Robust Fuzzy Support Vector Machines With Locally Linear Embedding Algorithm

Shiva Sanati
Department of Computer Engineering, Medical Imaging Lab
Ferdowsi University of Mashhad
Mashhad, Iran
shiva.sanati@stu.um.ac.ir

Saeid Abbaasi
Department of Computer Engineering, Machine Learning Lab
Ferdowsi University of Mashhad
Mashhad, Iran
s.abbaasi@stu.um.ac.ir

Marjan Mazruei
Department of Computer Engineering, Medical Imaging Lab
Ferdowsi University of Mashhad
Mashhad, Iran
marjan.mazruei@stu.um.ac.ir

Reza Monsefi
Department of Computer Engineering, Machine Learning Lab
Ferdowsi University of Mashhad
Mashhad, Iran
monsefi@um.ac.ir

***Abstract* — Support vector machines is one of the most popular binary classification methods in machine learning. One of the main problems of SVMs is its sensitivity to noise and outliers. Various methods have been proposed to overcome this problem. Fuzzy SVMs (FSVMs) main purpose is to consider the constant C in SVM (which is a penalty term to determine the trade-off between margin maximization and training error minimization) as a vector that will permit to some samples to have different amount of logical movement, so that it has been placed at an appropriate location in the feature space. Our proposed method is to use locally linear embedding (LLE) neighborhood concept to obtain a good penalty vector which allows noisy data to be placed at an appropriate location. Experiments indicate the superiority of the robustness of our proposed method dealing with noise impregnated data comparing with traditional SVM.**

Keywords- Support vector machine, Fuzzy Support Vector Machine, Locally Linear Embedding, Classification, noisy data, Robust SVM

I. INTRODUCTION

Support Vector Machines (SVMs) is a powerful statistical classification technique based on the idea of Structural Risk Minimization. Fuzzy sets are capable of handling uncertainty and impreciseness in corrupted data. Thus, by incorporation of Machine Learning and Fuzzy prediction, accuracy of the whole model is enhanced. FSVM is implemented for analyzing predictors as financial ratios [1]. SVM have been developed by Vapnik [2], [3], [4]. Kernel SVMs moves the input vectors into a higher dimensional feature space and an optimal separating hyperplane in this space is constructed. SVMs have been successfully applied to a number of applications ranging from bioinformatics to text categorization and face recognition or fingerprint identification. The good generalization ability of SVMs is achieved by finding a large margin between two classes [5]. In many applications, the theory of SVMs has been shown to provide higher performance than traditional learning machines [6] and has been introduced as powerful tools for solving classification problems.

The main difference between SVMs and FSVMs is that the cost C of FSVMs is multiplied by fuzzy membership s_i. Concept behind FSVMs model is to set a fuzzy membership to each input point and to reformulate SVMs that different input points can make different contributions to the learning of the decision surface. FSVMs is also based on the maximization of the margin similar to the classical SVMs. However, it uses fuzzy membership function instead of fixed weights to prevent noisy data points from making narrower margins [7].

Choosing an appropriate value for parameter plays an important role on the performance of FSVMs which is the main idea of this paper. Here we give more detail about SVMs and FSVMs.

A. SVMs formulation

Given a training sample $\{x_i, y_i\}$ *for* $\mathrm{i}=1,\ldots,n$, where $x_i \in \mathbb{R}^d$ is the ith input pattern, d denotes the dimension of the input space and y_i is its corresponding observed result, which is a binary variable, 1 or -1. We assume that the training set is linearly separable after being mapped into a higher dimensional feature space by a nonlinear function $\varphi(\bullet)$, the classifier should be constructed as follows:

$$\begin{cases} w^T\varphi(x_i)+b \geq +1 & if\ \ y_i = 1 \\ w^T\varphi(x_i)+b \leq -1 & if\ \ y_i = -1. \end{cases} \tag{1}$$

