International Journal of Database Theory and Application
Vol.8, No.6 (2015), pp.285-302
http://dx.doi.org/10.14257/ijdta.2015.8.6.26

Multiclass Least Squares Twin Support Vector Machine for Pattern Classification

Divya Tomar and Sonali Agarwal

Indian Institute of Information and Technology, Allahabad, India
divyatomar26@gmail.com and sonali@iiita.ac.in

Abstract

This paper proposes a Multiclass Least Squares Twin Support Vector Machine (MLSTSVM) classifier for multi-class classification problems. The formulation of MLSTSVM is obtained by extending the formulation of recently proposed binary Least Squares Twin Support Vector Machine (LSTSVM) classifier. For M-class classification problem, the proposed classifier seeks M-non parallel hyper-planes, one for each class, by solving M-linear equations. A regularization term is also added to improve the generalization ability. MLSTSVM works well for both linear and non-linear type of datasets. It is relatively simple and fast algorithm as compared to the other existing approaches. The performance of proposed approach has been evaluated on twelve benchmark datasets. The experimental result demonstrates the validity of proposed MLSTSVM classifier as compared to the typical multi-classifiers based on 'Support Vector Machine' and 'Twin Support Vector Machine'. Statistical analysis of the proposed classifier with existing classifiers is also performed by using Friedman's Test statistic and Nemenyi post hoc techniques.

Keywords: *Least Squares Twin Support Vector Machine, Multiclass Least Squares Twin Support Vector Machine, Pattern Classification, Twin Support Vector Machine*

1. Introduction

Vapnik *et al*. proposed an effective classifier, Support Vector Machine (SVM), on the basis of Structural Risk Minimization (SRM) concept in order to reduce the risk occurrence during training phase [1-4]. SVM provides a global solution to classify the data patterns of different classes. Earlier, SVM was a well-established and known technique for binary classification; later researchers successfully extended it for multi-class problem domain [5-13]. SVM is widely accepted as a supervised machine learning approach which is helpful to perform classification and regression tasks [14-20]. The basic concept of SVM is to generate an optimal separation among two classes with maximum margin. Implementation of SVM is complicated and time consuming which requires solving a complex Quadratic Programming Problem (QPP) with inequality constraints.

Recently, Mangasarian *et al*. introduced a Generalized Eigen-value Proximal SVM (GEPSVM) which generates two non-parallel hyper-planes for two class classification [21]. In this approach, the patterns of each class lie in the close proximity of one hyper-plane and maintain clear separation with other. On the basis of SVM and GEPSVM, Jayadeva *et al*. proposed a novel binary classifier, Twin Support Vector Machine (TWSVM), which classifies the patterns of two classes by using two non-parallel hyper-planes [22]. TWSVM solves a pair of QPPs instead of single complex QPP as in SVM which makes the learning of TWSVM four times faster as compared to conventional SVM [22-23]. In SVM, all patterns together provide constraints to QPP, while in TWSVM patterns of one of the two classes provide constraints to each QPP. TWSVM has been applied to various real life applications, for example, disease diagnosis, software defect prediction, intrusion detection, emotion recognition, image annotation, speaker identification *etc*. [24-32]. In various latest

ISSN: 2005-4270 IJDTA

research advancements, TWSVM is also recognized as a possible solution for multi-class problems due to its better computational speed and comparable predictive accuracy [32-35].

In TWSVM, solution of two simple QPPs leads to high computational cost. Therefore, Kumar *et al*. proposed LSTSVM, a binary classifier, that solves two linear problems instead of two QPPs and determines two non-parallel hyper-planes one for each class [36]. LSTSVM has better generalization ability and faster computational speed as compared to the traditional TWSVM, but it works only for two-class classification problems. Similar as SVM and TWSVM, LSTSVM started with its applicability in two-class problems. However, most of the real world applications such as activity recognition, disease detection, emotion recognition, text categorization, speaker identification *etc.* are related to the multi-class classifications. Thus, the main purpose of this research work is to extend the binary LSTSVM in order to handle the multi-class classification problem. In this paper, we propose a novel multiclass classification approach, termed as MLSTSVM, which is an extension of binary LSTSVM. A regularization term is also added to the formulation of MLSTSVM in order to implement the regularized risk minimization and to avoid the singularity problems. For M-class classification, MLSTSVM solves M-linear equations to generate 'M' non-parallel hyper-planes, one plane for each class. For i=1,…,M, the patterns of ith class lie in close proximity with the ith hyper-plane and as far as possible from rest of the hyper-planes. The decision regarding the class, which is assigned to the test pattern, depends upon its distance from hyper-planes. For each new pattern, its perpendicular distance is calculated from each hyper-plane and pattern is assigned to the class from which its distance is lesser.

The paper is organized into 5 sections. It starts with a brief introduction of background approaches, such as SVM, TWSVM and LSTSVM, mentioned in Section 2. The formulations of proposed MLTSVM classifier for linear and non-linear cases are discussed in Section 3. Section 4 highlights the experimental work of proposed approach and section 5 contains concluding remarks.

2. Background

This section provides the brief introduction of traditional SVM, TWSVM and LSTSVM. The format of training set for binary classification is given below:

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\} \qquad (1)$$

Where x_i represents the ith data sample or pattern in n-dimensional real space R and $y_i \in \{+1,-1\}$ represents class label. 'l' represents number of patterns in training dataset. Suppose positive and negative class contains l_1 and l_2 patterns correspondingly and $l=l_1 + l_2$.

2.1 Support Vector Machine

SVM uses following decision function to classify the patterns:

$$f(x) = \mathrm{sgn}((w.x) + b) \qquad (2)$$

SVM divides the patterns of two classes by constructing a hyper-plane that provides clear separation between them. The equation of hyper-plane is given below:

$$w.x + b=0 \qquad (3)$$

The above hyper-plane lies in between following planes:

$$w^T.x + b = 1 \text{ and } w^T.x + b = -1 \qquad (4)$$

Where $w \in R^n$ is a normal vector in n-dimensional real space R and $b \in R$ is a bias term. SVM solves following QPP in order to obtain the value of normal vector and bias:

$$\min_{w,b,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \quad \text{s.t.}$$
$$y_i((w.x_i) + b) \geq 1 - \xi_i \quad \text{and } \xi_i \geq 0 \qquad (5)$$

Where i=1,…,l, and the notation ξ_i and C>0 denote slack variables and penalty parameter respectively. Slack variables determine the degree of misclassification of data sample. The above QPP is solved by taking its dual form. In SVM, all patterns provide constraint to QPP *i.e.*, SVM dual formulation depends upon the number of all patterns in the training set. For 'l' training patterns, the complexity of SVM is $O(l^3)$ [4]. Figure 1 shows the geometric representation of the binary SVM.

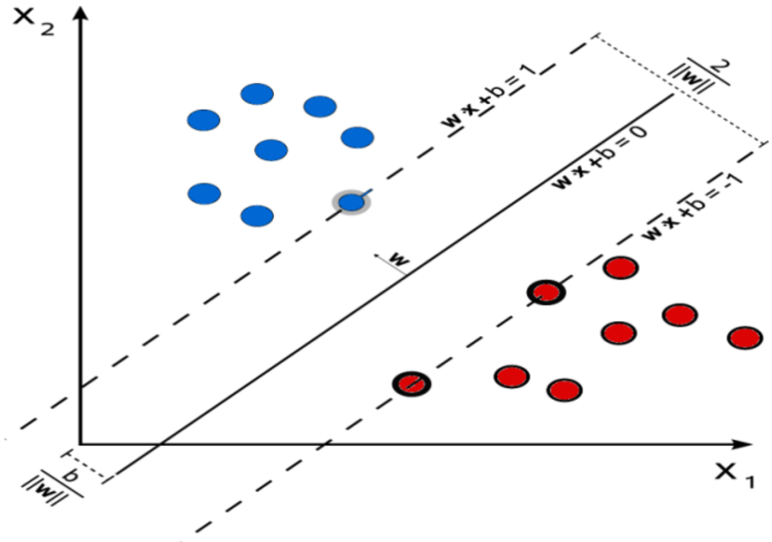

Figure 1. Geometric Representation of Binary Support Vector Machine

2.2. Twin Support Vector Machine

TWSVM uses following decision function in order to classify the patterns of two classes:

