A Fuzzy Based Lagrangian Twin Parametric-Margin Support Vector Machine (FLTPMSVM)

Deepak Gupta
Computer Science & Engineering
NIT Arunachal Pradesh,
Yupia, India
deepakjnu85@gmail.com

Parashjyoti Borah
Computer Science & Engineering
NIT Arunachal Pradesh,
Yupia, India
parashjyoti@hotmail.com

Mukesh Prasad
Centre of Artificial Intelligence
School of Software, FEIT
University of Technology Sydney,
Sydney, Australia
mukesh.nctu@gmail.com

***Abstract*—In the spirit of twin parametric-margin support vector machine (TPMSVM) and support vector machine based on fuzzy membership values (FSVM), a new method termed as fuzzy based Lagrangian twin parametric-margin support vector machine (FLTPMSVM) is proposed in this paper to reduce the effect of the outliers. In FLTPMSVM, we assign the weights to each data samples on the basis of fuzzy membership values to reduce the effect of outliers. Also, we consider the square of the 2-norm of slack variables to make the objective function strongly convex and find the solution of the proposed FLTPMSVM by solving simple linearly convergent iterative schemes instead of solving a pair of quadratic programming problems as in case of SVM, TWSVM, FTSVM and TPMSVM. No need of external toolbox is required for FLTPMSVM. The numerical experiments are performed on artificial as well as well known real-world datasets which show that our proposed FLTPMSVM is having better generalization performance and less training cost in comparison to support vector machine, twin support vector machine, fuzzy twin support vector machine and twin parametric-margin support vector machine.**

Keywords—Support vector machine; Twin support vector machine; Parametric-margin model; fuzzy membership; Iterative method

I. INTRODUCTION

One of the popular machine learning algorithms, support vector machine (SVM) [1] is an excellent kernel-based tool used in past few decades for wide variety of applications like text categorization [2], handwritten digit recognition [3], activity detection[4], stock exchange prediction [5], brain-computer interface [6], credit scoring [7] etc.

The computational complexity of SVM depends on solving a large sized quadratic programming problem (QPP) i.e. O(m^3) where m is the number of training data samples. This is the main disadvantage of this method for large scale datasets. Based on same principle, an efficient approach called twin support vector machine (TWSVM) has been proposed by Jayadeva et al. [8] to reduce the training cost and improve the generalization performance where it finds two non-parallel hyperplanes by solving two smaller sized QPPs instead of finding a single hyperplane by solving a single larger one in case of SVM, which results in a reduced training cost by approximately four times [8]. A least squares variant of SVM, called least squares support vector machine (LSSVM) [9], has been proposed to decrease the training cost. Mangasarian and Musicant [10] has proposed an iterative method based on an implicit Lagrangian formulation and named it Lagrangian support vector machine (LSVM). Further, Balasundaram et al. [11] proposed a new approach for training Lagrangian twin support vector machine using unconstrained convex minimization. For Heteroscedastic noise structure, recently, Hao [12] has proposed a novel approach termed as parametric-margin *v*-support vector machine (Par-*v*-SVM) which is based on *v*-support vector machine (*v*-SVM) [13]. Further, Peng [14] has proposed a novel approach, twin parametric-margin support vector machine (TPMSVM), where it solves two smaller sized QPPs instead of solving a single larger QPP as in case of Par-*v*-SVM. Hence, the training cost of TPMSVM is much lesser than Par-v-SVM.

In all the techniques discussed above, all the data samples belonging to one class contribute equally in finding the final classifier. But presence of outliers and noise in real-world datasets can effect in determining a more appropriate classifier. Hence, to lessen the effect of outliers and noise in finding the resultant classifier, a fuzzy-based SVM algorithm (FSVM) was proposed by Lin et al. [15]. In their algorithm, each training point is assigned a membership value which can be calculated using a suitable membership function depending on the nature of the problem. Samples with higher importance get a higher membership value whereas those who have less importance get a lower membership value. In due course, many variants based on FSVM have been proposed. Batuwita & Palade [16] has proposed FSVMs for class imbalance learning (FSVM-CIL) to handle the problem of class imbalance. Furthermore, Tsujinishi et al. [17] has proposed a fuzzy least squares support vector machine for multiclass problems. Similarly, Wang et al. [18] has proposed a model Bilateral-weighted FSVM (B-FSVM). To solve bankruptcy prediction problem, a new fuzzy SVM is proposed [19]. Further, a fuzzy least squares support vector machine for object tracking is proposed by Zhang et al. [20].

In this paper, a new technique is proposed, termed as fuzzy based Lagrangian twin parametric-margin support vector machine (FLTPMSVM) to handle the outlier points which uses fuzzy membership values in decision learning. To find the resultant decision classifier, our proposed method FLTPMSVM solves simple linearly convergent iterative schemes instead of solving a pair of quadratic programming problems (QPPs) as in TWSVM, FTSVM and TPMSVM. Here, we are using MOSEK toolbox to solve the QPPs in

SVM, TWSVM, FTSVM and TPMSVM. There is no need to use any external toolbox for our proposed FLTPMSVM. Hence, FLTPMSVM improves the generalization performance of the decision surface and takes less training time in comparison to others methods. Moreover, we have presented a comparative analysis of results in terms of classification accuracy and training time of SVM, TWSVM, FTSVM and TPMSVM with our method for synthetic and real-world datasets.

