

Smoothing Technique on Linear Programming Twin Support Vector Machines

M. Tanveer

Abstract—In this paper, a new smoothing technique on linear programming TWSVM formulation is proposed whose solution is obtained by solving a pair of dual exterior penalty problems as unconstrained minimization problems using Newton method. Our approach has the advantage that a pair of matrix equation of order equals to the number of input examples is solved at each iteration of the algorithm and can be easily implemented in MATLAB without using any optimization toolbox. Computational comparisons of our proposed method against original TWSVM, GEPSVM and SVM indicate that our method is not only fast, but also shows good generalization performance.

Index Terms—Linear programming, 1-norm support vector machines, Smoothing technique, Newton method, Twin support vector machines

I. INTRODUCTION

Support Vector Machine (SVM), introduced by Vapnik and coworkers [1], [2], [3], [4] is an excellent kernel based tool for binary data classification problems. 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.

Different from SVMs with two different hyper planes, some non hyper planes classifiers such as the generalized Eigen value proximal support vector machine (GEPSVM) [5] and Twin support vector machine (TWSVM) [6] have been proposed recently. TWSVM generates two nonparallel hyper planes such that each plane is closest to one of the classes and as far as possible from the other class. A fundamental difference between TWSVM and SVM is that TWSVM solves two small QPPs rather than solving one large QPP as in SVM. In terms of generalizations, TWSVM favorably compare with SVM and GEPSVM. Recently, some extensions have been proposed that includes the least square TWSVM [7]-[8], Smooth TWSVM [9], improvements on TWSVM (TBSVM) [10], TPMSVM [11], TSVR [12] and LTSVR [13]. Finally, on the recent study of 1-norm TWSVM the interested reader is referred to [14], [15].

To the best of our knowledge, 2-norm distance of residuals is sensitive to large errors and therefore less robust in comparison to 1-norm, many methods exist in the literature to solve 2-norm SVM formulated as a QPP but very little on 1-norm linear programming SVM. Motivated by the study of 1-norm SVM problem formulated as a linear programming

optimization problem [16] and Smooth TWSVM [9], we propose in this work smooth linear programming twin support vector machines (SLPTSVM) whose solution is obtained, by solving a pair of exterior penalty problems in dual as unconstrained optimization problems using Newton-Armijo algorithm. To demonstrate the effectiveness of the proposed method, we performed numerical experiments on a number of interesting real-world datasets and compared their results with other SVMs.

In this work, all vectors are taken as column vectors. The inner product of two vectors x, y in the n -dimensional real space R^n will be denoted by: $x^t y$, where x^t is the transpose of x . Whenever x is orthogonal to y , we write $x \perp y$. For $x = (x_1, \dots, x_n)^t \in R^n$, the plus function x_+ is defined as: $(x_+)_i = \max\{0, x_i\}$, where $i = 1, \dots, n$. The 2-norm of a vector x and a matrix Q will be denoted by $\|x\|$ and $\|Q\|$ respectively. We denote the vector of one's of dimension m by e and the identity matrix of appropriate size by I .

The paper is organized as follows. Section II gives the formulation of TWSVM and then gives the proposed formulation SLPTSVM problem whose solution is obtained by solving a pair of exterior penalty problems in dual as unconstrained optimization problems using Newton-Armijo algorithm. Numerical experiments have been performed on a number of interesting datasets and their results are compared with SVM, GEPSVM and TWSVM in Section III.

II. LINEAR PROGRAMMING TWIN SUPPORT VECTOR MACHINES

A. Twin Support Vector Machines

Suppose that all the data points in class +1 are denoted by a matrix $A \in R^{m_1 \times n}$, where the i^{th} row $A_i \in R^n$ and the matrix $B \in R^{m_2 \times n}$ represent the data points of class -1. Unlike SVM, the linear TWSVM [6] seeks a pair of nonparallel hyper planes:

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

Such that each hyper plane 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:

Manuscript received February 2, 2013; revised April 20, 2013.