The distance between the two boundary lines is $2/\|w\|_2$. Large distance is encouraged for the purpose of generalization ability. In the real world, the training set is usually not linearly separable even when mapped into a high dimensional feature space, which means we cannot find a perfect separating hyperplane that makes each x_k satisfy condition (1). A soft margin is introduced to incorporate the possibility of violation. The error term ξ_i of instance k is defined as follows:

$$\begin{cases} y_i\left[w^T\varphi(x_i)+b\right] \geq 1-\xi_i \\ \xi_i \geq 0 \end{cases} \text{, for } \mathrm{i}=1,\ldots,n. \tag{2}$$

It is expected that the training should maximize the classification margin and minimize the sum of error terms at the same time. When the training set is not nonlinear

978-1-5090-0139-2/15/$31.00 ©2015 IEEE

separable in the feature space, the two goals usually cannot be achieved at the same time. We can formulate the two group classification problems as the following optimization problem:

$$\begin{aligned} &\min \zeta(w,b,\xi) = \frac{1}{2}w^T w + C\sum_{i=1}^{n}\xi_i \\ &\text{Subject to}: y_i\left[w^T\varphi(x_i)+b\right] \geq 1-\xi_i, \text{ for i} = 1,\ldots,n \\ &\xi_i \geq 0, \text{ for i} = 1,\ldots,n. \end{aligned} \tag{3}$$

The larger C, the more the error term is emphasized. Small C means that the large classification margin is encouraged. By introducing Lagrange multiples α_i and β_i for the constraints in the (3), the problem can be transformed into its dual form then by derivation we have:

$$w = \sum_{\mathrm{k}=1}^{\mathrm{N}} \alpha_{\mathrm{k}} y_k \phi(x_k) \tag{4}$$

Where α_k is the solution of quadratic programming dual problem of (3). The merit of support vector machine is that, by a kernel function, $K(x_i, x_j)$, which is the inner product in the feature space, it tries to make the training data linearly separable in the high dimension feature space, thus achieve nonlinear-separability in the input space. By this way, we can find the optimal map function, even without specifying the explicit form of map function $\phi(x)$. Substituting $\phi(x_i)^T\phi(x_j)$ in dual form with kernel function $K(x_i, x_j)$ leads to the following optimization problem:

$$\begin{aligned} &\max_{\alpha} J(\alpha) = \sum_{k=1}^{N}\alpha_k - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i, x_j) \\ &\text{Subject to}: \sum_{\mathrm{k}=1}^{\mathrm{N}}\alpha_{\mathrm{k}} y_k = 0 \\ &0 \leq \alpha_k \leq C \text{ for } k = 1,\ldots,N. \end{aligned} \tag{5}$$

After solving (5) and substituting $w = \sum_{k=1}^{N}\alpha_k y_k \varphi(x_k)$ into the original classification problem, we obtain the following classifiers:

$$y(x) = sign\left(w^T\phi(x)+b\right) = sign\left(\sum_{k=1}^{N}\alpha_k y_k K(x, x_k)+b\right). \tag{6}$$

B. FSVM formulation

In order to decrease the effect of outliers or noises, we can assign each data point in the training dataset with a membership and sum the deviations weighted by their memberships. If one data point is detected as an outlier, it is assigned with a low penalty, so its contribution to total error term decreases. Unlike the equal treatment in standard SVM, this kind of SVM fuzzifies the penalty term in order to reduce the sensitivity of less important data points. In [11] and [15], the classification problem is modeled by the following programming:

$$\begin{aligned} &\min \zeta(w,b,\xi_i) = \frac{1}{2}w^T w + \sum_{i=1}^{n} s_i\xi_i \\ &\text{Subject to}: y_i\left[w^T\varphi(x_i)+b\right] \geq 1-\xi_i, \text{for i} = 1,\ldots,n \\ &\xi_i \geq 0, \text{ for i} = 1,\ldots,n. \end{aligned} \tag{7}$$

Where s_i is a separate multiplier for each sample.