II. Related Work

A. Support Vector Machine (SVM)

Let us suppose X is the input matrix of training samples and Y is the label vector. Let us consider input matrices X_1 and X_2 of size $l_1 \times n$ and $l_2 \times n$, where l_1 is the number of data points belonging to positive class and l_2 denotes the number of data points belonging to the negative class such that the total number of data samples $m = l_1 + l_2$ and n is the total number of attributes. The non-linear SVM maps the sample x to a higher dimensional feature space using a mapping function $\chi\pi(.)$ and finds the hyperplane $\chi\pi(x)w + b = 0$ by solving the following formulation

$$\min \frac{1}{2}\|w\|^2 + Ce^t\xi$$

$$\text{subject to } Y(\chi\pi(X)w + eb) + \xi \ge e,\ \xi \ge 0 \qquad (1)$$

where ξ represents slack variables; C is the penalty parameters; e is a unit vector of suitable dimension.

After finding the Lagrangian formulation of equation (1) and applying the Karush-Kuhn-Tucker (K.K.T) necessary and sufficient conditions [21], the Wolfe dual of (1) is derived as

$$\max\ e^t\alpha - \frac{1}{2}\alpha^t YXX^tY\alpha$$

$$\text{subject to } \alpha^t Y\alpha = 0,\ 0 \le \alpha \le C \qquad (2)$$

$$\text{subject to } (K(X_1, D^t)w_2 + e_1b_2) + \xi_2 \ge e_1,\ \xi_2 \ge 0 \qquad (5)$$

where ξ_1, ξ_2 represent slack variables; C_1, C_2 are penalty parameters; $D = [X_1; X_2]$; e_1, e_2 are vectors of suitable dimension having all values as 1's and $K(x^t, D^t) = (k(x, x_1), \ldots, k(x, x_m))$ is a row vector in R^m, where $k(x, x_i) = \chi\pi(x)^t.\chi\pi(x_i)$ for $i = 1, \ldots, m$ is an appropriately chosen kernel.

By introducing the Lagrangian functions of problems (4) & (5) and applying the Karush-Kuhn-Tucker (K.K.T) necessary and sufficient conditions [21], the Wolfe dual of (4) and (5) are written as

$$\max\ e_2^t\alpha_1 - \frac{1}{2}\alpha_1^t H(G^tG)^{-1}H^t\alpha_1 \qquad (6)$$

$$\text{subject to } 0 \le \alpha_1 \le C_1$$

$$\max\ e_1^t\alpha_2 - \frac{1}{2}\alpha_2^t G(H^tH)^{-1}G^t\alpha_2 \qquad (7)$$

$$\text{subject to } 0 \le \alpha_2 \le C_2$$

where $G = [K(X_1, D^t)\ e_1]$, $H = [K(X_2, D^t)\ e_2]$; $\alpha_i = (\alpha_{i1}, \ldots \alpha_{il2})^t \in R^{l2}$ and $\beta_i = (\beta_{i1}, \ldots \beta_{il2})^t \in R^{l2}$ are Lagrangian multipliers for $i = 1, 2$.

We compute the values of w_1, w_2, b_1 and b_2 using the following equations as

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix}^T = (G^tG + \delta I)^{-1}H^t\alpha_1 \qquad (8)$$

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix}^T = (H^tH + \delta I)^{-1}G^t\alpha_2 \qquad (9)$$

where δ is a small positive integer value and I is an identity where $\alpha \in R^m$ is the vector of Lagrangian multipliers. A new data point $x \in R^n$ is assigned to a given class 'i' as follows

$$class\ i = sign|\chi\pi(x)^t w + b| \tag{3}$$

B. Twin Support Vector Machine (TWSVM)

In TWSVM [8], two non-parallel hyperplanes are obtained such that each of them is nearer to one of the classes and as far as possible from the other class. In non-linear case, twin support vector machine finds a pair of non-parallel hyperplanes $K(x^t, D^t)w_1 + b_1 = 0$ and $K(x^t, D^t)w_2 + b_2 = 0$ from the solution of the following QPPs

$$\min \frac{1}{2} \| K(X_1, D^t)w_1 + e_1 b_1 \|^2 + C_1 e_2^t \xi$$

$$\text{subject to} \quad (K(X_2, D^t)w_1 + e_2 b_1) + \xi \quad e_2, \quad \xi \quad 0 \tag{4}$$

$$\min \frac{1}{2} \| K(X_2, D^t)w_2 + e_2 b_2 \|^2 + C_2 e_1^t$$

matrix of appropriate dimension [8].

Further, a test data point $x \in R^n$ is assigned to a given class 'i' by using the following formula

$$class\ i = \min |K(x_t, D_t)w_i + b_i| \quad \text{for } i = 1,2. \tag{10}$$

C. Fuzzy Twin Support Vector Machine (FTSVM)

In FTSVM, weights are given to the different data samples on the basis of fuzzy membership values and the training gets biased towards the samples of interest. To calculate the fuzzy membership, we have considered the centroid measure for the data samples of each class where the membership values are assigned based on the distance of the data points from the centroid of that class [22]. The membership values are used as a basis for giving weights to the error tolerance parameter C for every data point.

