpha\in R^{m_2}} \quad & \frac{1}{2}\alpha^t H\left(G^t G\right)^{-1} H^t \alpha - e_1^t \alpha \\ s.t. \quad & 0 \leq \alpha \leq C_1, \end{aligned} \tag{4}$$

and

$$\begin{aligned} \min_{\gamma\in R^{m_1}} \quad & \frac{1}{2}\gamma^t G\left(H^t H\right)^{-1} G^t \gamma - e_2^t \gamma \\ s.t. \quad & 0 \leq \gamma \leq C_2, \end{aligned} \tag{5}$$

where $G = [A \ \ e_2]$ and $H = [B \ \ e_1]$ are augmented matrices of sizes $m_1 \times (n+1)$ and $m_2 \times (n+1)$ respectively.

In order to deal with the case when $G^t G$ or $H^t H$ are singular and avoid the possible ill conditioning, the inverse matrices $(G^t G)^{-1}$ and $(H^t H)^{-1}$ are approximately replaced by $(G^t G + \delta I)^{-1}$ and $(H^t H + \delta I)^{-1}$, where δ is a very small positive scalar and I is an identity matrix of appropriate dimensions. Thus the nonparallel proximal hyperplanes are obtained from the solution $\alpha \in R^{m_2}$ and $\gamma \in R^{m_1}$ of (4) and (5) by

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -\left(G^t G + \delta I\right)^{-1} H^t \alpha \quad \text{and} \quad \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = \left(H^t H + \delta I\right)^{-1} G^t \gamma. \tag{6}$$

3. Smooth Twin Support Vector Machines via Unconstrained Convex Minimization

In this section, a new variant of the TWSVM in its dual is proposed as a pair of implicit UMPs and their solutions are computed by applying two popular smoothing approaches to improve the robustness. We construct two nonparallel proximal hyperplanes

$$f_1(x) = {w_1}^t x + b_1 = 0 \text{ and } f_2(x) = {w_2}^t x + b_2 = 0, \tag{7}$$

by considering the following 2-norm regularized TWSVM formulation of the form:

$$\begin{aligned} \min_{w_1 \in R^n, b_1 \in R} \quad & \frac{1}{2}\,\|Aw_1 + e_2 b_1\|^2 + \frac{C_1}{2}\,\|\xi_1\|^2 + \frac{C_3}{2}\left\|\begin{bmatrix} w_1 \\ b_1 \end{bmatrix}\right\|^2 \\ s.t. \quad & -(Bw_1 + e_1 b_1) + \xi_1 \geq e_1, \end{aligned} \tag{8}$$

$$\begin{aligned} \min_{w_2 \in R^n, b_2 \in R} \quad & \frac{1}{2}\,\|Bw_2 + e_1 b_2\|^2 + \frac{C_2}{2}\,\|\xi_2\|^2 + \frac{C_4}{2}\left\|\begin{bmatrix} w_2 \\ b_2 \end{bmatrix}\right\|^2 \\ s.t. \quad & (Aw_2 + e_2 b_2) + \xi_2 \geq e_2, \end{aligned} \tag{9}$$

where C_1, C_2, C_3, C_4 are all positive trade-off constants.

Let us first discuss the differences between our proposed model, TWSVM, TBSVM and STWSVM.

- Unlike TWSVM and STWSVM, our proposed formulation introduces regularization terms to each objective function with the idea of maximizing the margin. In addition, our proposed formulation becomes well-posed model due to adding extra regularization term which introduces invertibility in the dual formulation.
- Note that the 2-norm of the slack variables is minimized in our formulation instead of 1-norm as in TWSVM and TBSVM, to make the objective functions strongly convex. It implies the existence of global optimal solution.
- Unlike TWSVM and STWSVM, the structural risk minimization principle is implemented in our formulation which embodies the essence of statistical learning theory.
- Two smoothing techniques are proposed whose solution is obtained by solving two systems of linear equations rather than solving two QPPs in TWSVM and TBSVM.
- The experimental results on several benchmark datasets exhibit excellent performance of our formulation over existing approaches in terms of estimation accuracy with less training time.

By considering the Lagrangian functions corresponding to (8) and (9), and using the conditions that their partial derivatives with respect to the primal variables will be zero at optimality, the dual QPPs of (8) and (9) can be obtained by dropping the terms which are independent of the dual variables as a pair of minimization problems of the following form:

$$\min_{0 \leq u_1 \in R^{m_2}} \frac{1}{2} {u_1}^t \left(\frac{I}{C_1} + H(G^t G + C_3 I)^{-1} H^t\right) u_1 - {e_1}^t u_1, \tag{10}$$

$$\min_{0 \leq u_2 \in R^{m_1}} \frac{1}{2} {u_2}^t \left(\frac{I}{C_2} + G(H^t H + C_4 I)^{-1} G^t\right) u_2 - {e_2}^t u_2, \tag{11}$$

where $u_1 \in R^{m_2}, u_2 \in R^{m_1}$ are Lagrange multipliers; $G = [A \; e_2]$ and $H = [B \; e_1]$ are augmented matrices of sizes $m_1 \times (n+1)$ and $m_2 \times (n+1)$ respectively.
Define the matrices

$$M_1 = H(G^t G + C_3 I)^{-1} H^t \quad \text{and} \quad M_2 = G(H^t H + C_4 I)^{-1} G^t. \tag{12}$$

The dual QPPs (10) and (11) can be rewritten as a pair of minimization problems of the form:

$$\min_{0 \leq u_1 \in R^{m_2}} \frac{1}{2} {u_1}^t Q_1 u_1 - {e_1}^t u_1, \tag{13}$$

$$\min_{0\le u_2\in R^{m_1}} \frac{1}{2}{u_2}^t Q_2 u_2 - {e_2}^t u_2, \tag{14}$$

where

$$Q_1 = \frac{I}{C_1} + M_1 \text{ and } Q_2 = \frac{I}{C_2} + M_2. \tag{15}$$

The nonparallel proximal hyper planes are obtained from the solution u_1 and u_2 of (13) and (14) by

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -(G^tG + C_3I)^{-1}H^tu_1 \text{ and } \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (H^tH + C_4I)^{-1}G^tu_2. \tag{16}$$

Remark 3.1. *One can immediately notice that our proposed approach does not need to care about matrix singularity for finding the solution. It is worthwhile to note that the regularization parameter δ used in TWSVM formulation is just a fixed small scalar while penalty parameters C_3, C_4 used in our formulation are weighting factors which determine the trade-off between the regularization term and the empirical risk. Therefore, selecting appropriate parameters C_3, C_4 reflects the structural risk minimization principle. The experimental results in Section 4 shows the improve classification accuracy on adjusting the values of C_3, C_4.*

We cannot always handle classification problems using linear kernel. Therefore, we extend our results to nonlinear classifiers by considering the following kernel based surfaces instead of hyperplanes:

$$K(x^t, C^t)w_1 + b_1 = 0 \text{ and } K(x^t, C^t)w_2 + b_2 = 0, \tag{17}$$

where $C^t = [A\ B]^t$ and K is appropriately chosen kernel.