II. PRIOR WORK

As we mentioned earlier, since the optimal hyperplane obtained by the SVM depends on only a small part of the data points, it may become sensitive to noises or outliers in the training set [8]. To solve this problem, one approach is to do some preprocessing on the training data to remove noises or outliers, and then use the remaining set to learn the decision function [9].

This method is hard to implement if we do not have enough knowledge about noises or outliers. In many real world applications, we are given a set of training data without knowledge about noises or outliers. There are some risks to remove the meaningful data points as noises or outliers. There are many discussions in this topic and some of them show good performance. The theory of leave-one-out SVMs [10] (LOOSVMs) is a modified version of SVMs. This approach differs from classical SVMs in that it is based on the maximization of the margin, but minimizes the expression given by the bound in an attempt to minimize the leave-one-out error. No free parameter makes this algorithm easy to use, but it lacks the flexibility of tuning the relative degree of outliers as meaningful data points. Its generalization, the theory of adaptive margin SVMs (AM-SVMs) [11], uses a parameter k to adjust the margin for a given learning problem. It improves the flexibility of LOOSVMs and shows better performance. Experiments in both of them show the robustness against some outliers. FSVMs solve this kind of problems by introducing the fuzzy memberships of data points. The main advantage of FSVMs is that we can associate a fuzzy membership to each data point such that different data points can have different effects in the learning of the separating hyperplane. We can treat the noises or outliers as less importance and let these points have lower fuzzy membership and then they can move more to locate an appropriate place that can reduces the error. It is also based on the maximization of the margin like the classical SVMs, but uses fuzzy memberships to prevent noisy data points from making narrower margin. This equips FSVMs with the ability to train data with noises or outliers by setting lower fuzzy memberships to the data points that are considered as noises or outliers with higher probability. The previous work of FSVMs [12] did not address the issue of automatic setting of the fuzzy membership from the data points. We need to assume a noise model of the training data points, and then try and tune the fuzzy membership of each data point in the training. Without any knowledge of the distribution of data points, it is hard to associate the fuzzy membership to the data points.

In [13] and [14], fuzzy support vector machine is proposed to deal with the problem. Each instance will be assigned a membership that is according to its distance from its own class. In FSVMs each instance's contribution to the total error term in the objective function is weighted by its membership, instead of equal weight of 1. Experimental results show that the proposed FSVMs can actually reduce the effect of outliers and yield higher classification rate than traditional SVMs can do.

In [15] provides a new FSVM, to evaluate the credit risk of consumer lending. They treat each instance as both of positive and negative classes, but assigned with different memberships. This can also be regarded as constructing two instances from the original instance and assigning its memberships of positive and negative classes respectively. If one instance is detected as an outlier, which means that it is very likely to fall in this class, but in fact it has fallen in the other class, we treat it as a member of this class with large membership, at the same time treat it as a member of the contrary class with small membership. In [12], [16], and [17] provide a probability framework for understanding SVMs or least square SVMs. Their work is mainly focus on automatic adjustment of regularization parameter and the kernel parameter to the near optimal.

In [17] Tsujinishi and Abe defined a membership function in the direction perpendicular to the optimal separating hyperplane that separates a pair of classes, and then determine the memberships of any data for each class based on minimum or average operation. In their each two-class classification computation, the objective function is the same as the basic least square SVMs (LS-SVMs). In other words [18] uses memberships generated the results of pairwise classification by standard LS-SVMs to determine one instance's memberships of a class by minimum or average operator. In [19] presents a bilateral weighted fuzzy SVM in which every applicant in the training dataset is regarded as both good and bad class with different memberships that is generated by some basic credit scoring methods. By doing so, the authors expect the new fuzzy SVMs to achieve better generalization ability while keeping the merit of insensitive to outliers. They also reformulate the training problem into a quadratic programming problem.