$$f(x) = \arg\min_{i=1,2} \frac{|w_i.x + b_i|}{\|w_i\|} \quad (6)$$

TWSVM performs the classification task by generating two hyper-planes which are not parallel but obtained by optimizing a pair of QPPs as:

$$\min(w_1, b_1, \xi) \quad \frac{1}{2}\|X_1 w_1 + e_1 b_1\|^2 + c_1 e_2^T \xi$$

$$\text{s.t.} \quad -(X_2 w_1 + e_2 b_1) + \xi \geq e_2, \xi \geq 0 \quad (7)$$

$$\min(w_2, b_2, \eta) \quad \frac{1}{2}\|X_2 w_2 + e_2 b_2\|^2 + c_2 e_1^T \eta$$

$$\text{s.t.} \quad (X_1 w_2 + e_1 b_2) + \eta \geq e_1, \eta \geq 0 \quad (8)$$

Where matrices $X_1 \in R^{l_1 \times n}$ and $X_2 \in R^{l_2 \times n}$ include the patterns of positive and negative class correspondingly, c_1 , c_2 >0 are penalty parameters for misclassified samples, $e_1 \in R^{l_1}$ and $e_2 \in R^{l_2}$ are the vectors of 1's and $\xi \in R^{l_2}$ and $\eta \in R^{l_1}$ are slack variables due to negative and positive class correspondingly. TWSVM determines the following two non-parallel hyper-planes in n-dimensional space:

$$x^T w_1 + b_1 = 0 \quad \text{and} \quad x^T w_2 + b_2 = 0 \quad (9)$$

TWSVM solves two smaller size QPPs in which patterns of one of the two classes provide constraints to it. If number of patterns in each class is approximately equal to l/2, then the complexity of TSVM is $O(2\times(l/2)^3)$ which is four times faster than that of traditional SVM [22]. Figure 2 shows the geometric representation of the binary Twin Support Vector Machine.

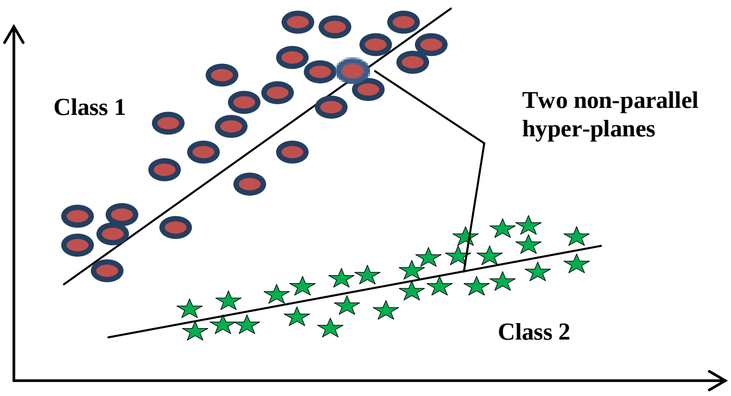

Figure 2. Geometric Representation of Binary Twin Support Vector Machine

2.3. Least Squares Twin Support Vector Machine

LSTSVM constructs two-non parallel hyper-planes by optimizing a pair of linear equations instead of a pair QPPs as:

$$\min_{(w_1,b_1,\xi)} \frac{1}{2}\|X_1w_1 + e_1b_1\|^2 + \frac{c_1}{2}\xi^T\xi$$
$$\text{s.t.} \quad -(X_2w_1 + e_2b_1) + \xi = e_2 \tag{10}$$

and

$$\min_{(w_2,b_2,\eta)} \frac{1}{2}\|X_2w_2 + e_2b_2\|^2 + \frac{c_2}{2}\eta^T\eta$$
$$s.t. \quad (X_1w_2 + e_1b_2) + \eta = e_1 \tag{11}$$

LSTSVM solves a pair of linear equations rather than QPPs due to equality constraints as opposed to inequality constraints as in TWSVM. After solving above equations, we can calculate the parameters of hyper-plane as:

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -\left(G^TG + \frac{1}{c_1}H^TH\right)^{-1} G^Te_2 \tag{12}$$

and

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = \left(H^TH + \frac{1}{c_2}G^TG\right)^{-1} H^Te_1 \tag{13}$$

Where, H= $[X_1 \ \ e_1]$ and G= $[X_2 \ \ e_2]$. Further, hyper-plane parameters (w_1,b_1) and (w_2,b_2) are helpful to generate two non-parallel planes by using equation 9. A class is assigned to a new pattern depending upon which of the plane lies nearest to it as:

$$f(x) = \arg\min_{i=+1,-1} \frac{|w_i x + b_i|}{\|w_i\|} \tag{14}$$

Where | . | denotes the perpendicular distance of the pattern from the plane. LSTSVM also classifies the non-linearly separable patterns by using kernel function and determines two kernel generated surfaces in higher-dimension as:

$$K(x^T, D^T)\mu_1 + \gamma_1 = 0 \text{ and } K(x^T, D^T)\mu_2 + \gamma_2 = 0 \tag{15}$$

Where 'K' is any kernel function and D=$[X_1 \ \ X_2]^T$. The optimization problems of non-linear LSTSVM are formulated as:

$$\min_{(\mu_1,\gamma_1,\xi)} \frac{1}{2}\left\|K(X_1,D^T)\mu_1 + e\gamma_1\right\|^2 + \frac{c_1}{2}\xi^T\xi$$
$$\text{s.t.} \quad -\left(K(X_2,D^T)\mu_1 + e\gamma_1\right) = e - \xi \tag{16}$$

and

$$\min_{(\mu_2,\gamma_2,\xi)} \frac{1}{2}\left\|K(X_2,D^T)\mu_2 + e\gamma_2\right\|^2 + \frac{c_2}{2}\eta^T\eta$$
$$\text{s.t.} \quad \left(K(X_1,D^T)\mu_2 + e\gamma_2\right) = e - \eta \tag{17}$$

Hyper-plane parameters are calculated as:

$$\begin{bmatrix}\mu_1\\ \gamma_1\end{bmatrix} = -(Q^TQ + \frac{1}{c_1}P^TP)^{-1}Q^Te \quad (18)$$

$$\begin{bmatrix}\mu_2\\ \gamma_2\end{bmatrix} = (P^TP + \frac{1}{c_2}Q^TQ)^{-1}P^Te \quad (19)$$

Where P= $[K(X_1, D^T) \;\; e]$, Q= $[K(X_2, D^T) \;\; e]$ and the class is evaluated as:

$$class(j) = argmin(j = 1,2)\frac{|K(x^T, D^T)\mu_j + \gamma_j|}{\|\mu_j\|} \quad (20)$$

3. Multiclass Least Squares Twin Support Vector Machine

In this classifier, each class is trained with rest of the other classes. For M-class classification problem, it constructs M-binary LSTSVM classifiers and m^{th} LSTSVM classifier treats the patterns of m^{th} class with positive class labels while the patterns of other classes with negative class labels. MLSTSVM solves M-linear programming problems and generates M hyper-planes, one plane for each class. For test pattern, its perpendicular distance is calculated from each hyper-plane and it is assigned to the class from which its distance is lesser. Suppose the patterns of the m^{th} class are denoted by the matrix $X_m \in R^{l_m \times n}$, where m=1,...,M and number of patterns of m^{th} class are indicated by l_m. The patterns of rest of the classes are represented by the following matrix:

$$Y_m = [(X_1)^T, (X_2)^T, \ldots, (X_{m-1})^T, (X_{m+1})^T, \ldots, (X_M)^T]^T \quad (21)$$

i.e., $Y_m \in R^{(l-l_m)\times n}$ contains all the patterns except m^{th} class. The formulations of MLTSVM for linear and non-linear cases are given below:

3.1. Linear MLSTSVM

Let the equation of m^{th} hyper-plane is:

$$(w_m.x)+b_m=0 \quad (22)$$

The objective function of MLSTSVM is formulated as:

$$\min(w_m, b_m, \xi_m, \xi_m^*) \quad \frac{1}{2}c_m^*(\|w_m\|^2 + b_m^2) + \frac{1}{2}\xi_m^{*T}\xi_m^* + \frac{c_m}{2}\xi_m^T\xi_m$$

$$\text{s.t.} \quad X_m w_m + e_{m1}b_m = \xi_m^*$$

$$(Y_m w_m + e_{m2}b_m) + \xi_m = e_{m2} \quad (23)$$

Where c_m>0 and $c_m^* > 0$ are penalty parameters. The patterns of m^{th} class and rest of the classes are comprised by X_m and Y_m matrix correspondingly. $e_{m1} \in R^{l_m}$ and $e_{m2} \in R^{(l-l_m)}$ are the vector of ones and ξ_m, ξ_m^* are slack variables. In the objective function of multi-class extension of LSTSVM, an extra regularization term $\frac{1}{2}c_m^*(\|w_m\|^2 + b_m^2)$ is introduced to implement the structural risk minimization principle. The second term of the objective function minimizes the empirical risk and thus tries to make the patterns of m^{th} class within the close affinity of its corresponding hyper-plane and at the same time, the patterns of rest of other classes far away from the hyper-plane. Lagrangian function of the equation 23 is obtained as follows:

$$L(w_m, b_m, \xi_m, \alpha_m) = \frac{1}{2}c_m^*(\|w_m\|^2 + b_m^2) + \frac{1}{2}\|X_m w_m + e_{m1}b_m\|^2 + \frac{c_m}{2}\xi_m^T\xi_m - \alpha_m^T((Y_m w_m + e_{m2}b_m) + \xi_m - e_{m2}) \quad (24)$$

Where α_m is a non-negative lagrangain multiplier. Following Karush-Kuhn-Tucker (KKT) conditions are obtained by differentiating the lagrangian function with respect to normal vector w_m, bias b_m, slack variable ξ_m and lagrangian multiplier α_m as:

$$\frac{\partial L}{\partial w_m} = c_m^* w_m + X_m^T (X_m w_m + e_{m1} b_m) - Y_m^T \alpha_m = 0 \quad (25)$$

$$\frac{\partial L}{\partial b_m} = c_m^* b_m + e_{m1}^T (X_m w_m + e_{m1} b_m) - e_{m2}^T \alpha_m = 0 \quad (26)$$

$$\frac{\partial L}{\partial \xi_m} = c_m \xi_m - \alpha_m = 0 \quad (27)$$

$$\frac{\partial L}{\partial \alpha_m} = (Y_m w_m + e_{m2} b_m) + \xi_m - e_{m2} = 0 \quad (28)$$

Following equation is obtained from equations 25 and 26:

$$c_m^* \begin{bmatrix} w_m \\ b_m \end{bmatrix} + \begin{bmatrix} X_m^T \\ e_{m1}^T \end{bmatrix} [X_m \ \ e_{m1}] \begin{bmatrix} w_m \\ b_m \end{bmatrix} - \begin{bmatrix} Y_m^T \\ e_{m2}^T \end{bmatrix} \alpha_m = 0 \quad (29)$$

Let us assume $A_m = [X_m \ \ e_{m1}]$ and $B_m = [Y_m \ \ e_{m2}]$ and $u_m = \begin{bmatrix} w_m \\ b_m \end{bmatrix}$. With these notations, equation 29 is reformulated as:

$$c_m^* u_m + A_m^T A_m u_m - B_m^T \alpha_m = 0 \quad (30)$$

$$u_m = \left(A_m^T A_m + c_m^* I\right)^{-1} B_m^T \alpha_m \quad (31)$$

Here I is an identity matrix of suitable dimensions. The solution of equation 27 and 28 leads to:

$$\alpha_m = c_m (e_{m2} - B_m u_m) \quad (32)$$

After putting α_m in equation 31, we achieved:

$$u_m = (B_m^T B_m + \frac{1}{c_m} A_m^T A_m + \frac{c_m^*}{c_m} I)^{-1} B_m^T e_{m2} \quad (33)$$

Normal vector and bias, obtained from equation 33, are further used to generate non-parallel hyper-planes for each class. A class is assigned to a test pattern as follows:

$$f(x) = \arg\min_{m=1,..,M} \frac{|w_m.x + b_m|}{\|w_m\|} \quad (34)$$

The perpendicular distance of the test pattern is calculated from each hyper-plane and the pattern is assigned to the class from which its distance is lesser. | . | indicates the perpendicular distance of test pattern 'x' from mth hyper-plane. Figure 3 represents the geometric representation of linear MLSTSVM for three classes in R^2. Different shapes represent the patterns of different classes. Figure shows three hyper-planes, plane 1, plane 2 and plane 3 for class 1, class 2 and class 3 correspondingly in such a way that the patterns of each class lie in the close proximity of the corresponding hyper-plane while as far as possible from other hyper-planes.

Algorithm: Linear MLSTSVM

1. For i=1 to M, where M represents number of classes in dataset.

a. Define two matrix A_i and B_i in such way that $A_i=[X_i \ \ e_{i1}]$ and $B_i=[Y_i \ e_{i2}]$ where X_i represents the patterns of ith class and Y_i represents the patterns of rest of the classes.

b. Select penalty parameters on the basis of validation.

c. Solve equations 25 to 28 and obtains weight and bias for each class by using equation 33.

2. Construct Decision Function using equation 34.

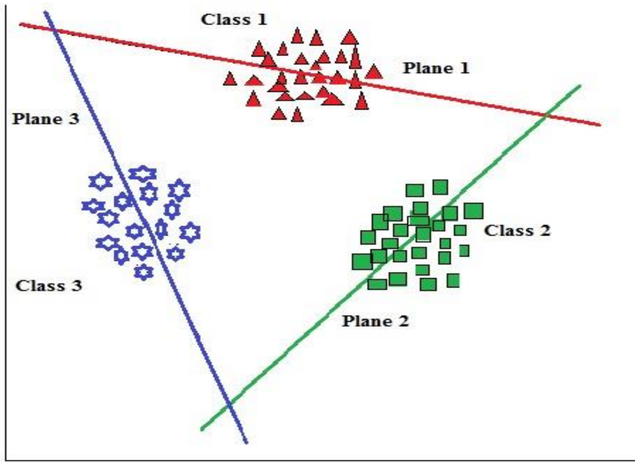

Figure 3. Geometric Representation of Linear MLSTSVM

3.2. Non-Linear MLSTSVM

Most of the data samples or patterns are not separable by linear class boundaries. For a classifier, it is important that it could be used for the classification of both linear and non-linear type of data samples. We extend the MLSTSVM for non-linear cases by utilizing the concept of kernel function. For non-linear case, firstly the input patterns are transformed into higher dimensional space by using kernel trick and then MLSTSVM classifies the patterns by generating non-linear or kernel surfaces. Let the equation of m^{th} non-linear surface is:

$$K(x, D^T)\mu_m + \gamma_m = 0 \quad \text{where } m=1,\dots, M \qquad (35)$$

Where K is a suitable kernel function and $D=[X_m\ Y_m]^T$. The non-linear MLSTSVM is formulated as:

$$\min(\mu_m,\gamma_m,\xi_m) \quad \frac{1}{2}c_m^*(\|\mu_m\|^2+\gamma_m{}^2)+\frac{1}{2}\xi_m{}^{*T}\xi_m{}^*+\frac{c_m}{2}\xi_m{}^T\xi_m$$
$$\text{s.t.} \quad K(X_m,D^T)\mu_m+e_{m1}\gamma_m=\xi_m{}^*$$
$$\left(K(Y_m,D^T)\mu_m+e_{m2}\gamma_m\right)+\xi_m=e_{m2} \qquad (36)$$