The fuzzy membership function for centroid based membership is written as

$$mem = 1 \quad \frac{d_{cen}}{\max(d_{cen})+\varpi} \qquad (11)$$

where d_{cen} is the Euclidean distance of each data point from the centroid of its class i.e. $d_{cen} = \| x_+ \quad x_i \|$ if $y_i = 1$ and otherwise $d_{cen} = \| x \quad x_i \|$, where x_+ and x are the centroids of positive and negative class respectively, and ϖ is a small positive integer value.

In FTSVM, the non-linear hyperplanes are obtained through the following problems

$$\min \frac{1}{2} ||K(X_1, D^t)w_1 + e_1 b_1||^2 + C_1 S_1^t \xi$$

$$\text{subject to} \quad (K(X_2, D^t)w_1 + e_2 b_1) + \xi \quad e_2 , \quad \xi \quad 0 \qquad (12)$$

$$\min \frac{1}{2} ||K(X_2, D^t)w_2 + e_2 b_2||^2 + C_2 S_2^t \eta$$

$$\text{subject to} \quad (K(X_1, D^t)w_2 + e_1 b_2) + \eta \quad e_1 , \quad 0 \qquad (13)$$

where S_1, S_2 are vectors having the membership values of the data samples of the positive and the negative class respectively.

By introducing the Lagrangian functions of problems (12) & (13) and applying the Karush-Kuhn-Tucker (K.K.T) necessary and sufficient conditions [21], the Wolfe dual of (12) and (13) are written as

$$\min \frac{1}{2} \alpha_1^t H(G^t G)^{1} H^t \alpha_1 \quad e_2^t \alpha_1$$

$$\text{subject to } 0 \quad \alpha_1 \quad S_1 C_1 \qquad (14)$$

$$\min \frac{1}{2} \alpha_2^t G(H^t H)^{1} G^t \alpha_2 \quad e_1^t \alpha_2$$

$$\text{subject to } 0 \quad \alpha_2 \quad S_2 C_2 \qquad (15)$$

$$\min \frac{1}{2} w_1^t w_1 + \frac{\varpi_1}{l_2} e_2^t(\chi\pi(X_2)w_1 + e_2 b_1) + \frac{C_1}{l_1} e_1^t \xi_1$$

$$\text{subject to } \chi\pi(X_1)w_1 + b_1 e_1 + \xi_1 \quad 0 \qquad (17)$$

and $\xi_1 \quad 0$,

$$\min \frac{1}{2} w_2^t w_2 \quad \frac{\varpi_2}{l_1} e_1^t(\chi\pi(X_1)w_2 + e_1 b_2) + \frac{C_2}{l_2} e_2^t \xi_2$$

$$\text{subject to } \chi\pi(X_2)w_2 + b_2 e_2 \quad 0 , \quad \xi_2 \quad 0 . \qquad (18)$$

One can write (17) and (18) in the form of dual QPPs by considering the Lagrangian multipliers and apply the KKT necessary and sufficient conditions, which are

$$\min \frac{1}{2} \alpha_1^t K(X_1, X_1)^t \alpha_1 - \frac{\varpi_1}{l_2} e_2^t K(X_2, X_1)^t \alpha_1$$

$$\text{subject to } 0 \quad \alpha_1 \quad \frac{C_1}{l_1} e_1 , \; e_1^t \alpha_1 = \varpi_1 \qquad (19)$$

and

$$\min \frac{1}{2} \alpha_2^t K(X_2, X_2)^t \alpha_2 \quad \frac{\varpi_2}{l_1} e_1^t K(X_1, X_2)^t \alpha_2$$

$$\text{subject to } 0 \quad \alpha_2 \quad \frac{C_2}{l_2} e_2 , \; e_2^t \alpha_2 = \varpi_2 \qquad (20)$$

respectively. After computing the values of α_1, α_2 by solving the QPPs (19) & (20), one can find the solution of (w_1, b_1) and (w_2, b_2) in the following manner

$$w_1 = \chi\pi(X_1)^t \alpha_1 \quad \frac{\varpi_1}{l_2} \chi\pi(X_2)^t e_2 \text{ and } b_1 = \frac{1}{|N_1|} \Lambda \chi\pi(X_1) w_1 \; {}_{i\mathrm{E}|N1|}$$

$$w_2 = \chi\pi(X_2)^t \alpha_2 + \frac{\varpi_2}{l_1} \chi\pi(X_1)^t e_1 \text{ and } b_2 = \frac{1}{|N_2|} \Lambda \chi\pi(X_2) w_2 \; {}_{i\mathrm{E}|N2|}$$

$\alpha_i \; \mathrm{E} \; \frac{C}{\mathrm{I}} \; 0,$

Further, we compute the values of w_1, w_2, b_1 and b_2 as

$$\mathrm{T}w_1 = (G^tG+\delta I)^{-1}H^t\alpha \quad \text{and} \quad \mathrm{T}w_2 = (H^tH+\delta I)^{-1}G^t\alpha \qquad (16)$$

$$\lambda_1 \; \vartheta \qquad \lambda_2 \; \vartheta$$

The resultant classifier is obtained by using the equation (10).