Tao and Wang [20] modified proposed FSVMs in [12] and obtained the optimization for linearly fuzzy inseparable classification problems which were called the new fuzzy support vector machines (NFSVMs). For avoiding noises, [15] applied fuzzy SVMs to train the data into positive and negative classes, and proposed an outliers detection method, which proved that FSVM actually relieved an outliers problem and yielded a higher classification rate than SVMs did. In [21] and [22] used FSVMs Gaussian kernel and applied different input data to make different contributions to the domain descriptions. Accordingly, the utility of FSVMs indeed improves the efficiency of classification in practice. In [23], they investigate the effects of the membership values in FSVMs and compare them to those of SVMs.

In [24] Zhao et al. proposed a new robust method to deal with unbalanced datasets. It uses FSVM and first get the support vectors and mark them. Second, he used the marked support vectors and the kernel modification method based on Riemannian metric in order to modify the kernel function. Finally, he used the new kernel to train the datasets again until the convergence of the classifier.

In [25], Fang et al. describes the research status and development of fuzzy theory and fuzzy support vector machine. They introduced several design methods of fuzzy membership function, and further probed into several fuzzy support vector machine principle and its properties. They finally summarized the fuzzy support vector machines in various application fields and its development prospects.

Wei-Yuan Cheng proposed a FM^3 algorithm for designing a Takagi-Sugeno-type (T–S-type) fuzzy model for classification problems. The rule number in the FM^3 is determined by clusters instead of support vectors. So it causes the model to be simpler. Parameter tuning using the incremental SVM (ISVM) minimizes a soft margin instead of training errors that it helps increasing the generalization ability of the FM^3 [26].

In [27] a new classification algorithm-FSVM with minimum within-class scatter (WCS-FSVM) proposed, which incorporates minimum within-class scatter in Fisher Discriminant Analysis (FDA) into FSVM. The main idea is that an optimal hyperplane is found such that the margin is maximized while the within-class scatter is kept as small as possible. In addition, they proposed a new fuzzy membership function. This function considers both the distance between each sample point and its class center and the affinity among sample points.

III. THE PROPOSED METHOD

Before start explaining the proposed method, we briefly introduce locally linear embedding (LLE) method.

A. LLE

Locally linear embedding (LLE) is an unsupervised learning algorithm that computes low-dimensional, neighborhood-preserving embedding of high-dimensional inputs. Unlike clustering methods for local dimensionality reduction, LLE maps its inputs into a single global coordinate system of lower dimensionality, and its optimizations do not involve local minima (as illustrated in Fig. 1). LLE is able to learn the global structure of nonlinear manifolds. LLE, eliminates the need to estimate pairwise distances between widely separated data points. Unlike previous methods, LLE recovers global nonlinear structure from locally linear fits [28].

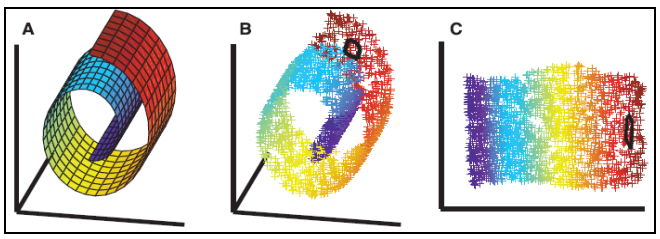

Figure 1. Nonlinear dimensionality reduction: The color coding illustrates the neighborhood preserving mapping discovered by LLE. black outlines in (B) and (C) show the neighborhood of a single point [27].