Lagrangian of the above equation is obtained as follows:

$$L(\mu_m,\gamma_m,\xi_m,\alpha_m)=\frac{1}{2}c_m^*(\|\mu_m\|^2+\gamma_m{}^2)+\frac{1}{2}\left\|K(X_m,D^T)\mu_m+e_{m1}\gamma_m\right\|^2+\frac{c_m}{2}\xi_m{}^T\xi_m -$$
$$\alpha_m^T(\left(K(Y_m,D^T)\mu_m+e_{m2}\gamma_m\right)+\xi_m-e_{m2}) \qquad (37)$$

KKT conditions of above Lagrangian function are given below:

$$\frac{\partial L}{\partial \mu_m}=c_m^*\mu_m+K\left(X_m,D^T\right)^T\left(K\left(X_m,D^T\right)\mu_m+e_{m1}\gamma_m\right)-K\left(Y_m,D^T\right)^T\alpha_m=0 \quad (38)$$

$$\frac{\partial L}{\partial \gamma_m}=c_m^*\gamma_m+e_{m1}^T\left(K\left(X_m,D^T\right)\mu_m+e_{m1}\gamma_m\right)-e_{m2}^T\alpha_m=0 \qquad (39)$$

$$\frac{\partial L}{\partial \xi_m}=c_m\xi_m-\alpha_m=0 \quad (40)$$

$$\frac{\partial L}{\partial \alpha_m}=\left(K(Y_m,D^T)\mu_m+e_{m2}\gamma_m\right)+\xi_m-e_{m2})=0 \qquad (41)$$

Following equation is obtained by combining equation 38 and 39:

$$c_m^*\begin{bmatrix}\mu_m\\ \gamma_m\end{bmatrix}+\begin{bmatrix}K\left(X_m,D^T\right)^T\\ e_{m1}^T\end{bmatrix}\left[K\left(X_m,D^T\right)\ e_{m1}\right]\begin{bmatrix}\mu_m\\ \gamma_m\end{bmatrix}-\begin{bmatrix}K\left(Y_m,D^T\right)^T\\ e_{m2}^T\end{bmatrix}\alpha_m=0 \quad (42)$$

Let $P_m = [K(X_m, D^T)\ \ e_{m1}]$ and $Q_m = [K(Y_m, D^T)\ \ e_{m2}]$. Equation 42 is reformulated as:

$$c_m^* \begin{bmatrix}\mu_m \\ \gamma_m\end{bmatrix} + P_m^T P_m \begin{bmatrix}\mu_m \\ \gamma_m\end{bmatrix} - Q_m^T \alpha_m = 0 \qquad (43)$$

After solving given equations 40, 41 and 43, the kernel-generated surface parameters can be obtained as follows:

$$\begin{bmatrix}\mu_m \\ \gamma_m\end{bmatrix} = (Q_m^T Q_m + \frac{1}{c_m} P_m^T P_m + \frac{c_m^*}{c_m} I)^{-1} Q_m^T e_{m2} \qquad (44)$$

For new pattern, its perpendicular distance is measured from each non-linear surface and pattern is assigned to the class from which its distance is lesser. The decision function for non-linear MLSTSVM is formulated as:

$$f(x) = \arg\min_{m=1,..,M} \frac{|\mu_m . K(x, D^T) + \gamma_m|}{\|\mu_m\|} \qquad (45)$$

If x_i and x_j represent vectors in the input space, then the RBF kernel function is defined as:

$$K_G(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\,\sigma^2}) \qquad (46)$$

Algorithm: Non-Linear MLSTSVM
1. Select a kernel function K.
2. For i=1 to M, where M represents number of classes in dataset.
a. Define two matrix P_i and Q_i in such way that $P_i=[K(X_i, D^T)\ \ e_{i1}]$ and $Q_i=[K(Y_i, D^T)\ \ e_{i2}]$ where X_i represents the pattern of ith class and Y_i represents the data sample of rest of the classes.
b. Select penalty parameters on the basis of validation.
c. Solve equations 38 to 41 and obtains weight and bias for each class using equation 43.
3. Classify new data sample by using equation 45.

Let each class in MLSTSVM contains approximately $(l/_M)$ patterns. MLSTSVM generates a hyper-plane for every class by training it with rest of the classes where patterns of the other classes provide constraint to it. So, the linear equation includes about $(M-1)l/_M$ constraints. Hence, the computational complexity of linear MLSTSVM is about $O(M\left((M-1)\,l/_M\right)^3) \cong O((M-1)^3 l^3/_{M^2})$ because it contains M linear equations like equation 23. MLSTSVM solves linear equations for both linear and non-linear cases whereas multi class TWSVM approaches solve quadratic equations. So, MLSTSVM is a simple and faster approach for multi-class classification.

4. Experimental Results

Here, we discuss the results of experiment of proposed MLSTSVM classifier on twelve benchmark datasets-Iris, Wine, Glass, Ecoli, Balance, Hayes-Roth, Dermatology, Pen Based, Page Block, Contraceptive, Thyroid and Shuttle. The datasets are taken from UCI Repository of machine learning database [37]. Table 1 presents the details of these datasets as follows:

Table 1. Details of Datasets

Datasets	# Samples	#Attributes	#Class
Iris	150	4	3
Wine	178	13	3
Glass	214	13	6
Ecoli	327	7	5
Balance	625	4	3
Hayes-Roth	132	5	3
Dermatology	358	34	6
Pen Based	1100	16	10
PageBlock	548	10	5
Contraceptive	1473	9	3
Thyroid	215	5	3
Shuttle	2175	9	5

In this experiment, 10-fold cross validation approach is used to evaluate the performance of the proposed classifier along with the other existing classifiers such as Multi-SVM, 1-versus-rest Twin Support Vector Machine (1-v-r TWSVM) [32], Multiple Birth Support Vector Machine (MBSVM)[33], Twin KSVC [34]. All these approaches are implemented in Matlab R2012a on Windows 7 with Intel core i-7 processor (3.4 GHz) with 12-GB RAM. The performance of proposed classifier is evaluated for both linear and non-linear cases. In the usual several kernel functions, the RBF kernel function can non-linearly transform data samples into higher dimensional space. The poly kernel function has more hyper-parameters than the RBF kernel function and sigmoid kernel function is not valid for some parameters. Therefore, in our research work, we have selected RBF kernel function to handle non-linear cases.

4.1. Parameters Selection

The performance of MLSTSVM also depends upon the selection of parameters. For this purpose, Grid Search approach is used for the suitable parameters selection. These parameters are sigma(σ) for RBF kernel function and penalty parameters c_m and c_m^*. Parameters are selected by using 10-fold cross validation from the following range: $c_m, c_m^* \in \{10^{-8}, \ldots, 10^{3}\}$, sigma $\in \{2^{-4}, \ldots, 2^{7}\}$. For non-linear case, we set $c_m = c_m^*$ in order to reduce the computational complexity of parameters selection. Figures 4-9 show the impact of parameters on the performance of proposed classifier on three datasets (Wine, Ecoli and Glass). Figure 4 shows the influence of penalty parameters (cm,cm*) on the performance of MLSTSVM classifier for Wine dataset. It is observed from the Figure that the impact of cm on the predictive accuracy of the proposed classifier is more as compared to cm*. For high value of cm, the performance of MLSTSVM suddenly degrades. Figure 5 depicts the influence of penalty parameter and sigma on the predictive accuracy of non-linear MLSTSVM for Wine dataset. For non-linear case, penalty parameters are set to equal ($c_m = c_m^*$) to ease the parameters selection process. From Figure 5, it is clear that the effect of penalty parameters on the classifier's performance is more as compared to sigma. The proposed classifier has shown better accuracy for low value of penalty parameters and high value of sigma.