M. Tanveer is with the Department of Computer Science & Engineering, The LNM Institute of Information Technology, Jaipur 302 031, India (e-mail:tanveergouri@gmail.com).

$$\begin{aligned} \min \quad & \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + C_1 \|\xi_1\| \\ \text{s.t.} \quad & -(Bw_1 + e_2 b_1) + \xi_1 \geq e_2, \xi_1 \geq 0 \end{aligned} \quad (2)$$

$$\begin{aligned} \min \quad & \frac{1}{2} \|Bw_2 + e_2 b_2\|^2 + C_2 \|\xi_2\| \\ \text{s.t.} \quad & (Aw_2 + e_1 b_2) + \xi_2 \geq e_1, \xi_2 \geq 0, \end{aligned} \quad (3)$$

where $C_1, C_2 > 0$ are parameters and e_1, e_2 are vectors of one's 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 [6].

Thus the nonparallel proximal hyper planes are obtained from the dual of (2) and (3) by

$$\begin{aligned} \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} &= -(G'G + \delta I)^{-1} H' \alpha & \text{and} \\ \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} &= (H' H + \delta I)^{-1} G' \gamma. \end{aligned} \quad (4)$$

B. Proposed Smoothing Technique on Linear Programming Twin Support Vector Machine (SLPTSVM)

Motivation from the works on exact 1-norm SVMs [16], TWSVM [6] and Smooth TWSVM [9], we propose a new linear programming TWSVM using smoothing technique by formulating a pair of unconstrained minimization problems whose solutions can be obtained using fast Newton method.

For this purpose, first let us consider the linear TWSVM problem defined in 1-norm as a pair of minimization problems of the form:

$$\begin{aligned} \min_{w_1 \in R^n, b_1 \in R} \quad & \|Aw_1 + e_1 b_1\|_1 + C_1 \|\xi_1\|_1 + \eta_1 \left\| \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} \right\|_1 \\ \text{s.t.} \quad & -(Bw_1 + e_2 b_1) + \xi_1 \geq e_2, \xi_1 \geq 0 \end{aligned} \quad (5)$$

$$\min_{w_2 \in R^n, b_2 \in R} \quad \|Bw_2 + e_2 b_2\|_1 + C_2 \|\xi_2\|_1 + \eta_2 \left\| \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} \right\|_1 \quad (5)$$

$$\text{s.t.} \quad (Aw_2 + e_1 b_2) + \xi_2 \geq e_1, \xi_2 \geq 0, \quad (6)$$

where A and B are matrices of sizes $m_1 \times n$ and $m_2 \times n$ respectively, e_1 and e_2 are the vectors of one's of sizes m_1 and m_2 respectively.

Following the approach of Mangasarian [16], we will obtain the solutions of the 1-norm TWSVM (5) and (6) by converting them into a pair of linear programming problems (LPPs) in primal and solving the exterior penalty functions of their duals for a finite value of a penalty parameter θ .

Let $G = [A \ e_1]$, $H = [B \ e_2]$ be two augmented matrices of sizes $m_1 \times (n + 1)$ and $m_2 \times (n + 1)$ respectively. Then, by setting

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = p_1 - q_1, \quad G(p_1 - q_1) = r_1 - s_1, \quad \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = p_2 - q_2, \quad H(p_2 - q_2) = r_2 - s_2,$$

where $p_1, q_1, p_2, q_2 \in R^{n+1}$; $r_1, s_1 \in R^{m_1}$ and $r_2, s_2 \in R^{m_2}$ satisfying the non-negativity constraints

$$p_1, q_1, p_2, q_2, r_1, s_1, r_2, s_2 \geq 0$$

The above pair of problems (5) and (6) can be converted into the following pair of linear programming twin support vector machine (LPTSVM) problems of the form: respectively, where e is the vector of one's of size $(n + 1)$.

$$\begin{aligned} \min_{r_1, s_1 \in R^{m_1}; p_1, q_1 \in R^{n+1}; \xi_1 \in R^{m_2}} \quad & e_1' (r_1 + s_1) + C_1 e_2' \xi_1 + \eta_1 e_1' (p_1 + q_1) \\ \text{s.t.} \quad & -H(p_1 - q_1) + \xi_1 \geq e_2 \\ & G(p_1 - q_1) - (r_1 - s_1) = 0 \\ & p_1, q_1, r_1, s_1, \xi_1 \geq 0 \end{aligned} \quad (7)$$

$$\begin{aligned} \min_{r_2, s_2 \in R^{m_2}; p_2, q_2 \in R^{n+1}; \xi_2 \in R^{m_1}} \quad & e_2' (r_2 + s_2) + C_2 e_1' \xi_2 + \eta_2 e_2' (p_2 + q_2) \\ \text{s.t.} \quad & G(p_2 - q_2) + \xi_2 \geq e_1 \\ & H(p_2 - q_2) - (r_2 - s_2) = 0 \\ & p_2, q_2, r_2, s_2, \xi_2 \geq 0 \end{aligned} \quad (8)$$

The method of converting the 1-norm linear TWSVM formulation defined by (5) and (6) into LPP formulations can be extended similarly to its nonlinear version.