D. Twin parametric-margin support vector machine (TPMSVM)

Recently, Peng [14] has proposed an efficient twin parametric-margin support vector machine which is an efficient learning approach of par-*v*-SVM. It finds two hyperplanes $f_1(x) = w_1^t\chi\pi(x)+b_1$ and $f_2(x) = w_2^t\chi\pi(x)+b_2$ in the feature space which are obtained by the following formulations:

where N_i is the index set of samples satisfying $_i \; \mathrm{I} \; l_i$ for $i=1,2.$ Finally, for any input sample $x \in R^n$, the classifier is given by

$$f(x) = sign\left(\sum_{i=1,2} \frac{w_i^t\chi\pi(x)}{\|w_i\|} + \frac{b_i}{\sum_{i=1,2}\|w_i\|} \right), \; i=1,2 \qquad (21)$$

III. Proposed Fuzzy based Lagrangian twin parametric-margin support vector machine (FLTPMSVM)

In this section, motivated by the work of Peng [14], we propose a new variant of TPMSVM which is based on TWSVM, called fuzzy based Lagrangian twin parametric-margin support vector machine where fuzzy membership values are calculated as similar in FTSVM. For the non-linear case, our proposed FLTPMSVM finds the positive and negative parametric margin hyperplanes $K(x^t,D^t)w_1+b_1=0$ and $K(x^t,D^t)w_2+b_2=0$.

In order to find the both parametric-margin hyperplanes, we have considered the following formulation

$$\min \frac{1}{2}(w_1^t w_1+b_1^2)+\varpi e_2^t(K(X_2,D^t)w_1+e_2 b_1)+\frac{S_1C_1}{2}\xi^t\xi$$

subject to $(K(X_1,D^t)w_1+e_1b_1)+\xi \quad 0$, (22)

and

$$\min \frac{1}{2}(w_2^t w_2+b_2^2) \quad \varpi e_1^t(K(X_1,D^t)w_2+e_1 b_2)+\frac{S_2C_2}{2}\eta^t\eta$$

subject to $(K(X_2,D^t)w_2+e_2b_2) \quad 0$. (23)

The Lagrangian functions of (22) and (23) are given as

$$L_1=\frac{1}{2}u_1^tu_1+\varpi e_2^t Hu_1+\frac{C_1S_1}{2}\xi^t\xi \quad \alpha_1^t(Gu_1+\xi) \qquad (24)$$

and

$$L_2=\frac{1}{2}u_2^tu_2 \quad \varpi e_1^t Gu_2+\frac{C_2S_2}{2}\eta^t\eta \quad \alpha_2^t(Hu_2+\eta) \qquad (25)$$

where $u_1=\begin{bmatrix}w_1\\ b_1\end{bmatrix}$, $u_2=\begin{bmatrix}w_2\\ b_2\end{bmatrix}$.

Now, one can apply the Karush-Kuhn-Tucker (K.K.T) necessary and sufficient conditions to find the Wolfe dual of equations (22) and (23) as

$$\min_{\alpha_1\geq 0} \frac{1}{2}\alpha_1^t\left(\frac{I}{S_1C_1}+GG^t\right)\alpha_1 \quad \varpi e_2^t HG^t\alpha_1 \qquad (26)$$

and

$Q_2=\frac{I}{S_2C_2}+HH^t$, $r_1=\varpi GH^te_2$ and $r_2=\varpi HG^te_1$ which will lead to following pair of classical complementary problems [23]:

$$0 \quad \alpha_k \perp (Q_k\alpha_k \quad r_k) \quad 0 \qquad (29)$$

To find the solution of problems (29), we consider the equivalent pair of problems: for any $\lambda_k>0$, for $k=1,2$, the relations

$$(Q_k\alpha_k \quad r_k)=(Q_k\alpha_k \quad \lambda_k\alpha_k \quad r_k)_+ \qquad (30)$$

So, we can solve the problem (30) by writing as the following iterative scheme: for *i=0,1,2…*

$$\alpha_k = Q_k^{-1}(r_k+(Q_k\alpha_k \quad \lambda_k\alpha_k \quad r_k)_+)$$

i.e.

$$\alpha_1^{i+1}=\left(\frac{I}{S_1C_1}+GG^t\right)^{-1}\left(r_1+\left(\left(\frac{I}{S_1C_1}+GG^t\right)\alpha_1^i \quad \lambda_1\alpha_1^i \quad r_1\right)_+\right) \qquad (31)$$

and

$$\alpha_2^{i+1}=\left(\frac{I}{S_2C_2}+HH^t\right)^{-1}\left(r_2+\left(\left(\frac{I}{S_2C_2}+HH^t\right)\alpha_2^i \quad \lambda_2\alpha_2^i \quad r_2\right)_+\right) \qquad (32)$$

when $0<\lambda_1<\frac{2}{C_1}$ and $0<\lambda_2<\frac{2}{C_2}$, the above iterative schemes will converge to the solutions α_1 and α_2 respectively [24]. The final classifier is defined by the equation (21).

min $\frac{1}{2}\alpha^t$ I

1

1

$+\varpi_t e^t$ $GH^t\alpha$ (27)

I

$+H_t$ I α ——

$\alpha_2 \; 0 \; \overline{2} \; \frac{^2\mathrm{I}}{C} S$ I 2 2 1 2

2 2

After computing the values of α_1 and α_2 from (26) and (27), we find the positive and negative class hyperplanes $[K(X_1, D^t) \quad e_1]u_1 = 0$ and $[K(X^2, D^t) \quad e_2]u_2 = 0$ respectively where $u_1 = G^t\alpha_1 \quad \varpi_1 H^t e_2$ and $u_2 = H^t\alpha_2^t + \varpi_2 G^t e_1$.