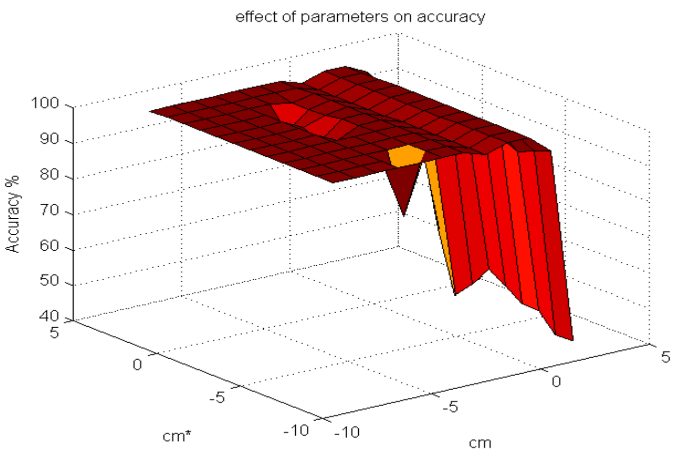

Figure 4. Influence of the Penalty Parameters on Linear MLSTSVM for Wine Dataset

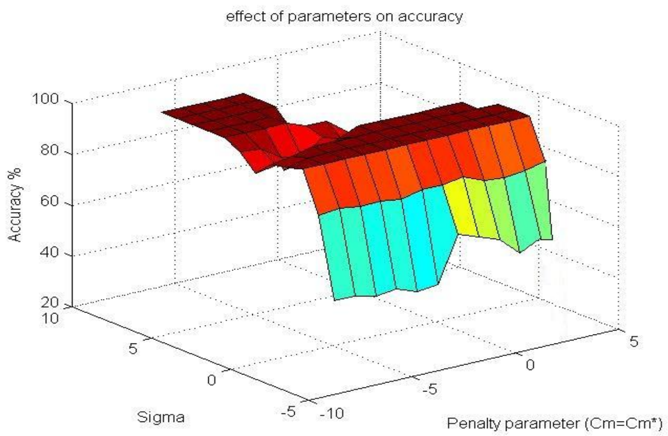

Figure 5. Influence of the Penalty Parameter (C_m=C_m^*) on Non-Linear MLSTSVM for Wine Dataset

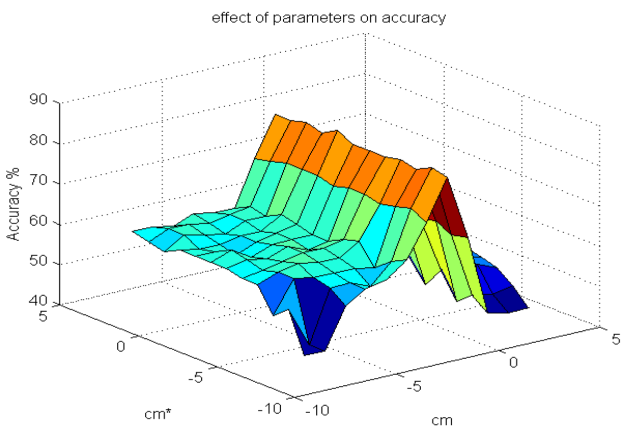

Figure 6. Influence of the Penalty Parameters on Linear MLSTSVM for Ecoli Dataset

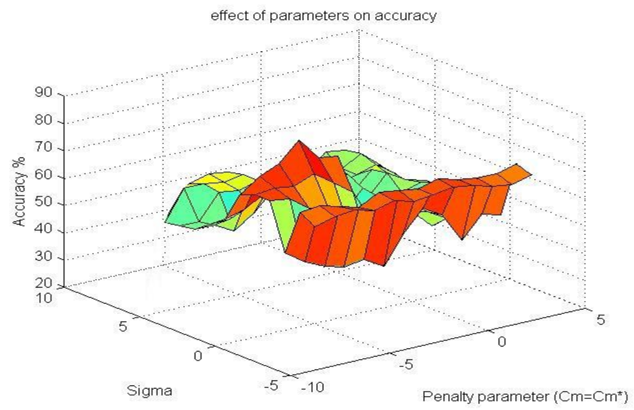

Figure 7. Influence of the Penalty Parameter (c_m=c_m^*) on Non-Linear MLSTSVM for Ecoli Dataset

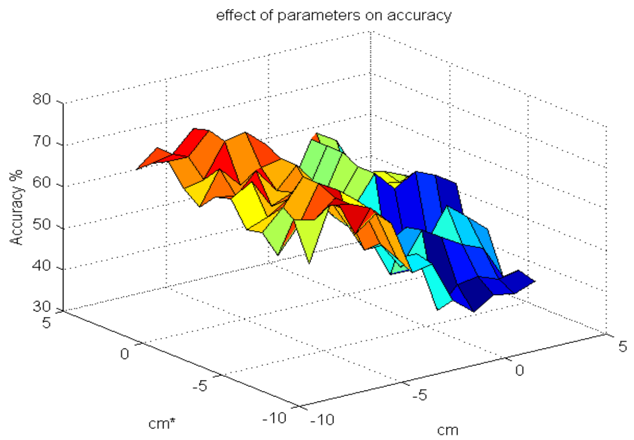

Figure 8. Influence of the Penalty Parameters on Linear MLSTSVM for Glass Dataset

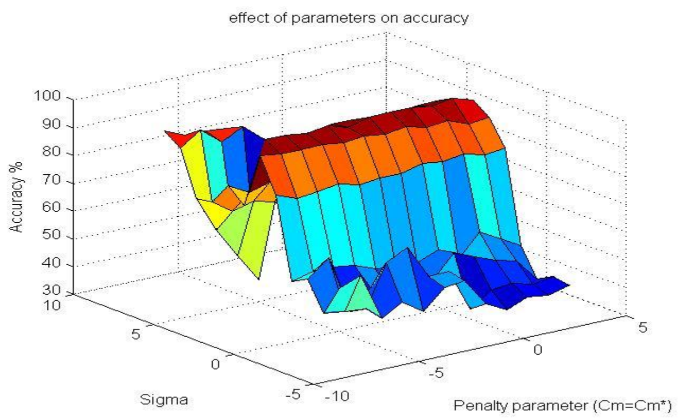

Figure 9. Influence of the Penalty Parameter (c_m=c_m^*) on Non-Linear MLSTSVM for Glass Dataset

Figure 6-9 show the impact of parameters on the classifier for Ecoli and Glass dataset respectively. For Glass dataset, the impact of sigma is more on the performance of non-linear MLSTSVM classifier. From these Figures, it is clear that the selection of penalty and kernel parameters affects the performance of MLSTSVM. Therefore, the appropriate choice of these parameters is one major issue of concern.

4.2. Result and Discussion

Table 2 and 3 gives the performance comparison in terms of accuracy and time of proposed classifier with existing classifiers, for example, Multi-SVM, 1-v-r TWSVM, MBSVM and Twin KSVC on the given twelve datasets. Accuracy refers to the correct classification rate and is calculated by taking the average testing accuracies. Time denotes the average value of the time (including training and testing time) recorded for ten experiments. The experimental results indicate that the proposed linear MLSTSVM classifier has shown better performance for Iris, Wine, Ecoli, Dermatology, Pen Based, Page Block, Contraceptive, Thyroid and Shuttle datasets as compared to the existing approaches, while on the rest of datasets the results of linear MLTSVM are comparable with other approaches.