Now we focus on the method of obtaining the solutions for both the linear and nonlinear TSVM defined by (7) and (8).

Using Theorem 1[16], the pair of unconstrained dual exterior penalty problems with penalty parameter $\theta > 0$, corresponding to the linear problems (7) and (8), defined by

$$\begin{aligned} \min_{(u_1, v_1) \in R^{2 \times (n+1)}} \quad & L_1(u_1, v_1) = -\theta e_1' u_1 + \frac{1}{2} [\|(-v_1 - e_1)_+\|^2 + \|(v_1 - e_1)_+\|^2 \\ & + \|(-H' u_1 + G' v_1 - \eta_1 e)_+\|^2 + \|(H' u_1 - G' v_1 - \eta_1 e)_+\|^2 + \\ & \|(u_1 - C_1 e_2)_+\|^2] + \|(-u_1)_+\|^2 \end{aligned} \quad (9)$$

and

$$\begin{aligned} \min_{(u_2, v_2) \in R^{2 \times (n+1)}} \quad & L_2(u_2, v_2) = -\theta e_2' u_2 + \frac{1}{2} [\|(-v_2 - e_2)_+\|^2 + \|(v_2 - e_2)_+\|^2 \\ & + \|(G' u_2 + H' v_2 - \eta_2 e)_+\|^2 + \|(-G' u_2 - H' v_2 - \eta_2 e)_+\|^2 + \\ & \|(u_2 - C_2 e_1)_+\|^2] + \|(-u_2)_+\|^2 \end{aligned} \quad (10)$$

are solvable for all $\theta > 0$ and further there exists $\bar{\theta} > 0$ such that for any $\theta \in (0, \bar{\theta}]$ we have: hold, where (u_1, v_1) and (u_2, v_2) are the solutions of the minimization problems (9) and (10) respectively.

$$\begin{bmatrix} r_1 \\ s_1 \\ p_1 \\ q_1 \\ \xi_1 \end{bmatrix} = \frac{1}{\theta} \begin{bmatrix} (-v_1 - e_1)_+ \\ (v_1 - e_1)_+ \\ (-H'u_1 + G'v_1 - \eta_1 e)_+ \\ (H'u_1 - G'v_1 - \eta_1 e)_+ \\ (u_1 - C_1 e_2)_+ \end{bmatrix} \text{ and} \quad (11)$$

$$\begin{bmatrix} r_2 \\ s_2 \\ p_2 \\ q_2 \\ \xi_2 \end{bmatrix} = \frac{1}{\theta} \begin{bmatrix} (-v_2 - e_2)_+ \\ (v_2 - e_2)_+ \\ (G'u_2 + H'v_2 - \eta_2 e)_+ \\ (-G'u_2 - H'v_2 - \eta_2 e)_+ \\ (u_2 - C_2 e_1)_+ \end{bmatrix}$$

For solving the unconstrained minimization problems (9) and (10) using Newton-Armijo algorithm, the gradient vector and the Hessian matrix of $L_k(.,.)$ should be known.

However, since the gradient of $L_k(.,.)$ for $k=1,2$, which can be from (9) and (10) is not differentiable and therefore the Hessian matrix of second order partial derivatives of $L_k(.,.)$ does not exist in the usual sense. To overcome this problem, a generalized Hessian in the sense of [17] may be used and the pair of problems (9) and (10) can be solved by Newton-Armijo algorithm.

In this work we employ smoothing technique where the function x_+ appearing in (9) and (10) is replaced by an infinitely differentiable smooth approximation function $p(x, \eta)$, defined by [18]:

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

In which $\eta > 0$ is a parameter, so that the gradient vectors and the Hessian matrices of the modified objective functions exist and hence Newton-Armijo algorithm can be applied.

