

Contents list available at JMCS

Journal of Mathematics and Computer Science

Journal Homepage: www.tjmcs.com



A New Fuzzy Membership Assignment Approach for Fuzzy SVM Based on Adaptive PSO in Classification Problems

Omid Naghash Almasi^{1*}, Hamed Sadeghi Gooqeri², Behnam Soleimanian Asl³, Wan Mei Tang⁴

¹Young Researchers and Elite Club, Mashhad Branch, Islamic Azad University, Mashhad, Iran
o.almasi@ieee.org

²Department of Electrical Engineering, Bandar Abbas Branch, Islamic Azad University, Bandar Abbas, Iran
sadeghi802@gmail.com

³Department of Electrical Engineering, Gonabad Branch, Islamic Azad University, Mashhad, Iran
behnam.soleimanian@gmail.com

⁴Department of Computer and Information Science, Chongqing Normal University, Chongqing 400047, People's Republic of China
cqtwm@163.com

Article history:

Received October 31 2014

Accepted 27 December 2014

Available online 27 December 2014

Abstract

Noises will confuse Support Vector Machine (SVM) in the training phase. To overcome this problem, SVM was extended to Fuzzy SVM (FSVM) by incorporating an appropriate fuzzy membership to each data point. Thus, how to choose a proper fuzzy membership is of paramount importance in FSVM. In this paper, Adaptive Particle Swarm Optimization (APSO) method minimizes the generalization error by changing the attributes values of positive and negative class centers to make them free of attribute-noise. As the APSO converged, the fuzzy memberships are assigned for each training data points based on their distance to the corresponding purified class centers with the same class-label. To demonstrate the effectiveness of the proposed FSVM, its performance on artificial and real-world data sets is compared with three FSVM algorithms in the literature.

Keywords: Fuzzy support vector machine; Fuzzy membership function; Adaptive particle swarm optimization; Attribute-noise.

1. Introduction

SVM was developed by Vapnik and his colleagues [1, 2]. SVM minimizes the structural risk rather than the empirical risk. Vapnik demonstrates that generalization error is bounded by the sum of the empirical error and a confidence interval term, which depends on the Vapnik–Chervonenkis (VC)

dimension [3]. He proved that the SVM gets better generalization performance by minimizing that bound.

Training SVM is equivalent to solving a convex quadratic problem, which has the significant computational benefit of not getting stuck in local minima in comparison with traditional neural networks. A complete tutorial on SVM classifier has been published by Burges [4].

One of the main drawbacks of the SVM is that its training phase is sensitive to the existing outliers and noises data in the training data set [5]. In some real-world data sets, neither of the training points exactly belongs to any of the two classes, because of existing outliers or noises. For instance, one training data point may belong 90% to the positive class and be 10% irrelevant to that class or belongs to the negative class.

Noises are irrelevant or meaningless data points in training data set and confuse SVM algorithm in the training phase [6]. Accordingly, accuracy and generalization ability are noticeably reduced [7, 8, 9]. Thus, an important phase associated with using of SVM is reducing the effect of noisy data in training data set [7, 10].

Generally, noisy data in the classification problems could be organized in three groups [10-14]. i) Data that their corresponding labels include noise (paradoxical labeling error for a data point or misclassifications errors . ii) Data that their attribute values get noisy. iii) Data that have noise both in their class labels and in their attributes.

Lin et al. reformulate SVM to FSVM by associating a fuzzy membership to each data point [15]. Fuzzy membership of the data point is specified by the distance between the point and its fixed class center. However, the class center is very sensitive to noises. It was proved that the FSVM has a better performance in comparison to the SVM encounter noises. But, there exists a problem in FSVM about how to generate appropriate fuzzy membership functions to cope with all the classes of noise. In Ref. [16], two factors named confident and trashy were introduced for the automatic determining of fuzzy memberships in FSVM. In this approach, large computation in high-dimensional feature space is needed and many parameters must be optimized, which makes it hard and complex to implement. Jiang et al. designed a new fuzzy membership function by a kernel extension of the FSVM formulation of Lin [17]. Based on the fixed class centers in Ref. [18], a fuzzy membership function is developed for separable and non-separable data sets in input space and feature space, respectively. It could be considered as a good extension of Refs. [15] and [17]. The optimal hyper-plane is constructed by a small part of data called Support Vectors (SVs) which are laid in the convex hull of each class in the feature space the same as outlier and third groups of noise. Therefore, in those approaches SVs and outliers could not be distinguished accurately. This will reduce the generalization performance and accuracy in the FSVM.

Moreover, attribute-noises have a tendency to occur more often in real-world data sets and there exists the risk of discarding the meaningful data points as noises or outliers. It may lead to loss of informative data. Just because a noisy data point contains noise in its attribute and class, it does not mean that this data point is completely meaningless and therefore needs to be removed from the other data.

In this paper, we propose a new fuzzy membership function based on the pure positive and negative class centers for FSVM in linear/nonlinear separable/non-separable classification problems. The class centers are purified by using APSO. The APSO method provides a variation in the position of class centers by changing their attributes values to make them free of attribute-noise. Therefore, the fuzzy memberships can properly generate and assign to each training data points based on their distance to the corresponding purified class centers with the same class-label.