Table 2. Performance Comparison on Benchmark Datasets for Linear Classifiers

Datasets	Multi- SVM (C) Acc±std(%) Time(s)	1-v-r TWSVM (C) Acc±std(%) Time(s)	MBSVM (C) Acc±std(%) Time(s)	Twin-KSVC (c_1, c_2) Acc±std(%) Time(s)	MLSTSVM (c_m, c_m^*) Acc±std(%) Time(s)
Iris	10^{0} 95.72±3.37 0.048	10^{-3} 94.35±3.62 0.0096	10^{-2} 96.12±3.08 0.007	10^{-3},10^{-3} 94.24±3.26 0.03024	10^{-3},10^{-2} **96.87±3.54** **0.0052**
Wine	10^{-1} 95.83±3.62 0.097	10^{0} 96.22±1.93 0.026	10^{0} 97.86±1.82 0.014	10^{-3},10^{-3} 97.09±2.03 0.0102	10^{-4},10^{-1} **100±0.0** **0.0082**
Glass	10^{-2} 72.06±11.85 1.35	10^{-3} 72.86±12.2 0.054	10^{-3} **75.31±9.13** 0.052	10^{-3},10^{-3} 70.89±8.83 0.088	10^{-5},10^{1} 75.11±7.53 **0.0078**
Ecoli	10^{-2} 74.64±8.48 6.46	10^{-2} 71.26±6.83 1.35	10^{-1} 74.77±6.13 1.127	10^{0},10^{-1} 71.62±5.66 2.08	10^{-1},10^{-6} **82.46±3.64** **0.08112**
Balance	10^{0} 82.43±6.02 5.36	10^{1} 85.43±4.81 0.358	10^{1} 87.92±5.86 0.2540	10^{-1},10^{-1} **88.01±4.35** 0.3280	10^{1},10^{0} 87.53±2.58 **0.0702**
Hayes-Roth	10^{0} 67.43±9.74 1.216	10^{-2} 68.07±9.23 0.108	10^{-1} 69.20±9.11 0.0096	10^{0},10^{-1} **74.48±9.83** 0.01083	10^{-1},10^{-1} 69.45±8.34 **0.00624**
Dermatology	10^{-3} 87.18±4.84 2.78	10^{-5} 84.08±4.03 0.0936	10^{-4} 86.69±2.13 0.09775	10^{-4},10^{-4} 84.06±4.12 0.108	10^{-4},10^{-5} **92.48±3.32** **0.0802**
Pen Based	10^{0} 81.73±4.73 4.509	10^{-1} 84.75±3.11 0.5612	10^{-1} 87.79±1.54 0.1928	10^{0},10^{-1} 82.18±2.62 0.1286	10^{-1},10^{-1} **88.23±2.24** **0.1088**
Page Block	10^{-2} 73.84±8.69 6.67	10^{-3},10^{-2} 84.13±6.71 4.02	10^{-3} 80.28±6.04 1.1499	10^{-1},10^{-1} 79.09±6.28 1.562	10^{-2},10^{-1} **92.14±4.6** **0.05148**
Contraceptive	10^{-2} 38.89±4.7 3.84	10^{-2} 45.23±3.9 0.6728	10^{-2} 43.45±3.25 0.4022	10^{-1}, 10^{-1} 39.91±3.7 0.5621	10^{-2}, 10^{-2} **49.53±3.2** **0.1076**
Thyroid	10^{0} 92.17±3.28 0.347	10^{-2} 96.54±2.27 0.011	10^{-1} 97.03±2.01 0.0093	10^{-2}, 10^{-2} 94.50±2.13 0.018	10^{-3}, 10^{0} **100±0.0** **0.00468**
Shuttle	10^{0}	10^{0}	10^{0}	10^{0}, 10^{0}	10^{-1}, 10^{-1}

	72.39±6.03 5.0193	78.85±4.12 1.34	82.07±3.4 1.118	79.23±4.23 0.9577	**95.11±1.2** **0.255**

For non-linear cases, the proposed classifier gives better predictive accuracy for Wine, Glass, Ecoli, Balance, Dermatology, Pen Based, Page Block, Contraceptive, Thyroid and Shuttle datasets. Bold value indicates the better performance of a classifier in terms of accuracy and time. Since the performance of MLSTSVM is better than that of other four classifiers for 9 out of 12 datasets, therefore it has better generalization ability. It is also analyzed that the proposed classifier takes lesser computation time on almost all type of datasets as compared to other methods. Hence, it could be concluded that MLSTSVM has highest computation efficiency.

Table 3. Performance Comparison on Benchmark Datasets for Non-Linear Classifiers

Datasets	Multi- SVM (C, σ) Acc±std(%) Time(s)	1-v-r TWSVM (C,σ) Acc±std(%) Time(s)	MBSVM (C,σ) Acc±std(%) Time(s)	Twin-KSVC ($c_1 = c_2, \sigma$) Acc±std(%) Time(s)	MLSTSVM ($c_m = c_m^*, \sigma$) Acc±std(%) Time(s)
Iris	$10^0,2^2$ 97.33±3.52 3.6	10^0, 2^3 96.47±3.26 1.19	$10^0,2^1$ 98.00±2.26 0.761	$10^0,2^2$ **98.13±2.66** 0.94	$10^{-1},2^3$ 97.85±2.43 **0.1154**
Wine	$10^0,2^1$ 98.71±1.36 0.8916	10^0, 2^2 98.38±1.02 0.93	$10^{-4},2^2$ 98.24±1.08 0.119	$10^{-3},2^6$ 97.75±3.24 1.10	$10^{-1},2^0$ **100±0.0** **0.092**
Glass	$10^{-4},2^0$ 71.03±6.02 6.32	$10^{-5},2^7$ 78.56±5.4 2.14	$10^{-6},2^1$ 69.52±5.15 1.93	$10^0,2^2$ 63.21±4.83 2.56	$10^{-4},2^2$ **98.02±2.32** **0.2418**
Ecoli	$10^{-6},2^1$ 83.35±6.28 3.28	$10^{-5},2^6$ 78.22±5.24 1.37	$10^{-6},2^7$ 82.01±5.17 0.972	$10^{-1},2^4$ 86.36±4.5 1.01	$10^{-4},2^1$ **87.42±4.77** **0.3915**
Balance	$10^{-2},2^4$ 84.59±6.67 7.49	$10^{-1},2^2$ 90.21±4.4 2.28	$10^{-6},2^2$ 89.22±5.88 2.05	$10^{-1},2^6$ 90.33±5.37 2.46	$10^{-7},2^3$ **94.87±3.4** **0.9702**
Hayes-Roth	$10^{-3},2^4$ 72.30±4.08 5.89	$10^{-4},2^3$ 72.43±3.63 1.393	$10^{-4},2^4$ 73.35±3.26 1.672	$10^{-3},2^3$ **81.10±3.95** 1.584	$10^{-3},2^3$ 75.56±3.34 **1.369**
Dermatology	$10^{-3},2^5$ 89.53±5.24 8.33	$10^{-5},2^7$ 87.82±4.08 1.53	$10^{-4},2^4$ 88.75±4.66 1.568	$10^{-5},2^7$ 84.26±2.01 1.4	$10^{--6},2^7$ **91.54±3.8** **0.6336**
Pen Based	$10^{-3},2^3$ 86.27±4.84 26.34	$10^{-5},2^4$ 87.11±4.34 14.72	$10^{-4},2^4$ 88.89±4.22 12.74	$10^{-4},2^3$ 88.56±4.67 12.63	$10^{-6},2^5$ **96.96±2.14** **10.48**
Page Block	$10^{-1},2^0$ 79.67±6.56 9.88	$10^{-1},2^1$ 81.18±5.04 6.65	$10^{-1},2^1$ 85.43±5.83 2.08	$10^{-1},2^0$ 82.39±5.69 2.35	$10^0,2^{-1}$ **90.19±4.64** **1.106**
Contraceptive	$10^{-1},2^3$ 42.01.89±6.02 26.02	$10^{-3},2^4$ 47.11±4.82 9.53	$10^{-2},2^5$ 48.18±5.16 7.47	10^{-3}, 2^4 43.58±4.77 8.39	10^{-2}, 2^3 **50.62±4.62** **5.6**
Thyroid	$10^{-2},2^2$ 96.11±2.15 1.64	$10^{-3},2^1$ 97.98±1.07 0.83	$10^{-3},2^0$ 98.95±1.23 0.365	10^{-3}, 2^1 98.59±1.13 0.572	10^{-4}, 2^1 **100±0.0** **0.148**
Shuttle	$10^{-5},2^1$ 75.86±4.18 18.17	$10^{-5},2^1$ 82.69±3.24 4.02	$10^{-4},2^3$ 85.8±3.53 2.5	10^{-4}, 2^2 82.48±3.66 **2.413**	10^{-5}, 2^1 **94.42±2.68** 2.46

4.3. Statistical Comparison of Classifiers

In this study, we have used Friedman's test statistic to analyze the performances of classifiers. Friedman's test assigns rank to each classifier according to their accuracy for each dataset individually [38-40]. It is a seven step process as follows:

Step 1. Set two hypotheses:

H0: There is no difference between the classifiers.

H1: There is a difference between the classifiers.

Step 2. Select the value of α. In this study, we select α=0.05.

Step 3. Calculate degree of freedom (df) as:

df=M-1=5-1=4 where M is number of classifiers.