Notice that $\nabla^2 L_k$ is positive semi-definite matrix of order m . Since it is possible that the matrix may be ill conditioned and therefore we will use $(\delta I + \nabla^2 L_k)^{-1}$ in place of inverse of $\nabla^2 L_k$ where the regularization parameter δ is taken as a very small positive number.

III. ANALYSIS OF EXPERIMENTAL RESULTS

In order to demonstrate the effectiveness of the proposed method, it is tested on cross-plane dataset, David Musicant's NDC Data generator [19] datasets as examples of large synthetic datasets and several well-known, publicly available, benchmark datasets. We compared their results with SVM, GEPSVM and TWSVM. All the experiments were performed in MATLAB R2010a environment on a PC running on Windows XP OS with 2.27 GHz Intel(R) Xeon(R) processor having 3 GB of RAM. The standard SVM was solved by LIBSVM [20]. For GEPSVM and TWSVM, however, we used the optimization toolbox of MATLAB. In all the examples considered, the 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 taken. The classification accuracy of each algorithm was computed using the well-known tenfold cross-validation methodology [21].

The optimal values of the parameters were determined by performing 10-fold cross-validation on the training dataset, where the regularization parameter values $C_1 = C_2 = C$ and the kernel parameter value μ were allowed to vary from the sets $\{10^{-5}, 10^{-4}, \dots, 10^5\}$ and $\{2^{-10}, 2^{-9}, \dots, 2^{10}\}$ respectively. Further, we set $\eta_1 = \eta_2 = 10^{-5}$ and the penalty parameter $\theta = 0.1$. For GEPSVM, the range of δ was allowed to vary from the set $\{2^{-7}, 2^{-6}, \dots, 2^7\}$. Finally, choosing these optimal values, the classification accuracy on the test dataset was calculated.

First, we consider a simple two dimensional "Cross Planes" dataset as an example of synthetic dataset. It was generated by perturbing the points of two intersecting lines. The linear classifiers obtained by our SLPTSVM and the standard SVM along with the input data are shown in Fig. 1(a) and Fig. 1(b) respectively. Also we computed the classification accuracy for nonlinear classifiers of each algorithm for "Cross Planes" and summarized them in Table I. The results clearly demonstrate the superiority of multi-plane/surface classifiers over the standard SVM.

Next, we experimented with NDC datasets, generated using David Musicant NDC Data generator [19] to get a clear representation of how all 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., $C_1 = 1, C_2 = 1, \mu = 2^{-4}$).

Furthermore, we performed numerical experiments on benchmark datasets from UCI repository [22] and compared their classifier accuracy with GEPSVM, TWSVM and SVM.

In all the real-world examples considered, each attribute of the original data is normalized as follows:

$$\bar{x}_{ij} = \frac{x_{ij} - x_j^{\min}}{x_j^{\max} - x_j^{\min}}$$

where x_{ij} is the (i,j)-th element of the input matrix A , \bar{x}_{ij} is its corresponding normalized value and $x_j^{\min} = \min_{i=1}^m(x_{ij})$ and $x_j^{\max} = \max_{i=1}^m(x_{ij})$ denote the minimum and maximum values, respectively, of the j-th column of A .

The size of training and test data, the number of attributes, the training time and the accuracy of each algorithm for nonlinear classifiers are summarized in Table I. Clearly one can observe from Table I, our method SLPTSVM shows either better or comparable generalization performance in comparison to SVM, GEPSVM and TWSVM. The empirical

results further reveal that our proposed iterative algorithm, whose solutions are obtained by solving system of linear equations, show better generalization performance than TWSVM. It is also worth mentioning that our proposed algorithm do not require any special optimization packages. Our results clearly demonstrate that proposed algorithm is a powerful method of solution for classification problems.

TABLE I: PERFORMANCE COMPARISON OF THE PROPOSED METHOD WITH SVM, GEPSVM AND TWSVM. TEST ACCURACY AND TRAINING TIME WERE USED FOR COMPARISON. GAUSSIAN KERNEL WAS EMPLOYED. BOLD TYPE SHOWS THE BEST RESULTS.