The optimization problem for our robust 2-norm regularized TWSVM in the kernel feature space can be reformulated as:

$$\begin{aligned} \min_{w_1\in R^m, b_1\in R} \quad & \tfrac{1}{2}\left\|K(A,C^t)w_1 + e_2b_1\right\|^2 + \tfrac{C_1}{2}\|\xi_1\|^2 + \tfrac{C_3}{2}\left\|\begin{bmatrix} w_1 \\ b_1 \end{bmatrix}\right\|^2 \\ s.t. \quad & -(K(B,C^t)w_1 + e_1b_1) + \xi_1 \ge e_1, \end{aligned} \tag{18}$$

$$\begin{aligned} \min_{w_2\in R^m, b_2\in R} \quad & \tfrac{1}{2}\left\|K(B,C^t)w_2 + e_1b_2\right\|^2 + \tfrac{C_2}{2}\|\xi_2\|^2 + \tfrac{C_4}{2}\left\|\begin{bmatrix} w_2 \\ b_2 \end{bmatrix}\right\|^2 \\ s.t. \quad & (K(A,C^t)w_2 + e_2b_2) + \xi_2 \ge e_2, \end{aligned} \tag{19}$$

where $K(A, C^t)$ and $K(B, C^t)$ are kernel matrices of sizes $m_1 \times m$ and $m_2 \times m$ respectively, where $m = m_1 + m_2$. By defining the augmented matrix $S = [K(A, C^t)\ e_2]$, $R = [K(B, C^t)\ e_1]$ and proceeding entirely similar process to the linear case, the pair of minimization problems (18) and (19) can be converted into nonlinear 2-norm regularized TWSVM problems again of the same form (13) and (14). Thus the nonparallel proximal hyper planes are obtained from the solution u_1 and u_2 of (13) and (14) by

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -(S^tS + C_3I)^{-1}R^tu_1 \text{ and } \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = (R^tR + C_4I)^{-1}S^tu_2. \tag{20}$$

3.1. Method of solution

Applying the Karush-Kuhn-Tucker (KKT) necessary and sufficient optimal conditions for the dual 2-norm robust TWSVM will lead to the following pair of classical complementarity problems [19]:

$$0 \le u_1 \perp (Q_1u_1 - e_1) \ge 0 \text{ and } 0 \le u_2 \perp (Q_2u_2 - e_2) \ge 0. \tag{21}$$

However, using the well-known identity between two vectors u, v :
$0 \le u \perp v \ge 0$ if and only if $u = (u - \alpha v)_+$ for any $\alpha \ge 0$. The solutions of the following equivalent pair of problems will be considered [21]: for any $\alpha_1, \alpha_2 > 0$,

$$(Q_1u_1 - e_1) = (Q_1u_1 - \alpha_1u_1 - e_1)_+ \text{ and } (Q_2u_2 - e_2) = (Q_2u_2 - \alpha_2u_2 - e_2)_+. \tag{22}$$

It turns out that (22) become necessary and sufficient conditions to be satisfied by the unconstrained minimum of the following pair of implicit Lagrangian's [22] associated to the pair of dual problems (13) and (14): for any $\alpha_1, \alpha_2 > 0$,

$$\min_{u_1 \in R^{m_2}} L_1(u_1) = \frac{1}{2} {u_1}^t Q_1 u_1 - {e_1}^t u_1 + \frac{1}{2\alpha_1} \left(\|(Q_1 u_1 - \alpha_1 u_1 - e_1)_+\|^2 - \|Q_1 u_1 - e_1\|^2 \right), \tag{23}$$

and

$$\min_{u_2 \in R^{m_1}} L_2(u_2) = \frac{1}{2} {u_2}^t Q_2 u_2 - {e_2}^t u_2 + \frac{1}{2\alpha_2} \left(\|(Q_2 u_2 - \alpha_2 u_2 - e_2)_+\|^2 - \|Q_2 u_2 - e_2\|^2 \right). \tag{24}$$

One can obtain the gradient of (23) and (24) as:

$$\nabla L_k(u_k) = \left(\frac{\alpha_k I - Q_k}{\alpha_k} \right) \left[(Q_k u_k - e_k) - (Q_k u_k - \alpha_k u_k - e_k)_+ \right], \; k = 1, 2. \tag{25}$$

Since the gradient $\nabla L_k(u_k)$ is not differentiable and therefore the Hessian matrix of second order partial derivatives of $L_k(u_k)$ is not defined in the usual sense. It is proposed to introduce two smoothing approaches, studied in [16, 18, 27, 33].

3.1.1. Smooth Approach I (SNTSVM-1)

Smoothing techniques are enormously used for solving many optimization problems, such as SSVM [18], RSVM [17], STWSVM [16], SLPTSVM [33], STSVR [7], PTSVR [27] and others.

First, we consider the following smooth approximation function, denoted by $p_1(x, \eta)$ for x_+ with parameter $\eta > 0$, defined as [17, 18, 33]: for any real value x,

$$p_1(x, \eta) = x + \frac{1}{\eta} \log\left(1 + \exp(-\eta x)\right). \tag{26}$$

Infact, for the vectors $p_1(u, \eta) = (p_1(u_1, \eta), ..., p_1(u_m, \eta))^t$, the pair of dual UMPs (23) and (24) will get modified into

$$\min_{u_1 \in R^{m_2}} L_1(u_1) = \frac{1}{2} {u_1}^t Q_1 u_1 - {e_1}^t u_1 + \frac{1}{2\alpha_1} \left(\|p_1\left((Q_1 u_1 - \alpha_1 u_1 - e_1), \eta_1\right)\|^2 - \|Q_1 u_1 - e_1\|^2 \right), \tag{27}$$

and

$$\min_{u_2 \in R^{m_1}} L_2(u_2) = \frac{1}{2} {u_2}^t Q_2 u_2 - {e_2}^t u_2 + \frac{1}{2\alpha_2} \left(\|p_1\left((Q_2 u_2 - \alpha_2 u_2 - e_2), \eta_2\right)\|^2 - \|Q_2 u_2 - e_2\|^2 \right), \tag{28}$$

respectively, where $\eta_1, \eta_2 > 0$, u_1 and u_2 are the solutions of the minimization problems (27) and (28) respectively.