Step 4. Critical value from chi-square Table for 4 degree of freedom is 9.48773 [41].

Step 5. If Friedman test statistic is greater than 9.48773 than reject the null hypothesis.

Step 6. Friedman test statistic is calculated as [30]:

$$\text{Friedman Test Statistic}=\chi_F^2 = \frac{12D}{M(M+1)}\left[\sum_{j=1}^{M} AR_j^2 - \frac{M(M+1)^2}{4}\right] \quad (47)$$

Where
$$AR_j = \frac{1}{D}\sum_{i=1}^{D} r_i^j \quad (48)$$

Here, D denotes the number of datasets used in this research, $AR_j$is the average rank of j-th classifier. r_i^j is the rank of j[th] classifier on ith dataset. Friedman Test Statistic for linear and non-linear classifiers is 25.87 and 22.43 respectively.

Step 7. Since the value of Friedman Test statistic for both linear and non-linear cases are greater than the critical value. Therefore, we reject the null hypothesis.

Since, the Friedman test rejects the null hypothesis, so there is a difference between classifiers. Nemenyi post hoc test on individual classifiers gives out any significant difference between them [42]. The Critical Difference (CD) in this test is defined as:

$$CD = q_\alpha\sqrt{\frac{K(K+1)}{6D}} \quad (49)$$

Where q_α is based on the Studentized range statistic. If the average rank of two or more classifiers differ by at least CD, we conclude that their performance are significantly different. Critical values for the two-tailed Nemenyi post hoc test after the Friedman test are shown by Table 4 [38].

Table 4. Critical Values for the Two-Tailed Nemenyi Post Hoc Test [38]

#Classifiers	2	3	4	5	6	7	8	9	10
$q_{0.05}$	1.960	2.343	2.569	2.728	2.850	2.949	3.031	3.102	3.164
$q_{0.10}$	1.645	2.052	2.291	2.459	2.589	2.693	2.780	2.855	2.920

Critical value $q_{0.05}$ for 5 classifiers is 2.728. So, critical difference CD is:

$$\text{CD}=2.728\sqrt{\frac{5\times 6}{6\times 12}} = 1.76$$

A modified version of Demsar diagram is used to depict the results of Friedman's test statistic and Nemenyi Post hoc test. For linear and non-linear cases, the rank of each classifier is shown by Table 5 and 6. The rank of classifier is cacluated according to their predictive accuracy. From these Tables, it is observed that the proposed MLSTSVM classifier has gained highest Friedman score (average rank) among all classifiers for both linear and non-linear cases.

Table 5. Rank of Linear Classifiers on Benchmark Datasets

Datasets	Multi-SVM	1-v-r TWSVM	MBSVM	Twin KSVC	MLSTSVM
Iris	3	4	2	5	1
Wine	5	4	2	3	1
Glass	4	3	1	5	2
Ecoli	3	5	2	4	1
Balance	5	4	2	1	3
Hayes-Roth	5	4	3	1	2
Dermatology	2	4	3	5	1
Pen Based	5	3	2	4	1
Page Block	5	2	3	4	1
Contraceptive	5	2	3	4	1
Thyroid	5	3	2	4	1
Shuttle	5	4	2	3	1
Average Rank	4.3	3.5	2.25	3.58	**1.33**
Friedman test statistic=25.87					

Table 6. Rank of Non-Linear Classifiers on Benchmark Datasets

Datasets	Multi-SVM	1-v-r TWSVM	MBSVM	Twin KSVC	MLSTSVM
Iris	4	5	2	1	3
Wine	2	3	4	5	1
Glass	3	2	4	5	1
Ecoli	3	5	4	2	1
Balance	5	3	4	2	1
Hayes-Roth	5	4	3	1	2
Dermatology	2	4	3	5	1
Pen Based	5	4	2	3	1
Page Block	5	4	2	3	1
Contraceptive	5	3	2	4	1
Thyroid	5	4	2	3	1
Shuttle	5	3	2	4	1
Average Rank	4.08	3.67	2.83	3.16	**1.25**
Friedman test statistic=22.43					

In Figure 10 and 11, the y-axis shows the ascending order of classifiers according to their performance and x-axis represents the average rank of classifiers across all twelve datasets for linear and non-linear cases. The difference between the end of the best performing classifier's tail and the start of the next significantly different classifier is represented by two vertical lines. From these Figures, it is observed that there is no significant difference between MLSTSVM and MBSVM classifier while MLSTSVM perform significantly better than the 1-v-r TWSVM, Twin KSVC and Multi SVM classifiers with values of 3.5, 3.58 and 4.3 respectively for linear cases and 3.67, 3.16 and 4.08 respectively for non-linear cases. In each case, the proposed MLSTSVM is the best performing classifier. Hence, it is evident that the performance of our proposed MLSTSVM classifier is better than the already existing multi-classifiers based on SVM and TWSVM.

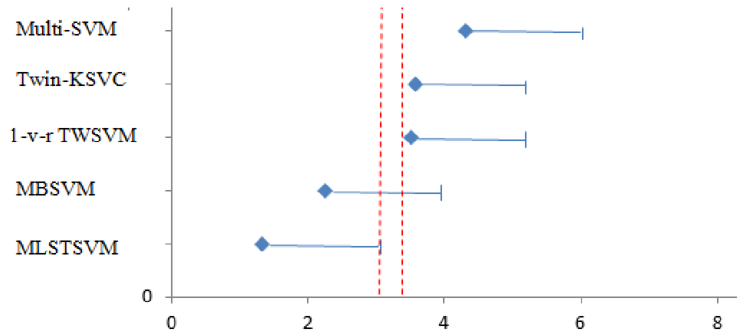

Figure 10. Average Rank Comparisons of Classifiers for Linear Cases

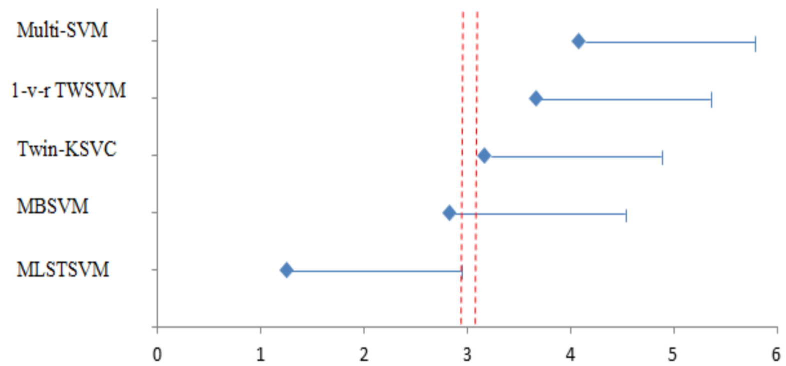

Figure 11. Average Rank Comparisons of Classifiers for Non-Linear Cases

5. Conclusion

This research work proposes a novel multi-classifier, termed as MLSTSVM which is the extension of binary LSTSVM. Due to equality constraints, MLSTSVM solves only linear equations rather than complex QPPs as compared to typical multi-classifiers based on TWSVM, which make it simple and faster. In this classifier, the patterns of each class are trained with the patterns of rest of the classes and generate non-parallel hyper-plane for each class. A test pattern is classified on the basis of minimum distance criteria. From the experimental results, it is observed that the MLSTSVM classifier yields the highest prediction accuracy for most of cases and takes lesser computational time as compared to the other classifiers.

The future work is to investigate the performance of MLSTSVM classifier with real world data. Apart from this, parameter selection is a practical problem which should be addressed in the future.

References

[1] C. Corinna and V. Vapnik, “Support-vector networks”, Machine learning, vol. 20, no. 3, **(1995)**, pp. 273-297.

[2] V. Vladimir, “The nature of statistical learning theory”, Springer, **(2000)**.

[3] C. Nello and J. S. Taylor, “An Introduction to Support Vector Machines, and other kernel-based learning methods”, Cambridge University Press, **(2000)**.

[4] C. Nello and J. S. Taylor, “An Introduction to Support Vector Machines”, Cambridge University Press, **(2000)**.