Datasets (Train size, Test size)	SVM Time (C, μ)	GEPSVM Time (μ, δ)	TSM Time ($C_1 = C_2, \mu$)	SLPTSVM Time ($C_1 = C_2, \mu$)
Cross Planes (90×2,40×2)	97.50 1.2474 ($10^2, 2^7$)	90.00 0.1782 ($2^9, 2^{-5}$)	97.50 0.4094 ($10^{-5}, 2^9$)	97.50 0.0187 ($10^{-1}, 2^6$)
Ionosphere (246×34,105×34)	93.33 0.6064 ($10^1, 2^{-3}$)	73.33 1.4843 ($2^3, 2^{-7}$)	94.28 0.1290 ($10^{-3}, 2^1$)	95.23 0.1287 ($10^{-5}, 2^{-1}$)
Bupa (241×6,104×6)	64.42 0.5156 ($10^4, 2^{-3}$)	46.15 1.125 ($2^{-6}, 2^5$)	64.42 0.1799 ($10^{-5}, 2^1$)	71.15 0.0922 ($10^{-1}, 2^{-4}$)
Votes (306×16,129×16)	96.90 0.5781 ($10^1, 2^{-4}$)	93.02 1.734 ($2^{-4}, 2^{-1}$)	96.90 0.1812 ($10^{-5}, 2^{-7}$)	96.90 0.9234 ($10^1, 2^{-7}$)
CMC (1000×9,473×9)	75.05 2.8594 ($10^0, 2^{-5}$)	75.05 40.234 ($2^7, 2^{-4}$)	74.63 20.020 ($10^{-5}, 2^{-2}$)	74.84 5.4641 ($10^1, 2^0$)
Pima (537×8,231×8)	78.78 0.7031 ($10^0, 2^{-9}$)	77.48 5.7623 ($2^{-1}, 2^5$)	73.16 1.6562 ($10^1, 2^0$)	79.22 4.1232 ($10^0, 2^{-4}$)
Australian (540×14,150×14)	89.33 1.6401 ($10^{-1}, 2^{-5}$)	89.33 5.7812 ($2^0, 2^{-7}$)	76.00 0.8016 ($10^{-5}, 2^{-4}$)	89.33 4.0654 ($10^1, 2^{-7}$)
NDC-500 (500×32,50×32)	94.00 0.5407 ($10^0, 2^{-4}$)	95.00 6.2033 ($2^{-3}, 2^{-4}$)	94.00 0.3139 ($10^0, 2^{-4}$)	96.00 1.0818 ($10^0, 2^{-4}$)
NDC-1k (1000×32,100×32)	95.00 1.3366 ($10^0, 2^{-4}$)	97.00 8.543 ($2^{-3}, 2^{-4}$)	97.00 2.1347 ($10^0, 2^{-4}$)	97.00 1.9940 ($10^0, 2^{-4}$)

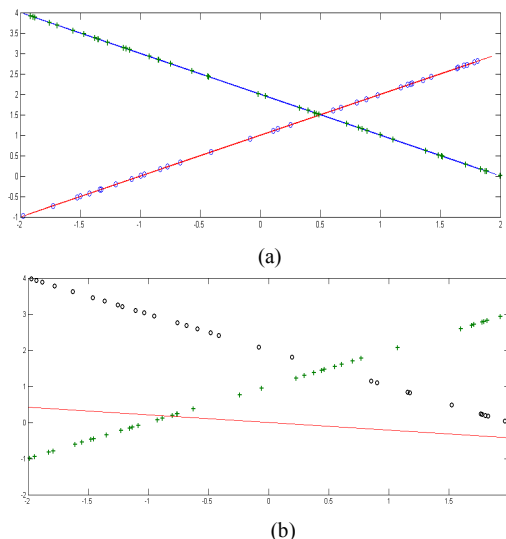


Fig. 1. Classification results of (a) SLPTSVM and (b) SVM for “Cross planes” datasets

IV. CONCLUSION

In this paper, we propose a new smooth approach for linear programming twin support vector machines (SLPTSVM) whose solution is obtained, by solving a pair of exterior penalty problems in dual as unconstrained optimization problems using Newton-Armijo algorithm. The experimental results show the generalization of our method is better than TWSVM, GEPSVM and SVM. Our SLPTSVM is significantly faster than TWSVM in most of the datasets. Also, our approach has the advantage that a pair of matrix equation of order equals to the number of input examples is solved at each iteration of the algorithm and can be easily implemented in MATLAB without using commands of optimization toolbox. The efficiency of the proposed method is demonstrated by experimental results on a number of interesting synthetic and real-world datasets.