Since the modified minimization problems (27) and (28) are smooth, one can obtain the Hessian matrix of $L_k(.,.)$ as:

$\nabla^2 L_k(u_k) = \left(\frac{\alpha_k I - Q_k}{\alpha_k} \right) \left[(I - D_k) Q_k + \alpha_k D_k \right]$,

where $D_k = 1 / \left(1 + e^{-\alpha_k (Q_k u_k - \alpha_k u_k - e_k)} \right)$.

Now we summarize the Newton algorithm for solving UMPs (27) and (28) for an arbitrary positive definite matrix $Q_k, k = 1, 2$.

Algorithm 1 (SNTSVM-1). For solving pair of UMPs (27) and (28) with $k = 1, 2$:
Input

- Set the parameters $C > 0, \epsilon > 0$ and $\alpha_k = 1.9$.
- *tol*=error tolerance for learning accuracy, *itmax*=maximum number of iterations.

- $Q = \frac{I}{C} + M$.
- $i = 0, u = u^0$ (initial guess)

Step 1.

- compute $D_k = 1/\left(1 + e^{-\alpha_k(Q_k u_k - \alpha_k u_k - e_k)}\right)$

Step 2.

- $\|u_k^{i+1} - u_k^i\| < tol$ or $i < itmax$
- calculate $\nabla^2 L_k u_k^i$, $\nabla L_k u_k^i$
- compute $u_k^{i+1} = u_k^i$ - $(\nabla^2 L_k u_k^i)^{-1}$ $\nabla L_k u_k^i$
- $i = i + 1$.

3.1.2. Smooth Approach II (SNTSVM-2)

For solving the pair of dual UMPs (23) and (24), we use the following another smooth approximation function $p_2(x, x_0)$ for x_+, defined [27, 33] as: for any real value x,

$$p_2(x, x_0) = \frac{1}{4}\frac{x^2}{|x_0|} + \frac{1}{2}x + \frac{1}{4}|x_0|, \tag{29}$$

where x_0 is a non-zero real number. Notice that $p_2(x, x_0)$ is a quadratic function and twice differentiable. Also, the value of $p_2(x, x_0)$ is closer to x_+ when the value of $|x_0|$ is closer to $|x|$. Specifically, $p_2(x, x_0) = x_+$ when $|x_0| = |x| \neq 0$.

By replacing u_+ by $p_2(u, u_0)$, the pair of dual UMPs (23) and (24) will get modified into

$$\min_{u_1 \in R^{m_2}} L_1(u_1) = \frac{1}{2}{u_1}^t Q_1 u_1 - {e_1}^t u_1 + \frac{1}{2\alpha_1}\left(\|p_2\left((Q_1 u_1 - \alpha_1 u_1 - e_1), u_0\right)\|^2 - \|Q_1 u_1 - e_1\|^2\right) \tag{30}$$

and

$$\min_{u_2 \in R^{m_1}} L_2(u_2) = \frac{1}{2}{u_2}^t Q_2 u_2 - {e_2}^t u_2 + \frac{1}{2\alpha_2}\left(\|p_2\left((Q_2 u_2 - \alpha_2 u_2 - e_2), u_0\right)\|^2 - \|Q_2 u_2 - e_2\|^2\right), \tag{31}$$

respectively, where the components of $u_0 \in R^n$ are non-zero, u_1 and u_2 are the solutions of the minimization problems (30) and (31) respectively.

Since the modified minimization problems (30) and (31) are smooth, one can obtain the Hessian matrix of $L_k(.,.)$ as:
$\nabla^2 L_k(u_k) = \left(\frac{\alpha_k I - Q_k}{\alpha_k}\right)\left[(I - D_k)\, Q_k + \alpha_k D_k\right]$,
where $D_k = \frac{1}{2}\left[\frac{Q_k u_k - \alpha_k u_k - e_k}{|u_0|} + 1\right]$.

Remark 3.2. *Notice that the Hessian matrix of $L_k, k = 1, 2$ is positive definite. Therefore, the pair of UMPs (30) and (31) will have a unique, global solution at its extreme point [27, 33].*

Now we summarize the Newton algorithm for solving UMPs (30) and (31) for an arbitrary positive definite matrix $Q_k, k = 1, 2$.

Algorithm 2 (SNTSVM-2). For solving pair of UMPs (30) and (31) with $k = 1, 2$:
Input

- Set the parameters $C > 0, \epsilon > 0, \sigma = 10^{-6}$ and $\alpha_k = 1.9$.

- *tol*=error tolerance for learning accuracy, *itmax*=maximum number of iterations.
- $Q = \frac{I}{C} + M$.
- $i = 0, u = u^0$ (initial guess)

Step 1.

- compute $D_k = \frac{1}{2}\left[\text{diag}(\|u_0^i\|)^{-1}(Q_k u_k^i - \alpha_k u_k^i - e_k) + 1\right]$

Step 2.

- if $\|u_k^{i+1} - u_k^i\| < tol$ or $i < itmax$
- compute ${u_0}^i = (Q_k u_k^i - \alpha_k u_k^i - e_k) + \sigma e$ where + is defined as a+b=a if a≠0; otherwise b.
- calculate $\nabla^2 L_k u_k^i$, $\nabla L_k u_k^i$
- compute $u_k^{i+1} = u_k^i$ - $(\nabla^2 L_k u_k^i)^{-1}$ $\nabla L_k u_k^i$
- $i = i + 1$.

For $k = 1, 2$, the basic Newton's step of iterative algorithm is in determining the unknown u^{i+1} at the $(i + 1)^{th}$ iteration using the current i^{th} iterate u^i using

$$\nabla L_k(u_k^i) + \nabla^2 L_k \left(u_k^{i+1} - u_k^i\right) = 0, \text{ where } i = 0, 1, 2, ... \quad (32)$$

The convergence of the above algorithm and its finite termination are derived in [20].