[5] H. C. Wei and C. J. Lin, “A comparison of methods for multiclass support vector machines”, Neural Networks, IEEE Transactions on, vol. 13, no. 2, **(2002)**, pp. 415-425.

[6] Dietterich T. G. and G. Bakiri, “Solving multiclass learning problems via error-correcting output codes”, Journal of Artificial Intelligence Research, vol. 2, **(1995)**, pp.263–286.

[7] Platt J. C., N. Chrisianini and J. S. Taylor, "Large Margin DAGs for Multiclass Classification", In Advances in Neural Information Processing Systems 12. Cambridge, MA: MIT Press, **(2000)**, pp. 547-553.
[8] Kerebel Ulrich H. G., "Pairwise Classification and Support Vector Machine", In Advances in kernel methods, MIT Press, **(1999)**, pp. 255-268.
[9] L. Yoonkyung, Y. Lin and G. Wahba, "Multi-category support vector machines", Computer Sci. Stat 33, **(2001)**, pp. 498–512.
[10] W. Jason and C. Watkins, "Multi-class support vector machines", Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway. University of London, UK, **(1998)**.
[11] C. Koby and Y. Singer, "On the learnability and design of output codes for multiclass problems", Machine Learning, vol. 47, **(2002)**, pp. 201–233.
[12] H. Trevor and R. Tibshirani, "Classification by pairwise coupling", The annals of statistics, vol. 26, no. 2, **(1998)**, pp. 451-471.
[13] N. Hassiba and Y. Chibani, "Multi-class SVMs based on fuzzy integral mixture for handwritten digit recognition", In Geometric modeling and imaging—new trends, **(2006)**, pp. 145–149.
[14] Gunn S. R., "Support vector machines for classification and regression", ISIS technical report vol. 14, **(1998)**.
[15] Suykens Johan A. K., J. D. Brabanter, L. Lukas and J. Vandewalle, "Weighted least squares support vector machines: robustness and sparse approximation", Neuro-computing, vol. 48, no. 1, **(2002)**, pp. 85-105.
[16] H. Weimin and L. Shen, "Weighted support vector regression algorithm based on data description", In Computing, Communication, Control, and Management, 2008. CCCM'08. ISECS International Colloquium on, IEEE, vol. 1, **(2008)**, pp. 250-254.
[17] A. Sonali, "Weighted support vector regression approach for remote healthcare monitoring", In Recent Trends in Information Technology (ICRTIT), 2011 International Conference on, IEEE, **(2011)**, pp. 969-974.
[18] X. Zhenxia and W. Liu, "A fuzzy rough support vector regression machine", In Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on, IEEE, **(2012)**, pp. 840-844.
[19] T. Divya, R. Arya and S. Agarwal, "Prediction of profitability of industries using weighted SVR", International Journal on Computer Science and Engineering, vol. 3, no. 5, **(2011)**, pp. 1938-1945.
[20] C. J. Xing, "Support vector regression based on optimal training subset and adaptive particle swarm optimization algorithm", Applied Soft Computing, vol. 13, no. 8, **(2013)**, pp. 3473-3481.
[21] Mangasarian O. L. and Edward W. W., "Multi-surface proximal support vector classification via generalized eigenvalues", IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 28, no. 1, **(2006)**, pp.69–74.
[22] J. R. Khemchandani and S. Chandra, "Twin support vector machine for pattern classification", IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 29, no. 5, **(2007)**, pp. 905–910.
[23] T. Divya and S. Agarwal, "Twin Support Vector Machine: A review from 2007 to 2014", Egyptian Informatics Journal, vol. 16, no. 1, **(2015)**, pp. 55-69.
[24] T. Divya and S. Agarwal, "Hybrid Feature Selection Based Weighted Least Squares Twin Support Vector Machine Approach for Diagnosing Breast Cancer, Hepatitis, and Diabetes", Advances in Artificial Neural Systems, vol. 2015, **(2015)**.
[25] T. Divya and S. Agarwal, "Feature selection based least squares twin support vector machine for diagnosis of heart disease", International Journal of Bio-Science and Bio-Technology, vol. 6, no. 2, **(2014)**, pp. 69-82.
[26] T. Divya and S. Agarwal, "Predictive model for diabetic patients using hybrid twin support vector machine", 5th International Conference on advances in communication Network and Computing (CNC-2014), **(2014)**.
[27] A. Sonali and D. Tomar, "A feature selection based model for software defect prediction", International Journal of Advanced Science and Technology, vol. 65, **(2014)**, pp. 39–58.
[28] D. Xuejun, G. Zhang, Y. Ke, B. Ma and Z. Li, "High efficient intrusion detection methodology with twin support vector machines", In Information Science and Engineering, 2008. ISISE'08. International Symposium on, vol. 1, **(2008)**, pp. 560-564.
[29] J. He and S. H. Zheng, "Intrusion detection model with twin support vector machines", Journal of Shanghai Jiaotong University (Science), vol. 19, **(2014)**, pp. 448-454.
[30] T. Divya, D. Ojha and S. Agarwal, "An emotion detection system based on multi least squares twin support vector machine", Advances in Artificial Intelligence, 2014, vol. 8, **(2014)**.
[31] H. Y. Yang, X. Y. Wang, P. P. Niu and Y. C. Liu, "Image denoising using non-subsampled shearlet transform and twin support vector machines", Neural Networks, vol. 57, **(2014)**, pp. 152-165.
[32] C. Hanhan, C. Yang and X. Pu, "Efficient speaker recognition based on multi-class twin support vector machines and GMMs", In 2008 IEEE conference on robotics, automation and mechatronics, **(2008)**, pp. 348-352.
[33] Y. Z. Xia, Y. H. Shao and X. S. Zhang, "Multiple birth support vector machine for multi-class classification", Neural Computing and Applications, vol. 22, no. 1, **(2013)**, pp. 153-161.
[34] X. Yitian, R. Guo and L. Wang, "A twin multi-class classification support vector machine", Cognitive Computation, vol. 5, no. 4, **(2013)**, pp.580-588.

[35] S. Y. Hai, W. J. Chen, W. B. Huang, Z. M. Yang and N. Y. Deng, "The best separating decision tree twin support vector machine for multi-class classification", Proceeding Computer Science, vol. 17, **(2013)**, pp.1032-1038.
[36] K. M. Arun and M. Gopal, "Least squares twin support vector machines for pattern classification", Expert Systems with Applications, vol. 36, **(2009)**, pp. 7535–7543.
[37] M. Lichman, "UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]", Irvine, CA: University of California, School of Information and Computer Science, **(2013)**.
[38] D. Janez, "Statistical comparisons of classifiers over multiple data sets", The Journal of Machine Learning Research, vol. 7, **(2006)**, pp. 1-30.
[39] F. Milton, "A comparison of alternative tests of significance for the problem of m rankings", The Annals of Mathematical Statistics, vol. 11, no. 1, **(1940)**, pp. 86-92.
[40] B. Iain and C. Mues, "An experimental comparison of classification algorithms for imbalanced credit scoring data sets", Expert Systems with Applications, vol. 39, no. 3, **(2012)**, pp. 3446-3453.
[41] http://sites.stat.psu.edu/~mga/401/Tables/Chi-square-Table.pdf
[42] N. Peter, "Distribution-free multiple comparisons. Ph.D. Thesis", Princeton University, **(1963)**.

Authors

Divya Tomar, She is a research scholar in Information Technology Division of Indian Institute of Information Technology (IIIT), Allahabad, India under the supervision of Dr. Sonali Agarwal. Her primary research interests are Data Mining, Data Warehousing especially with the application in the area of Medical Healthcare. She has published more than 25 research papers in reputed international Journal and Conferences.

Sonali Agarwal, she is working as an Assistant Professor in the Information Technology Division of Indian Institute of Information Technology (IIIT), Allahabad, India. Her primary research interests are in the areas of Data Mining, Data Warehousing, E Governance and Software Engineering. Her current focus in the last few years is on the research issues in Data Mining application especially in E Governance and Healthcare. She has published more than 50 research papers in reputed international Journal and Conferences.