To predict the class of new data sample $x \in R^n$ in case of FLTPMSVM, we find the class label by using equation (21).

These dual QPPs (26) and (27) are of the form

$$\min_{\alpha_k \geq 0} \frac{1}{2} \alpha_k^t Q_k \alpha_k + r_k^t \alpha_k \qquad (28)$$

IV. EXPERIMENTAL RESULTS

To validate the effectiveness of our method, we have conducted an experiment and have compared our proposed FLTPMSVM with SVM, TWSVM, FTSVM and TPMSVM on well-known real-world datasets as well as one artificial dataset i.e. Ripley's dataset [25]. All the experiments are conducted on a PC with 64 bit, 3.40 GHz Intel© Core™ i7-3770 CPU and 4 GB RAM, running Windows 7 operating system. The software package used is MATLAB R2008a along with MOSEK optimization toolbox for TWSVM, FTSVM and TPMSVM, available at https://www.mosek.com. The datasets are normalized to the range [0,1] before experiment is performed on them. In this experiment, we implemented all the methods for non-linear case using Gaussian kernel which is given by

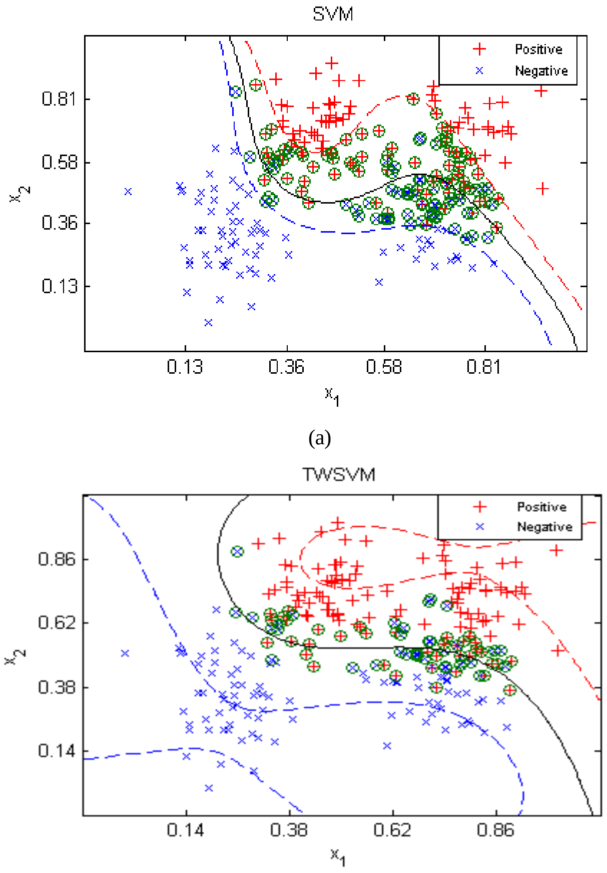
$$K(x_i, x_j) = \exp\left(\frac{\| x_i \quad x_j \|^2}{2\alpha^2} \right).$$

The optimum value of kernel parameter α is for $k=1,2$ respectively, where $Q_1 = \left[\frac{I}{S_1 C_1} + GG^t \right]$,

obtained from the set $\{2^{5},...,2^{5}\}$, C_i $(i=1,2)$ are also obtained from the set $\{10^{5},...,10^{5}\}$ and for TPMSVM and FLTPMSVM, the optimum values of $C_i / \upsilon_i \ (i=1,2)$ are selected from $\{0.1,\ldots,0.9\}$ by using 10-fold cross-validation of the training data.