LLE algorithm can be summarized in 3 steps, which maps high dimensional inputs X_i to low dimensional outputs Y_i via local linear reconstruction weights W_{ij}:

- Compute the neighbors of each data point, X_i.
- Compute the weights W_{ij} that lead to a good reconstruction for each data point X_i by its

neighbors, minimizing the cost in Equation (8) by constrained linear fits.

$$E(W) = \sum_i \left| \vec{x}_i - \sum_j W_{ij} \vec{X}_j \right|^2, \tag{8}$$

- Compute the vectors Y_i having the best reconstruction by the weights W_{ij}, minimizing the quadratic form in Equation (9) by its bottom nonzero eigenvectors.

$$\phi(Y) = \sum_i \left| \vec{Y}_i - \sum_j W_{ij} \vec{Y}_j \right|^2. \tag{9}$$

B. *Robust fuzzy support vector machines with LLE*

As we mentioned earlier, LLE uses the local neighbors to reduce the dimension of data. The geometry of data will have no change in reduced space. It's clearly that if a sample in not neighbor of other samples in its own class, the probability of that data being noise is increased and vice versa. On the other hand, an outlier or noisy data, usually has a few of instances in its close vicinity. If we can detect this samples and assign a low penalty term it will lead them having more flexibility to take a place in the feature space hence improving classification task.

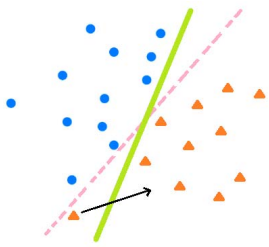

Figure 2. The red dashed classification line of SVM has less generality. The green bold classification line seems to have better accuracy againste unseen data. The marked triangle is supposed a noisy sample.

As we illustrated in figure above, suppose that the red marked triangle, is a noisy sample and the result of SVMs will be the improper dashed line which has low generality. Whereas if this point had a low penalty for ξ_i, it could move logically more to the proper position (illustrated with arrow) and then the classification line (the green bold one) will have higher generality facing with unseen data.

By means of LLE neighborhood concept, we use the number of times that a sample has been the neighbor of other samples, kept in a vector S. Then the normalized S is used as a penalty term for ξ_i based on FSVMs formulation. It's obvious that by this method, each samples have a different penalty value, unlike in SVMs that the constant C is the same for all samples. To clarify the concept, suppose that we have a 2D, two class dataset as illustrated in figure below.

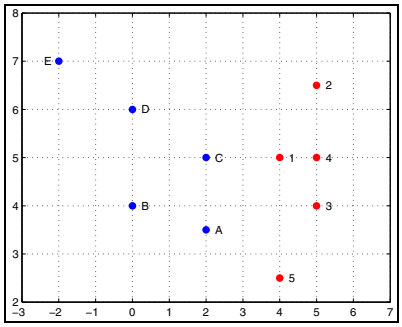

Figure 3. Example of a 2D two class data.

In figure 3 we have considered samples '5' and 'E' as noisy data. If we use the LLE feature reduction with KNN method (K=3) to find the neighbors, the neighborhood matrix for each class will be defined as in table 1 and 2. The last row of each tables sums the times which a sample was neighbor of the other ones. For '5' and 'E' this numbers is small and it show that the probability of these two samples to be noise is increased. So we could allow this noisy data to have more freedom to move (by assigning smaller s_i that will causes larger ξ_i) in other to take a proper position for a better classification. We can use normalize version of this number (last row of table 1 and 2) that have a specific average (for example, 1 in below tables) as an input penalty vector (S) for FSVMs. For example ticks in row one shows that the neighbors of sample 1 are 2, 3 and 4. Samples '5' and 'E' have less weight s_i so they can take bigger slack ξ_i.

TABLE 1. NEIGHBORS OF RED TRIANGLES IN FIGURE 3

	1	2	3	4	5
1		✓	✓	✓	
2	✓		✓	✓	
3	✓			✓	✓
4	✓	✓	✓		
5	✓		✓	✓	
Count	**4**	**2**	**4**	**4**	**1**
S	**1.3333**	**0.6667**	**1.3333**	**1.3333**	**0.3333**

TABLE 2. NEIGHBORS OF BLUE BULLETS IN FIGURE 3

	A	B	C	D	E
A		✓	✓	✓	
B	✓		✓	✓	
C	✓	✓		✓	
D		✓	✓		✓
E		✓	✓	✓	
Count	**2**	**4**	**4**	**4**	**1**
S	**0.6667**	**1.3333**	**1.3333**	**1.3333**	**0.3333**

Here is a pseudocode describing our proposed method in figure 4.