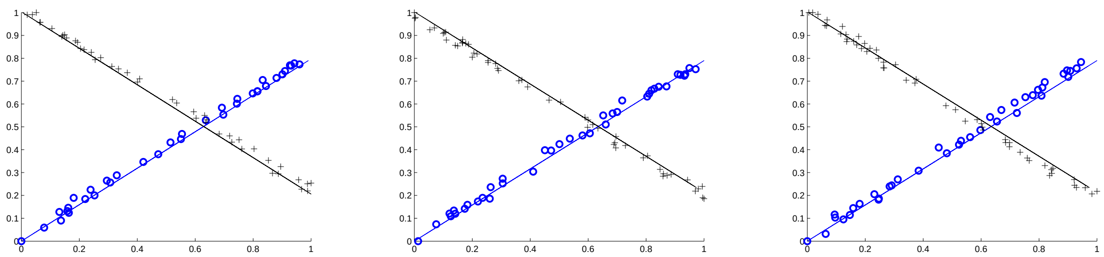
Remark 3.3. *Our SNTSVM-1 and SNTSVM-2 solve two systems of linear equations rather than solving two quadratic programming problems in TWSVM, which makes the learning speed extremely fast than TWSVM.*

4. Experimental Results

In this section, we performed numerical experiments to demonstrate the effectiveness of our proposed SNTSVM-1 and SNTSVM-2 in comparison to GEPSVM and TWSVM on 'Cross-Planes' dataset as an example of synthetic dataset and several well-known, publicly available, benchmark datasets [25]. All the classifiers are implemented in MATLAB R2008b environment on a PC with 3.30 GHz Intel(R) Core(TM) i3 processor having 4 GB RAM. The pair of QPP involved in GEPSVM and TWSVM are solved using optimization toolbox of MATLAB. In order to construct nonlinear classifier, Gaussian kernel function with parameter $\mu > 0$, defined by: for $x_1, x_2 \in R^m$, $K(x_1, x_2) = \exp(-\mu\|x_1 - x_2\|^2)$ is used. The classification accuracy of each algorithm was computed using the well-known ten-fold cross-validation methodology [10]. The optimal values of the parameters were chosen by the grid search method [12], which is the commonly used method in this field. Furthermore, to degrade the computational cost of parameter selection, in our experiments, we set $C_1 = C_2$ for TWSVM, $C_1 = C_2, C_3 = C_4$ for SNTSVM-1 and SNTSVM-2, and the kernel parameter value μ were allowed to vary from the sets $\{10^{-5}, 10^{-4}, ..., 10^{5}\}$ and $\{2^{-10}, 2^{-9}, ..., 2^{10}\}$ respectively. For GEPSVM, the range of δ for linear and Gaussian kernel were allowed to vary from the sets $\{10^{-10}, 10^{-4}, ..., 10^{10}\}$ and $\{2^{-7}, 2^{-6}, ..., 2^{7}\}$ respectively. The value of the smooth parameter $\eta = 5$ was set in SNTSVM-1 due to its successful results in [18, 33]. For SNTSVM-2, the regularized parameter σ should be very small having the property that the diagonal matrix in Step 1 of Algorithm 2 should be invertible for all the datasets, its value is taken to be $1e - 6$. Finally, choosing these optimal values, the classification accuracy and computational efficiency are adopted to measure the performances of these algorithms, and the best classification accuracy is shown by bold figures.

4.1. *Toy Example*

We consider a simple two dimensional "Cross Planes" dataset as an example of synthetic dataset which was also tested in [23, 31, 33]. It was generated by perturbing points lying on two intersecting lines and the intersection point is not in the center. Fig. 1(a-c) shows the dataset and the linear classifiers obtained by TWSVM and our proposed SNTSVM-1 and SNTSVM-2. One can easily observe that the result of our proposed SNTSVM-1 and SNTSVM-2 are better than TWSVM. This clearly indicate that our proposed method can handle the "Cross Planes" dataset much better than TWSVM. The average results of GEPSVM, TWSVM, SNTSVM-1 and SNTSVM-2 for linear and nonlinear classifiers are reported in Table 1 and Table 2. The results demonstrate the superior performance of SNTSVM-1 and SNTSVM-2.



(a) TWSVM (b) SNTSVM-1 (c) SNTSVM-2

Figure 1: Classification results of (a) TWSVM (b) SNTSVM-1 (c) SNTSVM-2 for Cross Planes dataset.

4.2. *Real-world benchmark datasets*

In this sub-section, we performed numerical experiments linear and non-linearly to demonstrate the performance of the proposed SNTSVM-1 and SNTSVM-2 in comparison to GEPSVM and TWSVM on 11 UCI datasets [25], some of which are used in [13, 31, 33]. In all the real-world examples considered, each attribute of the original data is normalized as follows:

$$\bar{x_{ij}} = \frac{x_{ij} - x_j^{\text{min}}}{x_j^{\text{max}} - x_j^{\text{min}}},$$

where x_{ij} is the (i,j)-th element of the input matrix A, $\bar{x_{ij}}$ is its corresponding normalized value and $x_j^{\text{min}} = \min_{i=1}^{m}(x_{ij})$ and $x_j^{\text{max}} = \max_{i=1}^{m}(x_{ij})$ denote the minimum and maximum values, respectively, of the j-th column of A. Each dataset is randomly split into testing and training. The specific number of training and testing samples, the number of attributes, training time and accuracies of each algorithm for linear and nonlinear classifiers are summarized in Table 1 and Table 4 respectively. According to Table 1, it can be found that our SNTSVM-1 and SNTSVM-2 show better generalization performance and computational speed. For Heart dataset, the experimental result (accuracy 81.43%) by our SNTSVM-1 and SNTSVM-2 outperform other two algorithms i.e., TWSVM (accuracy 55.71%) and GEPSVM (accuracy 68.57%). We obtain the similar conclusions for Ionosphere, Votes, Wpbc, Monks-3, Cleve, Australian, Haberman, Splice and Tic-Tac-Toe datasets. For WDBC dataset, the classification accuracy obtained by our SNTSVM-1 and SNTSVM-2 (76.81%) are lower than TWSVM (94.20%) and GEPSVM (92.75%) but computationaly faster than TWSVM. One can observe from Table 1 that SNTSVM-1 and SNTSVM-2 show exactly the same accuracy on all the datasets considered. This clearly indicates that both the approaches are equally preferred.