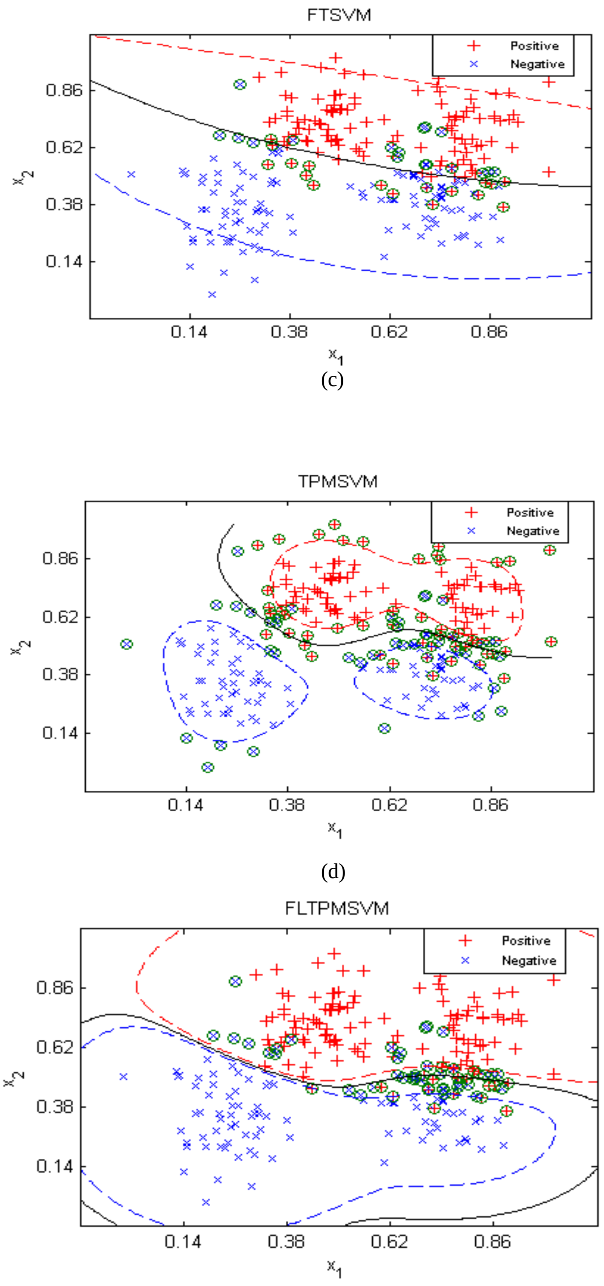
We set average accuracy and average training time as the performance evaluation criteria for all the algorithms. Statistical result analysis is performed on testing data to calculate the average accuracy, standard deviation of result and average training time. We have considered the artificially-generated Ripley's synthetic dataset. The Ripley's dataset is a synthetic dataset in R^2 that contains 250 training samples and 1000 samples for testing. In the figure, the positive class samples and the negative class samples are depicted using ‘ξ’ and ‘+’ symbols respectively. Support vectors are marked using circles around them. The learning results of our experiment on artificial Ripley's dataset for SVM, TWSVM, FTSVM, TPMSVM and FLTPMSVM are shown in Figures 1(a-e).

One can observe that FLTPMSVM obtains better decision classifiers in comparison to SVM, TWSVM, FTSVM, and TPMSVM. In Table 1, we show the predicted accuracies, optimum parameters with learning time for FLTPMSVM with others methods. One can notice from Table 1 that our proposed FLTPMSVM has best classification result among these algorithms as well as the learning time of our proposed method is less which show that FLTPMSVM takes less computation time when compared with the other considered methods.



(a)

(b)



(c)

(d)

(e)

Fig. 1. Discriminant boundaries of FLTPMSVM with TWSVM, FTSVM and TPMSVM on Ripley's dataset using Gaussian kernel

Further, we have considered 10 UCI benchmark well-known real-world datasets i.e. Australian-Credit, Breast-Cancer, BUPA liver, Cleveland, Haberman, Heart-Statlog, Ionosphere, Pima-indians-diabetes, Transfusion, Wpbc from UCI repository [26]. The classification accuracy along with optimum parameters and training time of all algorithms are presented in Table 2. Our proposed FLTPMSVM is performed better in 6 out of 10 datasets. The solution of the proposed

TABLE I. The result of FLTPMSVM, TWSVM, FTSVM and TPMSVM on Ripley's dataset

Dataset (Train size, Test size)	SVM (C,μ) Time	TWSVM $(C_1 = C_2, \mu)$ Time	FTSVM $(C_1 = C_2, \mu)$ Time	TPMSVM $(C = C_1 = C_2, \mu, \varpi /$ $C)$	FLTPMSVM $(C = C_1 = C_2, \mu, \varpi / C)$ Time
Ripley (250x2, 1000x2)	90.3 (10^-5, 2^2) 1.5464	88.9 (10^0, 2^-1) 0.124	88.7 (10^1, 2^1) 0.1268	Time 90 (10^-3, 2^-3,0.4) 0.1212	**90.9** (10^0, 2^-3, 0.9) 0.1137

TABLE II. The results of FLTPMSVM with TWSVM, FTSVM and TPMSVM using Gaussian kernel on real-world datasets