Input:

A training set $X = \{x_i\}_{i=1}^{n}$ with n instances $x_i \in \mathbb{R}^d$ and $Y = \{y_i\}_{i=1}^{n}$, $y_i \in \{-1,+1\}$ as target of classes.

Constant C and K.

A test set X_{test}

Output:

Predicted label set $\hat{Y}$ for X_{test}

Algorithm:

Step 1. Reduce dimension of X using LLE with K as number of neighbors.

Step 2.

Initialize S with 0 of length n.

for i = 1 to n

By using KNN in reduced space:

if I be indexes of K nearest instances to x_i

$S[I] \leftarrow S[I]+1$

end

Step 3. Normalize S so that has the average equal to C.

Step 4. Train FSVM Using X and Y from input and normalized S as weights for slack variables.

Step 5. Evaluate label set $\hat{Y}$ for X_{test}

Figure 4. Pseudo code implementation for the proposed method.

IV. Experimental Validation

Experiments are carried out on both real and synthetic data. For all of these, we used the k-Fold Cross validation method with k=10 (number of folds) and also repeated all the experiments ten times to increase confidence. We also used 50 percent rate of feature reduction in LLE phase and considered constant C = 1 in classical SVM. The penalty vector S in our method was normalized in a way that we have an average equal to 1 for being fair in comparison with SVM.

A. Classification Experiments on Synthetic Data

We conducted classification experiments on synthetic data to understand the behavior of the proposed method. The synthetic database was generated in 2D and 3D models as appears in figures 5 and 6.

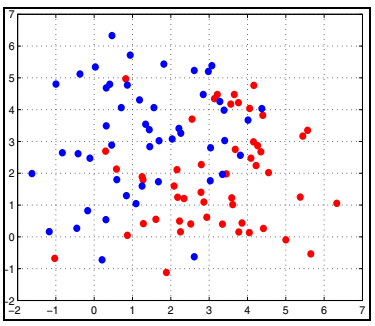

Figure 5. 2D synthetic data set. The blue inctances show class 1 and reds class 2. The instances are generated such that the optimal line be X = Y.

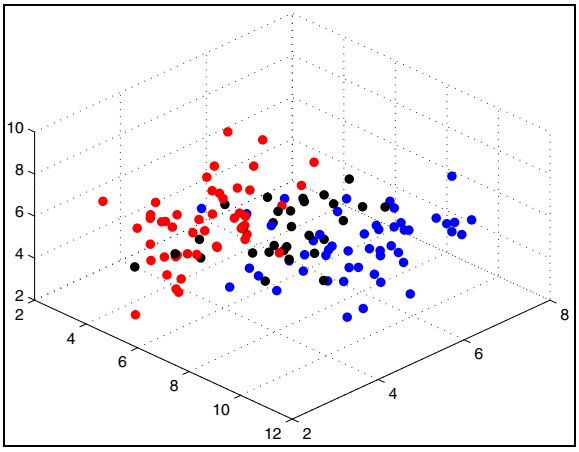

Figure 6. 3D synthetic data set. The blue inctances show class 1 and reds are class 2 and the black instances are noise data that have label 1 or 2 randomly.

Table 3. Experimental results on Synthetic datasets.