Table 1: Performance comparisons of GEPSVM, TWSVM, SNTSVM-1 and SNTSVM-2 on twelve datasets using linear kernel

Datasets (Train size, Test size)	GEPSVM Accuracy(%) Time(s)	TWSVM Accuracy(%) Time(s)	SNTSVM-1 Accuracy(%) Time(s)	SNTSVM-2 Accuracy(%) Time(s)
Cross Planes	84.42	91.42	**100.00**	**100.00**
(81×2, 40×2)	0.0468	0.5821	0.0081	0.0170
Heart-Statlog	68.57	55.71	**81.43**	**81.43**
(200×3, 106×3)	0.1560	0.2218	0.0034	0.0136
WDBC	92.75	**94.20**	76.81	76.81
(500×30, 69×30)	0.0312	0.4401	0.1926	0.1384
Ionosphere	70.48	79.05	**84.76**	**84.76**
(246×34, 105×34)	0.2964	1.1114	0.0420	0.0300
Votes	90.70	95.35	**96.12**	**96.12**
(306×16, 129×16)	0.0312	0.2197	0.0123	0.0200
WPBC	**77.19**	57.89	**77.19**	**77.19**
(137×33, 57×33)	0.0468	0.1800	0.0092	0.0143
Cleve	55.83	75.83	**82.50**	**82.50**
(177×13, 120×13)	0.0624	0.2469	0.0024	0.0507
Monks-3	63.65	75.69	**81.94**	**81.94**
(432×7, 122×7)	0.0468	0.2093	0.0064	0.0121
Australian	68.66	46.67	**90.66**	**90.66**
(540×14, 150×14)	0.0468	1.7045	0.0964	0.1643
Haberman	**76.42**	74.52	**76.42**	**76.42**
(200×3, 106×3)	0.0780	0.1094	0.0034	0.01098
Splice	80.60	80.00	**83.29**	**83.29**
(500×60, 2675×60)	0.1716	0.5702	0.2325	0.1193
Tic-Tac-Toe	56.09	**94.43**	**94.43**	**94.43**
(671×9, 287×9)	0.1248	1.0732	0.0878	0.2284

Table 2: Optimal parameters of GEPSVM, TWSVM, SNTSVM-1 and SNTSVM-2 for linear kernel

Datasets	GEPSVM	TWSVM	SNTSVM-1		SNTSVM-2	
	δ	$C_1 = C_2$	$C_1 = C_2$	$C_3 = C_4$	$C_1 = C_2$	$C_3 = C_4$
Cross Planes	10^4	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}
Heart-Statlog	10^1	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}
WDBC	10^{-5}	10^{-5}	10^{-1}	10^{-1}	10^{-1}	10^{-1}
Ionosphere	10^5	10^{-5}	10^{-2}	10^{-2}	10^{-2}	10^{-2}
Votes	10^2	10^{-5}	10^{-5}	10^0	10^{-5}	10^0
WPBC	10^{-2}	10^{-5}	10^1	10^{-1}	10^1	10^{-1}
Cleve	10^1	10^{-5}	10^{-5}	10^0	10^{-5}	10^0
Monks-3	10^0	10^1	10^4	10^1	10^3	10^1
Australian	10^1	10^{-5}	10^0	10^{-1}	10^0	10^{-1}
Haberman	10^{-5}	10^3	10^{-5}	10^1	10^{-5}	10^1
Splice	10^3	10^{-5}	10^1	10^3	10^1	10^3
Tic-Tac-Toe	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}	10^{-5}

Table 3: Average ranks of GEPSVM, TWSVM, SNTSVM-1 and SNTSVM-2 with linear kernel

Datasets	GEPSVM	TWSVM	SNTSVM-1	SNTSVM-2
Cross Planes	4	3	1.5	1.5
Heart-Statlog	3	4	1.5	1.5
WDBC	2	1	3.5	3.5
Ionosphere	4	3	1.5	1.5
Votes	4	3	1.5	1.5
WPBC	2	4	2	2
Cleve	4	3	1.5	1.5
Monks-3	4	3	1.5	1.5
Australian	3	4	1.5	1.5
Haberman	2	4	2	2
Splice	3	4	1.5	1.5
Tic-Tac-Toe	4	2	2	2
Average Rank	3.250	3.166	1.7916	1.7916

Table 4: Performance comparisons of GEPSVM, TWSVM, SNTSVM-1 and SNTSVM-2 on twelve datasets using Gaussian kernel

Datasets (Train size, Test size)	GEPSVM Accuracy(%) Time(s)	TWSVM Accuracy(%) Time(s)	SNTSVM-1 Accuracy(%) Time(s)	SNTSVM-2 Accuracy(%) Time(s)
Cross Planes	90.00	97.50	**100.00**	**100.00**
(81×2, 40×2)	0.1782	0.8575	0.0333	0.0435
Heart-Statlog	75.71	81.43	82.86	**84.29**
(200×13, 70×13)	0.8125	0.4404	0.00234	0.0216
WDBC	**84.06**	78.26	76.81	73.92
(500×30, 69×30)	4.9375	0.6929	0.3363	0.1980
Ionosphere	73.33	94.29	95.24	**96.19**
(246×34, 105×34)	1.4843	0.3374	0.0272	0.04303
Votes	93.02	**96.90**	**96.90**	**96.90**
(306×16, 129×16)	1.7320	0.3638	0.0460	0.1298
WPBC	64.91	75.44	**78.95**	75.44
(137×33, 57×33)	0.3750	0.2142	0.0014	0.0408
Cleve	71.67	72.50	**84.17**	**84.17**
(177×13, 120×13)	0.5625	0.2655	0.0139	0.0408
Monks-3	81.48	90.74	93.29	**93.52**
(432×7, 122×7)	0.6875	0.1773	0.0083	0.0452
Australian	**89.33**	76.00	88.66	88.66
(540×14, 150×14)	5.7812	1.6721	0.0229	0.0724
Haberman	66.98	76.41	**77.36**	**77.36**
(200×3, 106×3)	1.078	1.2963	0.04983	0.0216
Splice	67.03	88.33	88.64	**88.68**
(500×60, 2675×60)	673.359	0.6190	0.1116	0.08402
Tic-Tac-Toe	**94.43**	**94.43**	**94.43**	**94.43**
(671×9, 287×9)	1.7892	2.6819	0.1895	0.2451

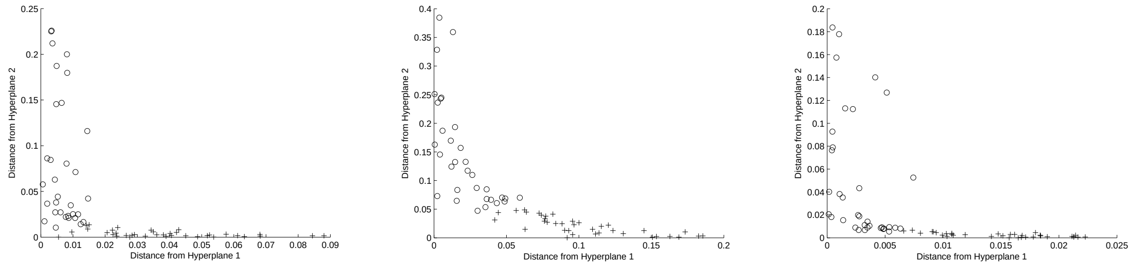
Table 5: Optimal parameters of GEPSVM, TWSVM, SNTSVM-1 and SNTSVM-2 for Gaussian kernel