Dataset (Train size, Test size)	SVM (C,μ) Time	TWSVM $(C_1 = C_2, \mu)$ Time	FTSVM $(C_1 = C_2, \mu)$ Time	TPMSVM $(C = C_1 = C_2, \mu, \varpi /$ $C)$	FLTPMSVM $(C = C_1 = C_2, \mu, \varpi / C)$ Time
				Time	
Australian-Credit (413x14, 277x14)	79.4974 ± 8.1208 (10^-5, 2^-3) 1.3881	84.4444 ± 8.5695 (10^-2, 2^1) 0.1193	**85.2116** ± 7.811 (10^-5, 2^-1) 0.1435	84.828 ± 9.5534 (10^-2, 2^3, 0.1) 0.1205	84.8413 ± 5.8409 (10^-3, 2^3, 0.1) 0.1176
Breast-Cancer (149x9, 550x9)	96.3636 ± 2.2677 (10^-5, 2^-1) 5.4858	96.3636 ± 3.3195 (10^-5, 2^-1) 0.4643	96.7273 ± 3.1840 (10^-5, 2^-1) 0.4654	97.0909 ± 2.5997 (10^-2, 2^0, 0.2) 0.4225	**97.2727** ± 2.6068 (10^-2, 2^3, 0.1) 0.4131
BUPA liver (241x6, 104x6)	49.7273 ± 18.388 (10^-5, 2^3) 0.2019	56.1818 ± 18.615 (10^-2, 2^-1) 0.0248	58.3636 ± 20.7384 (10^0, 2^-1) 0.0257	59.1818 ± 18.1285 (10^0, 2^0, 0.4) 0.0222	**60.2727** ± 27.2897 (10^-4, 2^4, 0.1) 0.0176
Cleveland (178x13, 119x13)	78.1061 ± 9.08 (10^-5, 2^-2) 0.2595	83.1818 ± 5.5762 (10^0, 2^3) 0.0314	**84.0909** ± 6.0501 (10^-5, 2^2) 0.0269	79.0152 ± 4.2564 (10^-1, 2^5, 0.2) 0.0259	79.9242 ± 7.9315 (10^0, 2^0, 0.2) 0.0230
Haberman (183x3, 123x3)	68.3974 ± 12.4044 (10^-5, 2^3) 0.2761	75.7051 ± 12.6211 (10^-1, 2^0) 0.0262	74.8077 ± 11.3109 (10^-5, 2^0) 0.0328	69.1667 ± 12.8754 (10^-3, 2^-5, 0.5) 0.0256	**75.8333** ± 11.8779 (10^-2, 2^-2, 0.2) 0.0241
Heart-Statlog (161x13, 109x13)	79.8182 ± 10.3012 (10^-5, 2^0) 0.2209	79.7273 ± 12.182 (10^-2, 2^0) 0.0224	79.7273 ± 12.9139 (10^1, 2^2) 0.0297 **93.2727** ± 4.6592	81.5455 ± 7.8326 (10^-3, 2^0, 0.5) 0.0226	**84.4545** ± 7.4035 (10^-2, 2^1, 0.2) 0.0198
Ionosphere (246x34, 105x34)	83.6364 ± 11.5311 (10^-5, 2^-1) 0.2089	**93.2727** ± 4.6592 (10^-2, 2^0) 0.0287	(10^-2, 2^0) 0.028 77.6735 ± 6.7968	92.5455 ± 7.2322 (10^-2, 2^0, 0.2) 0.0231	**93.2727** ± 7.8928 (10^-4, 2^0, 0.1) 0.0182
Pima-indians- diabetes (307x8, 461x8)	77.6781 ± 8.1724 (10^-5, 2^5) 3.8896	**77.8908** ± 7.5103 (10^0, 2^1) 0.3198	(10^0, 2^2) 0.3304 77.6667 ± 20.3093	75.7216 ± 8.0453 (10^0, 2^2, 0.2) 0.2965	72.0352 ± 9.2246 (10^-2, 2^0, 0.1) 0.2780
Transfusion (448x4, 300x4)	**80.3333** ± 19.0807 (10^-5, 2^4) 1.6251	77.6667 ± 19.5031 (10^-1, 2^1) 0.135	(10^-5, 2^0) 0.1357 79.4643 ± 16.544	66 ± 28.8376 (10^-2, 2^-1, 0.1) 0.123	80 ± 18.8562 (10^-5, 2^1, 0.5) 0.1202
Wpbc (116x33, 78x33)	76.9643 ± 22.3583 (10^-5, 2^-2) 0.117	79.4643 ± 16.544 (10^-2, 2^-1) 0.015	(10^-2, 2^-1) 0.0199	78.2143 ± 18.3426 (10^-1, 2^-1, 0.1) 0.0155	**81.9643** ± 17.6466 (10^-4, 2^-1, 0.1) 0.0135

TABLE III. Average ranks of TWSVM, FTSVM, TPMSVM and FLTPMSVM using Gaussian kernel on real-world datasets

Dataset (Train size, Test size)	SVM	TWSVM	FTSVM	TPMSVM	FLTPMSVM
Australian-Credit	5	4	1	3	2
Breast-Cancer BUPA	4.5	4.5	3	2	1
liver Cleveland	5	4	3	2	1
Haberman	5	2	1	4	3
Heart-Statlog	5	2	3	4	1
Ionosphere	3	4.5	4.5	2	1
Pima-indians-diabetes	5	2	2	4	2
Transfusion	2	1	3	4	5
Wpbc	1	3.5	3.5	5	2
Average Rank	5	2.5	2.5	4	1
	4.05	3	2.65	3.4	1.9

FLTPMSVM is obtained by using simple linearly convergent iterative approach instead of solving two QPPs as in the case of TWSVM, FTSVM and TPMSVM. Hence, FLTPMSVM achieves a comparatively lower training cost as compared to the others. Further, the average ranks of all the methods are shown in Table 3, where rank is calculated on the basis of accuracies. One can observe from this table that proposed FLTPMSVM has the lowest average rank among all methods.

V. CONCLUSION

In this paper, a fuzzy-based Lagrangian twin parametric-margin support vector machine (FLTPMSVM) is proposed to lessen the effect of the outliers, by applying the concept of fuzzy support vector machine (FSVM) [15] on twin parametric-margin support vector machine (TPMSVM) [14]. Furthermore, the solution of FLTPMSVM is obtained by solving simple linearly convergent iterative schemes instead of solving a pair of QPPs as in case of TWSVM, FTSVM and TPMSVM. Experiments are carried out for non-linear case on publicly available real-world datasets as well as on one artificial dataset. Result shows that FLTPMSVM delivers comparative or better classification accuracy with the other considered methods and also suitable for heteroscedastic error structure. Moreover, our proposed method could achieve a faster training time as compared to all the other reported algorithms for all the datasets considered. Similar to TPMSVM, our proposed method loses the sparseness that can be one of the future works.