Dataset	SVM	Proposed Method
2D (Synthetic Dataset)	**96.52**	**96.92**
3D (Synthetic Dataset)	**98.90**	**99.20**

Table 4. Experimental results on Synthetic datasets (corrupted with noise)

Dataset (corrupted with noise)	SVM	Proposed Method
2D (Synthetic Dataset)	**81.79**	**82.90**
3D (Synthetic Dataset)	**85.75**	**86.75**

B. Classification Experiments on UCI Databases

We next performed classification experiments on five databases chosen from the UCI repository [29] to test performance of two models.

The database details are summarized in Table 5.

Table 5. UCI database detailes.

Database	No. of classes	No. of instances	No. of features
Pima Indians Diabetes	**2**	**768**	**8**
Heart	**2**	**270**	**13**
Cancer	**2**	**699**	**9**
Fisheriris	**3**	**150**	**4**
Ionosphere	**2**	**351**	**34**

The results of experiments on real data on noise free condition is given in table 6.

To understand the behavior of our method in noisy environment, we adapted an additive Gaussian zero-mean noise with covariance equal to identity matrix (I) with the dimension of feature vector size, to all above datasets. The results of experiments are shown in table 7.

TABLE 6. EXPERIMENTAL RESULTS ON UCI DATABASES.

Dataset	SVM	Proposed Method
Pima Indians Diabetes	**75.39**	**76.56**
Heart	**81.85**	**83.70**
Cancer	**97.00**	**96.85**
Fisheriris	**100**	**100**
Ionosphere	**86.08**	**86.87**

TABLE 7. EXPERIMENTAL RESULTS ON UCI DATABASES (WITH ADDITIVE GAUSSIAN ZERO-MEAN NOISE WITH COVARIENCE EQUAL TO IDENTITY MATRIX)

Dataset (corrupted with noise)	SVM	Proposed Method
Pima Indians Diabetes	**74.35**	**75.40**
Heart	**77.41**	**80.74**
Cancer	**70.10**	**73.96**
Fisheriris	**57.00**	**61.00**
Ionosphere	**82.89**	**83.77**

As it can be observed, the proposed method gives the better classification accuracy for UCI databases and synthetic data. The privilege of this method is more clearly on datasets corrupted with normal noise $\mathcal{N}(0,1)$.

V. CONCLUSION

The traditional soft margin support vector machines, have a constant scalar penalty term C which causes all samples to have almost equal permission to give ξ to logically move and allocate a place which is proper for classification tasks. Therefore, it will not make it robust, facing noisy data that must move more. In this paper we proposed a new penalty vector (S) for FSVMs using LLE neighborhood concept to customize all samples having different penalty coefficient. This calculated values are sensitive to noisy and outlier data. It causes noisy data having more permission to move and allocate a proper location which leads to robustness in comparison with SVMs. Experiments on synthetic and real data and also noisy data, show that the proposed approach most of the times is better than (or at least as good as) the classical version of SVM in terms of accuracy.

REFERENCES

[1] A. Chaudhuri and K. De, "Fuzzy support vector machine for bankruptcy prediction," *Applied Soft Computing,* vol. 11, pp. 2472-2486, 2011.

[2] V. Vapnik, "200 0," *The Nature of Statistical Learning Theory,* 1995.

[3] V. N. Vapnik and V. Vapnik, *Statistical learning theory* vol. 1: Wiley New York, 1998.

[4] V. Vapnik, "The support vector method of function estimation," in *Nonlinear Modeling*, ed: Springer, 1998, pp. 55-85.

[5] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony, "Structural risk minimization over data-dependent hierarchies," *Information Theory, IEEE Transactions on,* vol. 44, pp. 1926-1940, 1998.

[6] P. Bartlett and J. Shawe-Taylor, "Generalization performance of support vector machines and other pattern classifiers," *Advances in Kernel Methods—Support Vector Learning,* pp. 43-54, 1999.

[7] T.-Y. Wang and H.-M. Chiang, "Fuzzy support vector machine for multi-class text categorization," *Information Processing & Management,* vol. 43, pp. 914-929, 2007.

[8] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 144-152.