ACKNOWLEDGEMENT

I would thank to my thesis advisor Professor S. Balasundaram for his help during the preparation of this manuscript. I acknowledge the financial support given by Council of Scientific and Industrial Research, India and The LNM Institute of Information Technology, Jaipur, India.

REFERENCES

- [1] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, pp. 1-43, 1998.
- [2] C. Cortes and V. N. Vapnik, “Support vector networks,” *Machine Learning*, vol. 20, pp. 273-297, 1995.
- [3] N. Cristianini and J. S. Taylor, “*An introduction to support vector machines and other kernel based learning method*,” Cambridge University Press, Cambridge, 2000.
- [4] V. N. Vapnik, “*Statistical Learning Theory*,” Wiley, New York, 1998.
- [5] O. L. Mangasarian and E. W. Wild, “Multisurface proximal support vector classification via generalized eigenvalues,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 69-74, 2006.
- [6] Jayadeva, R. Khemchandani, and S. Chandra, “Twin support vector machines for pattern classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 905-910, 2007.
- [7] M. A. Kumar and M. Gopal, “Least squares twin support vector machines for pattern classification,” *Expert Systems with Applications*, vol. 36, pp. 7535-7543, 2009.
- [8] M. A. Kumar, R. Khemchandani, M. Gopal, and S. Chandra, “Knowledge based Least Squares Twin support vector machines,” *Information Science*, vol. 180, no. 23, pp. 4606-4618, 2010.
- [9] M. A. Kumar and M. Gopal, “Application of smoothing technique on twin support vector machines,” *Pattern Recognition Letters*, vol. 29, pp. 1842-1848, 2008.
- [10] Y. H. Shao, C. H. Zhang, X. B. Wang, and N. Y. Deng, “Improvements on twin support vector machines,” *IEEE Transactions on Neural Networks*, vol. 22, no. 6, pp. 962-968, 2011.
- [11] X. Peng, “TPMSVM: A novel twin parametric-margin support vector machine for pattern recognition,” *Pattern Recognition*, vol. 44, pp. 2678-2692, 2011.
- [12] X. Peng, “TSVR: An efficient twin support vector machine for regression,” *Neural Networks*, vol. 23, no. 3, pp. 365-372, 2010.
- [13] S. Balasundaram and M. Tanveer, “On Lagrangian twin support vector regression,” *Neural Computing & Applications*, 2012.
- [14] S. Gao, Q. Ye, and N. Ye, “1-Norm least squares twin support vector machines,” *Neurocomputing*, vol. 74, pp. 3590-3597, 2001.
- [15] P. Zhong, Y. Xu, and Y. Zhao, “Training twin support vector regression via linear programming,” *Neural Computing & Applications*, vol. 21, no. 2, pp. 399-407, 2012.
- [16] O. L. Mangasarian, “Exact 1-norm support vector machines via unconstrained convex differentiable minimization,” *Journal of*

Machine Learning Research, vol. 7, pp. 1517–1530, 2006.

- [17] J. B. H. Urruty, J. J. Strodiot, and V. H. Nguyen, “Generalized hessian matrix and second order optimality conditions for problems with CL1 data,” *Applied Mathematics and Optimization*, vol. 11, pp. 43–56, 1984.
- [18] Y. J. Lee and O. L. Mangasarian, “A smooth support vector machine for classification,” *Computational Optimization and Applications*, vol. 20, no. 1, pp. 5-22, 2001.
- [19] D. R. Musicant. NDC: Normally distributed clustered datasets. (1998). [Online]. Available: <http://www.cs.wisc.edu/~musicant/data/ndc>.
- [20] C. C. Chang and C. J. Lin. LIBSVM: A library for support vector machines. [Online]. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2001.
- [21] R. O. Duda, P. R. Hart, and D. G. Stork, “Pattern Classification,” second ed.. John Wiley and Sons, 2001.

- [22] P. M. Murphy and D. W. Aha, “UCI repository of machine learning databases,” University of California, Irvine, 1992.



M. Tanveer was born on April 18, 1983 in Saharanpur, U.P. India. He received his M.Sc and M.Phil degree in Mathematics from Aligarh Muslim University, Aligarh, India. He completed his Ph.D degree in Computer Science from Jawaharlal Nehru University, New Delhi, India. He is currently working as a faculty of Computer Science & Engineering at The LNM Institute of Information Technology, Jaipur, India. His research interests include support vector machine and fixed point theory.