REFERENCES

[1]. V.N. Vapnik, ‘Statistical Learning Theory,” John Wiley & Sons, New York, 1998.

[2]. T. Joachims, C. Ndellec & C. Rouveriol, “Text categorization with support vector machines: learning with many relevant features,” in European Conference on Machine Learning No.10, Chemnitz, Germany, pp.137-142, 1998.

[3]. C. Cortes & V. Vapnik, “Support-Vector Networks, Machine Learning,” 20(6), pp.273-297, 1995.

[4]. R. Khemchandani & S. Sharma, “Robust Least Squares Twin Support Vector Machine for Human Activity Recognition,” Applied Soft Computing, 47, pp.33–46, 2016.

[5]. Y-K. Bao, Z-T. Liu, L. Guo & W. Wang, “Forecasting stock composite index by fuzzy support vector machines regression,” in Proceeding of International Conference on Machine Learning and Cybernetics, 6, pp.3535 – 3540, 2005.

[6]. T. Ebrahimi, G. N. Garcia & J.M. Vesin, “Joint time-frequency-space classification of EEG in a brain-computer interface application,” Journal of Applied Signal Processing, vol. 1(10), pp.713-729, 2003.

[7]. R. Malhotra, & D.K. Malhotra, “Evaluating consumer loans using neural networks,” Omega 31, pp.83-96, 2003.

[8]. Jayadeva, R. Khemchandani, & S. Chandra, “Twin support vector machines for pattern classification,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29(8), pp.905-910, 2007.

[9]. J.A.K. Suykens & J. Vandewalle, “Least squares support vector machine classifiers,” Neural Processing Letters, 9(6), pp.293-300, 1999.

[10]. Mangasarian, Olvi L., and David R. Musicant. "Lagrangian support vector machines." *Journal of Machine Learning Research* 1.Mar (2001): 161-177.

[11]. Balasundaram, S., Deepak Gupta, and Subhash Chandra Prasad. "A new approach for training Lagrangian twin support vector machine via unconstrained convex minimization." *Applied Intelligence* 46.1 (2017): 124-134.

[12]. P-Y. Hao, “New support vector algorithms with parametric insensitive/margin model,” Neural Networks, 23 (4), pp.60–73, 2010.

[13]. B. Scholkopf, A.J. Smola, R.C. Williamson & P.L. Bartlett, “New Support Vector Algorithms,” Neural Computation, 12 (8), pp.1207– 1245, 2000.

[14]. Peng, X., (2011), “TPMSVM: A novel twin parametric-margin support vector machine for pattern recognition,” Pattern Recognition, 44, pp.2678-2692.

[15]. C.-F. Lin &, S.-D. Wang,, “Fuzzy Support Vector Machines,” IEEE Trans. Neural Networks, vol. 13(5), pp. 464-471 (2002).

[16]. R. Batuwita & V. Palade, “FSVM-CIL: Fuzzy Support Vector Machines for Class Imbalance Learning,” IEEE Trans. Fuzzy Systems, vol. 18, no. 3, pp. 558-571, June (2010).

[17]. D. Tsujinishi & S. Abe, “Fuzzy least squares support vector machines,” Proceedings of the International Joint Conference on Neural Networks, Portland, Oregon, pp. 1599–1604, (2003).

[18]. Y. Wang, S. Wang & K. K. Lai, “A new fuzzy support vector machine to evaluate credit risk.” IEEE Transactions on Fuzzy Systems, Vol. 13(9), pp. 820-831 (2005).

[19]. Chaudhuri & K. De, “Fuzzy support vector machine for bankruptcy prediction,” Applied Soft Computing, 11(5), pp. 2472-2486 (2010).

[20]. S. Zhang, S. Zhao, Y. Sui & L. Zhang, “Single object tracking with fuzzy least squares support vector machine.” IEEE Transactions Image Processing, Vol. 24, pp. 5723–5738 (2015)

[21]. N. Cristianini & J. Shawe-Taylor, “An introduction to support vector machines and other kernel based learning methods,” Cambridge University Press, Cambridge, 2000.

[22]. R. Batuwita & V. Palade, "FSVM-CIL: Fuzzy Support Vector Machines for Class Imbalance Learning", IEEE Trans. Fuzzy Systems, vol. 18, no. 3, pp. 558-571, June (2010).

[23]. O.L. Mangasarian, “Nonlinear Programming,” SIAM Philadelphia, PA, 1994.

[24]. O.L. Mangasarian, & D.R. Musicant, “Lagrangian support vector machines,’ Journal of Machine Learning Research, 1, pp.161-177, 2001.

[25]. B.D. Ripley, “pattern recognition and Neural Networks,” Cambridge university Press, Cambridge, 1996.

[26]. P.M. Murphy & D.W. Aha, “UCI repository of machine learning databases,” University of California, Irvine. 1992, *http://www.ics.uci.edu/~mlearn*.