[9] L. J. Cao, H. P. Lee, and W. K. Chong, "Modified support vector novelty detector using training data with outliers," *Pattern Recognition Letters,* vol. 24, pp. 2479-2487, 2003.

[10] J. Weston and R. Herbrich, "Adaptive margin support vector machines," *Advances In Neural Information Processing Systems,* pp. 281-296, 1999.

[11] A. J. Smola, *Advances in large margin classifiers*: MIT press, 2000.

[12] C.-F. Lin and S.-D. Wang, "Fuzzy support vector machines," *Neural Networks, IEEE Transactions on,* vol. 13, pp. 464-471, 2002.

[13] I. Guyon, N. Matic, and V. Vapnik, "Discovering Informative Patterns and Data Cleaning," ed, 1996.

[14] X. Zhang, "Using class-center vectors to build support vector machines," in *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, 1999, pp. 3-11.

[15] H.-P. Huang and Y.-H. Liu, "Fuzzy support vector machines for pattern recognition and data mining," *Intl. J. of Fuzzy Systems,* vol. 4, pp. 826-835, 2002.

[16] T.-S. Lee, C.-C. Chiu, C.-J. Lu, and I.-F. Chen, "Credit scoring using the hybrid neural discriminant technique," *Expert Systems with applications,* vol. 23, pp. 245-254, 2002.

[17] T. Van Gestel, J. A. Suykens, G. Lanckriet, A. Lambrechts, B. De Moor, and J. Vandewalle, "Bayesian framework for least-squares support vector machine classifiers, Gaussian processes, and kernel Fisher discriminant analysis," *Neural Computation,* vol. 14, pp. 1115-1147, 2002.

[18] D. Tsujinishi and S. Abe, "Fuzzy least squares support vector machines for multiclass problems," *Neural Networks,* vol. 16, pp. 785-792, 2003.

[19] Y. Wang, S. Wang, and K. K. Lai, "A new fuzzy support vector machine to evaluate credit risk," *Fuzzy Systems, IEEE Transactions on,* vol. 13, pp. 820-831, 2005.

[20] Q. Tao and J. Wang, "A new fuzzy support vector machine based on the weighted margin," *Neural Processing Letters,* vol. 20, pp. 139-150, 2004.

[21] L.-L. Wei, W.-J. Long, and W.-X. Zhang, "Fuzzy data domain description using support vector machines," in *Machine Learning and Cybernetics, 2003 International Conference on*, 2003, pp. 3082-3085.

[22] C. H. Zheng and L. C. Jiao, "Fuzzy pre-extracting method for support vector machine," in *Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on*, 2002, pp. 2026-2030.

[23] M.-D. Shieh and C.-C. Yang, "Multiclass SVM-RFE for product form feature selection," *Expert Systems with Applications,* vol. 35, pp. 531-541, 2008.

[24] S.-J. Zhao, H.-P. Zhang, and L. Li, "A new algorithm for imbalanced datasets in presence of outliers and noise," in *Natural Computation (ICNC), 2012 Eighth International Conference on*, 2012, pp. 30-34.

[25] F.-f. MA, W.-g. TONG, and Y.-q. SONG, "The Research and Application of Fuzzy Support Vector Machine (FSVM)[J]," *Computer and Information Technology,* vol. 1, p. 008, 2013.

[26] W.-Y. Cheng and C.-F. Juang, "A fuzzy model with online incremental SVM and margin-selective gradient descent learning for classification problems," *Fuzzy Systems, IEEE Transactions on,* vol. 22, pp. 324-337, 2014.

[27] W. An and M. Liang, "Fuzzy support vector machine based on within-class scatter for classification problems with outliers or noises," *Neurocomputing,* vol. 110, pp. 101-110, 2013.

[28] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science,* vol. 290, pp. 2323-2326, 2000.

[29] C. Blake and C. J. Merz, "{UCI} Repository of machine learning databases," 1998.