Datasets	GEPSVM		TWSVM		SNTSVM-1			SNTSVM-2		
	μ	δ	μ	$C_1 = C_2$	μ	$C_1 = C_2$	$C_3 = C_4$	μ	$C_1 = C_2$	$C_3 = C_4$
Cross Planes	2^9	2^{-5}	2^9	10^{-5}	2^{-10}	10^{-5}	10^{-5}	2^{-10}	10^{-5}	10^{-5}
Heart-Statlog	2^5	2^2	2^{-5}	10^{-5}	2^{-10}	10^{-5}	10^{-4}	2^{-1}	10^1	10^1
WDBC	2^4	2^6	2^{-1}	10^{-5}	2^{-1}	10^0	10^{-2}	2^{-3}	10^0	10^{-4}
Ionosphere	2^3	2^{-7}	2^1	10^{-3}	2^{-2}	10^{-1}	10^{-1}	2^{-1}	10^{-1}	10^{-1}
Votes	2^{-4}	2^{-1}	2^{-7}	10^{-5}	2^{-7}	10^2	10^{-4}	2^{-7}	10^2	10^{-4}
WPBC	2^3	2^6	2^0	10^{-5}	2^1	10^{-5}	10^{-5}	2^0	10^{-3}	10^{-5}
Cleve	2^6	2^2	2^{-5}	10^{-5}	2^{-7}	10^1	10^{-1}	2^{-7}	10^1	10^{-1}
Monks-3	2^6	2^{-3}	2^{-3}	10^{-5}	2^{-3}	10^{-1}	10^{-5}	2^{-3}	10^{-1}	10^{-5}
Australian	2^0	2^{-7}	2^{-4}	10^{-5}	2^{-5}	10^{-5}	10^{-4}	2^{-5}	10^{-5}	10^{-4}
Haberman	2^6	2^2	2^1	10^{-5}	2^{-9}	10^{-2}	10^{-5}	2^{-9}	10^{-2}	10^{-5}
Splice	2^{-8}	2^2	2^{-6}	10^{-5}	2^{-3}	10^{-1}	10^{-1}	2^{-3}	10^{-1}	10^{-1}
Tic-Tac-Toe	2^3	2^2	2^{-6}	10^{-5}	2^{-10}	10^0	10^{-5}	2^{-10}	10^0	10^{-5}

Table 6: Average ranks of GEPSVM, TWSVM, SNTSVM-1 and SNTSVM-2 with Gaussian kernel.

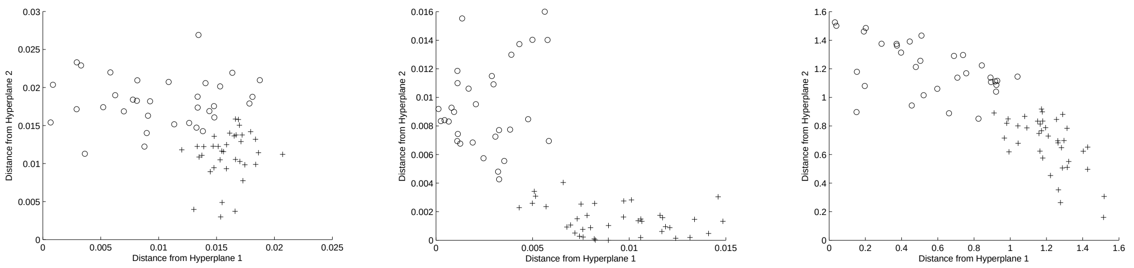
Datasets	GEPSVM	TWSVM	SNTSVM-1	SNTSVM-2
Cross Planes	4	3	1.5	1.5
Heart-Statlog	4	3	2	1
WDBC	1	2	3	4
Ionosphere	4	3	2	1
Votes	4	2	2	2
WPBC	4	2.5	1	2.5
Cleve	4	3	1.5	1.5
Monks-3	4	3	2	1
Australian	1	4	2.5	2.5
Haberman	4	3	1.5	1.5
Splice	4	3	2	1
Tic-Tac-Toe	2.5	2.5	2.5	2.5
Average Rank	3.375	2.833	1.958	1.833

In addition, we also compare our results non-linearly. One can observe from Table 4 that, in comparison to GEPSVM and TWSVM, our methods show better generalization performance. In details, for Heart-Statlog dataset, the experimental results by SNTSVM-1 (82.86%) and SNTSVM-2 (84.29%) are higher than other two algorithms, i.e., TWSVM (81.43%) and GEPSVM (75.71%). We obtained the similar conclusions for Ionosphere, Votes, WPBC, Cleve, Monks-3, Haberman, Splice and Tic-Tac-Toe datasets. For Australian dataset, the classification accuracies obtained by our algorithms (88.66%) are slightly lower than GEPSVM (89.33%), it is higher than TWSVM (76.00%). The empirical results further reveal that our proposed algorithms SNTSVM-1 and SNTSVM-2, whose solutions are obtained by solving system of linear equations, are faster than TWSVM on most of the datasets. One can observe from Table 4 that SNTSVM-1 and SNTSVM-2 show exactly the same accuracy on six of twelve datasets. This clearly indicates that both the approaches are equally preferred. From the perspective of training speed, our proposed SNTSVM-1 and SNTSVM-2 outperform GEPSVM and TWSVM on most of the datasets considered which clearly indicates its superiority. It is worthwhile notice that choosing the values of the parameters C_3 and C_4 affect the results significantly and these values are varying in our SNTSVM-1 and SNTSVM-2 rather than small fixed positive scalar in TWSVM. The details of optimal parameters for linear and Gaussian kernel are listed in Table 2 and Table 5 respectively. It clearly indicates that adding the regularization terms in our formulation are useful.



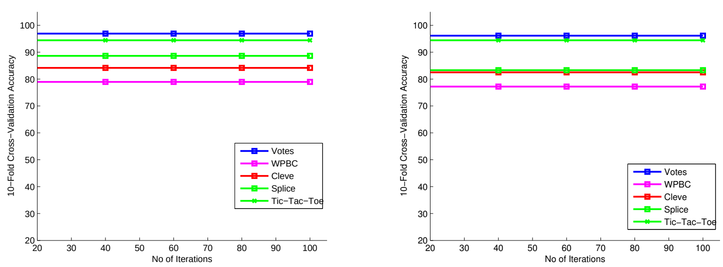
(a) TWSVM (b) SNTSVM-1 (c) SNTSVM-2

Figure 2: 2-D projections of (a) TWSVM (b) SNTSVM-1 (c) SNTSVM-2 from WDBC dataset. +: scatter plot of the positive points. O: scatter plot of the negative points.



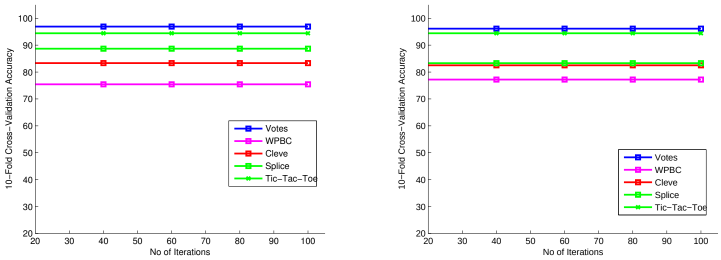
(a) TWSVM (b) SNTSVM-1 (c) SNTSVM-2

Figure 3: 2-D projections of (a) TWSVM (b) SNTSVM-1 (c) SNTSVM-2 from Heart-Statlog dataset. +: scatter plot of the positive points. O: scatter plot of the negative points.



(a) Gaussian (b) Linear

Figure 4: The performance of SNTSVM-1 on number of iterations with 10-fold cross validation accuracy on UCI datasets.



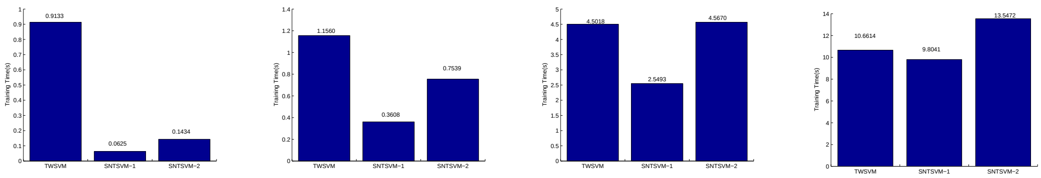
(a) Gaussian (b) Linear

Figure 5: The performance of SNTSVM-2 on number of iterations with 10-fold cross validation accuracy on UCI datasets.

To further compare our SNTSVM-1 and SNTSVM-2 with TWSVM, we also compare it with the two-dimensional scatter plots that were obtained from the part test points for the WDBC and Heart Statlog datasets. The plots were obtained by plotting points with coordinates: perpendicular distance of a test point x from positive hyperplane 1 and the distance from negative hyperplane 2. In the figures, positive points are plotted as "+" and negative points are plotted as "o". One can see from Fig. 2(a-c) and Fig. 3(a-c) that, our proposed SNTSVM-1 and SNTSVM-2 obtained large distances from the test samples to the opposite hyperplanes. In contrast, the TWSVM obtained small distances from the test points to the hyperplane pair. It means that our SNTSVM-1 and SNTSVM-2 are much more robust when compared with the TWSVM. In addition, Fig. 4 and 5 show the relationship between the accuracy and the number of iterations for SNTSVM-1 and SNTSVM-2 respectively. One can observe from figures that the accuracy tends to a stable value quickly after twenty iterations on most of the datasets.

4.3. NDC datasets

We further experimented with four NDC datasets as examples of large synthetic datasets. David Musicant NDC Data generator [24] is used to explore the computing time for these algorithms scale with respect to number of data points. In all the examples considered, the original data is normalized with mean zero and standard deviation equals to 1. For experiments with all NDC datasets, we fixed penalty parameters of all algorithms to be as (i.e. $C_1 = C_2 = C_3 = C_4, \mu = 2^{-4}$). One can observe from Fig. 6 that our proposed SNTSVM-1 and SNTSVM-2 obtained less training time in comparison with TWSVM.



(a) NDC-500 (b) NDC-1k (c) NDC-2k (d) NDC-3k

Figure 6: Comparison of training time on NDC datasets among TWSVM, SNTSVM-1 and SNTSVM-2.

4.4. Friedman Test

Friedman test is used in this paper to verify the statistical significance of our proposed SNTSVM-1 and SNTSVM-2 in comparison to GEPSVM and TWSVM. Friedman test with the corresponding post hoc tests is pointed out to be a simple, safe, and robust non parametric test for comparison of more classifiers over multiple datasets [9], we use it to compare the performance of four algorithms. The average ranks of all the algorithms on accuracies for linear kernel were computed and listed in Table 3. We employ the Friedman test to check whether the measured average ranks are significantly different from the mean rank $R_j = 2.5$

expected under the null hypothesis:

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_{j=1}^{4} R_j^2 - \frac{k(k+1)^2}{4}\right]$$

is distributed according to χ_F^2 with $k-1$ degree of freedom. where k is the number of methods and N is the number of datasets.

$$\chi_F^2 = \frac{12 \times 12}{4(4+1)}\left[3.25^2 + 3.166^2 + 1.7916^2 + 1.7916^2 - \frac{4(5)^2}{4}\right] = 14.4403.$$

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1) - \chi_F^2} = \frac{(12-1) \times 14.4403}{12(4-1) - 14.4403} = 7.3676.$$

With four algorithms and twelve datasets, F_F is distributed according to the $F-$distribution with $(k-1)$ and $(k-1)(N-1) = (3,33)$ degrees of freedom. The critical value of $F(3,33)$ for $\alpha = 0.05$ is 2.892. So, we reject the null hypothesis $(F_F > F(3,33))$. We use the Nemenyi test for further pairwise comparison. According to [9], at $p = 0.10$, critical difference (CD) $= q_\alpha \sqrt{\frac{k(k+1)}{6N}} = 1.2959$. Since the difference between TWSVM and our proposed SNTSVM-I and SNTSVM-2 is larger than the critical difference 1.2959($3.166 - 1.7916 = 1.3744 > 1.2959$), we can identify that the performance of SNTSVM-I and SNTSVM-2 are significantly better than TWSVM. In the same way, we see that the performance of SNTSVM-I and SNTSVM-2 are better than GEPSVM.

For the nonlinear case, we also compare the performance of four algorithms statistically. The average ranks of all the algorithms on accuracies were computed and listed in Table 6. We can calculate that the F_F value on accuracies is 5.2251, which is larger than the critical value 2.892, so we reject the null hypothesis. Since the difference between SNTSVM-1, SNTSVM-2 and TWSVM is smaller than the critical difference, we can conclude that there is no significant difference among the algorithms. Further, the performance of SNTSVM-1 is significantly better than GEPSVM ($3.375 - 1.958 = 1.4170 > 1.2959$). In the same way, we can conclude that the performance of SNTSVM-2 is significantly better than GEPSVM and TWSVM.

5. Conclusions and Future Works

In this paper, we proposed two smoothing approaches for an implicit Lagrangian twin support vector machine classifiers by formulating a pair of unconstrained minimization problems in dual variables whose solutions will be obtained by solving two systems of linear equations rather than solving two QPPs in TWSVM. Our proposed formulation introduces regularization term to each objective function with the idea of maximizing the margin. In addition, our proposed formulation becomes well-posed model due to this term, which introduces invertibility in the dual formulation. Moreover, the structural risk minimization principle is implemented in our formulation which embodies the essence of statistical learning theory. The experimental results on several benchmark datasets show that our SNTSVM-1 and SNTSVM-2 are feasible and effective on both generalization ability and training speed. Parameter selection of our formulation is a practical problem and should be addressed in the future studies. Furthermore, we feel that extending our formulation to multi-class classification and semi-supervised learning are also interesting and under our consideration.

References

[1] Balasundaram S, Tanveer M (2012) On proximal bilateral-weighted fuzzy support vector machine classifiers. International Journal of Advanced Intelligence Paradigms 4(3-4):199-210

[2] Balasundaram S, Tanveer M (2013) On Lagrangian twin support vector regression. Neural Computing and Applications 22(1):257-267

[3] Boser B, Guyon L, Vapnik VN (1992) A training algorithm for optimal margin classifiers In:Proceedings of Fifth Annual Workshop on Computational Learning Theory ACM Press, Pittsburgh, 144-152

[4] Burges C (1998) A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2:1-43

[5] Brown MPS, Grundy WN, Lin D (2000) Knowledge-based analysis of micro-array gene expression data using support vector machine In: Proceedings of the National Academy of Sciences of USA 97(1):262-267

[6] Chang CC, Lin CJ (2011) LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2(3):27

[7] Chen X, Yang J, Liang J, Ye Q (2012) Smooth twin support vector regression. Neural Computing and Applications 21(3):505-513

[8] Cortes C, Vapnik VN (1995) Support vector networks. Machine Learning 20: 273-297

[9] Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7:1-30

[10] Duda RO, Hart PR, Stork D.G. (2001) Pattern Classification second ed. John Wiley and Sons

[11] Fung G, Mangasarian OL (2003) Finite Newton method for Lagrangian support vector machine classification. Neurocomputing 55(1-2):39-55

[12] Hsu CW, Lin CJ (2002) A comparison of methods for multi-class support vector machines. IEEE Transactions on Neural Networks 13:415-425

[13] Jayadeva, Khemchandani R, Chandra S (2007) Twin support vector machines for pattern classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(5):905-910

[14] Joachims T, Ndellec C, Rouveriol (1998) Text categorization with support vector machines: learning with many relevant features. In: European Conference on Machine Learning, Chemnitz, Germany 10:137-142

[15] Joachims T (1999) Making large-scale support vector machine learning practical. Advances in Kernel Methods. Support Vector Learning, MIT Press, Cambridge, MA

[16] Kumar, MA, Gopal, M (2008) Application of smoothing technique on twin support vector machines. Pattern Recognition Letters 29:1842-1848

[17] Lee YJ, Mangasarian OL (2001a) RSVM: Reduced support vector machines, In:Proceedings of the First SIAM International Conference on Data Mining, pp 5-7

[18] Lee YJ, Mangasarian OL (2001b) SSVM: A Smooth support vector machine for classification. Computational Optimization and Applications 20(1):5-22

[19] Mangasarian OL (1994) Nonlinear Programming, SIAM Philadelphia, PA

[20] Mangasarian OL (2002) A finite Newton method for classification. Optimization Methods Softwares 17:913-929.

[21] Mangasarian OL, Musicant DR (2001) Lagrangian support vector machines. Journal of Machine Learning Research 1:161-177.

[22] Mangasarian OL, Solodov MV (1993) Nonlinear complementarity as unconstrained and constrained minimization. Mathematical Programming, Series B 62:277-297

[23] Mangasarian OL, Wild EW (2006) Multisurface proximal support vector classification via generalized eigenvalues. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(1):69-74

[24] Musicant, DR (1998) NDC: Normally distributed clustered datasets. http://www.cs.wisc.edu/ musicant/data/ndc

[25] Murphy PM, Aha DW (1992) UCI repository of machine learning databases. University of California, Irvine. http://www.ics.uci.edu/ mlearn

[26] Osuna E, Freund R, Girosi F (1997) Training support vector machines: An application to face detection. In: Proceedings of Computer Vision and Pattern Recognition, pp 130-136

[27] Peng X (2010) Primal twin support vector regression and its sparse approximation. Neurocomputing 73:2846-2858

[28] Peng X (2010) TSVR: An efficient twin support vector machine for regression. Neural Networks 23(3):365-372

[29] Platt J (1999) Fast training of support vector machines using sequential minimal optimization In: B. Scholkopf, Burges CJC, Smola AJ (1999), Advances in Kernel Methods-Support Vector Learning, MIT Press, Cambridge, MA pp 185-208

[30] Tanveer M (2013) Smoothing technique on linear programming twin support vector machines. International Journal of Machine Learning and Computing 3 (2):240-244

[31] Shao YH, Zhang CH, Wang XB, Deng NY (2011) Improvements on twin support vector machines. IEEE Transactions on Neural Networks 22(6):962-968

[32] Shao YH, Chen WJ, Zhang JJ, Wang Z, Deng NY (2014) An efficient weighted Lagrangian twin support vector machine for imbalanced data classification. Pattern Recognition 47(9):3158-3167

[33] Tanveer M (2015a) Application of smoothing techniques for linear programming twin support vector machines. Knowledge and Information Systems 45(1):191-214

[34] Tanveer M (2015b) Robust and sparse linear programming twin support vector machines. Cognitive Computation 7:137-149

[35] Tanveer M (2017) Linear programming twin support vector regression. Filomat 31(7) 2123–2142

[36] Tian Y, Ping Y (2014) Large-scale linear nonparallel support vector machine solver. Neural Networks 50: 166-174

[37] Tsang IW, Kocsor A, Kwok JT (2007) Simpler core vector machines with enclosing balls In: Proceedings of the 24th International Conference on Machine Learning, Corvallis, pp 911-918

[38] Xu Y, Wang L (2012) A weighted twin support vector regression. Knowledge-Based Systems 33:92-101

[39] Zhong P, Xu Y, Zhao Y (2012) Training twin support vector regression via linear programming. Neural Computing and Applications 21(2):399-407

[40] Vapnik VN (1998) Statistical Learning Theory, Wiley, New York