This paper is organized as follows: Basic SVM and FSVM formulation for binary classification are reviewed in Sections 2 and 3, respectively. In Section 4, the APSO as an optimization method is introduced for realizing the proposed approach. The details of the new fuzzy membership function assignment for FSVM are discussed in Section 5. In Section 6, some experimental results are presented and discussed to illustrate the effectiveness of the proposed method. Some concluding remarks are given in Section 7.

2. Support Vector Machine

The mathematical formulation of SVMs for classification problems is reviewed in this section. Assume a two classes set Ω of labeled training points (x_i, y_i) is given. Each training point $x_i \in R^n$ belongs to either of two classes as determined by a corresponding label $y_i \in \{-1, 1\}$ for $i = 1, \dots, n$. The optimal hyper plane is obtained by solving quadratic optimization problem Eq. (1) (known as primal form), whose number of variables is as large as training data size n .

$$\begin{aligned} \text{Min } \varphi(w, \xi) &= \frac{1}{2} w^T w + C \sum_{i=1}^n \xi_i \\ \text{s. t.} & \\ y_i (w^T \cdot x_i + b) &\geq 1 - \xi_i, & i = 1, 2, \dots, n \\ \xi_i &\geq 0, & i = 1, 2, \dots, n \end{aligned} \tag{1}$$

where ξ_i s are slack variables representing the violation of pattern separation condition. The user defined parameter C is regarded as a regularization parameter controlling the model complexity. For nonlinear separable data, a kernel trick is utilized to map the input space into a high dimensional space named feature space. The optimal hyper-plane is obtained in the feature space. The primal optimal problem Eq. (1) can be transformed into dual form as follows:

$$\begin{aligned} \text{Max } Q(\alpha) &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{j=1}^n \alpha_j \\ \text{s. t.} & \\ \sum_{j=1}^l \alpha_j y_j &= 0 \\ 0 &\leq \alpha_i \leq C, & i = 1, \dots, n \end{aligned} \tag{2}$$

where $K(\dots)$ is a kernel function. In practical applications of SVMs, there are several frequently substitutions for selecting the kernel function $K(\dots)$ among. Some of the conventional kernel functions are listed in Table 1. In this Table, σ and d are constant and those parameters must be set by a user. For MLP kernel a suitable choice for β_0 and β_1 is needed to enable the kernel function to satisfy Mercer's condition [19, 20]. Furthermore, in Eq. (2) $\alpha = (\alpha_1, \dots, \alpha_n)$ is the vector of non-negative Lagrange multipliers. The solution vector $\alpha = (\alpha_1, \dots, \alpha_n)$ is sparse, i.e. $\alpha_i = 0$ for most indices of training data. This is the so-called SVM sparseness property. The points x_i corresponds to nonzero α_i are called support vectors. Therefore, the points x_i corresponds to $\alpha_i = 0$ have no contribution in construction of the optimal hyper-plane and only part of training data, support vectors, construct the optimal hyper-plane. Let v be the index set of support vectors, then the optimal hyperplane is

$$f(x) = \sum_{i \in v}^{\#sv} \alpha_i y_i K(x_i, x_j) + b = 0 \tag{3}$$

and the resulting classifier is

Table 1- The conventional kernel functions

Name	Kernel function expression
Linear Kernel	$K(x_i, x_j) = x_i^T x_j$
Polynomial Kernel	$K(x_i, x_j) = (t + x_i^T x_j)^d$
RBF Kernel	$K(x_i, x_j) = \exp(-\ x_i - x_j\ ^c / \sigma^2)$
MLP (*) Kernel	$K(x_i, x_j) = \tanh(\beta_0 x_i^T x_j + \beta_1)$

$$y(x) = \text{sgn} \left[\sum_{i \in v}^{\#sv} \alpha_i y_i K(x_i, x_j) + b \right] \tag{4}$$

where b is easily determined by KKT conditions.

3. Fuzzy SVM

In many real-world applications, each data point is not fully classified to one of the two classes. Based on this fact, Lin extended the theory of classical SVM to Fuzzy SVM (FSVM) [15]. In FSVM, each data point can make a different contribution to the construction of the optimal hyper-plane in contrast SVM, with all data points having the same effect on the optimal decision surface. To materialize this idea, fuzzy memberships are assigned to each data point to make them have different importance. Assume the training data in the following form:

$$\Omega = \{(x_i, y_i, s_i), i = 1, \dots, n\} \tag{5}$$

where $x_i \in R^n$ and y_i are a training sample and its corresponding label. s_i is represented a fuzzy membership satisfying $\sigma \leq s_i \leq 1$ with adequately positive small constant σ . The optimal hyper-plane problem in FSVM is regarded as the solution to

$$\begin{aligned} \text{Min } \varphi(w, \xi) &= \frac{1}{2} w^T w + C \sum_{i=1}^n s_i \xi_i \\ \text{s. t.} \\ y_i (w^T \cdot x_i + b) &\geq 1 - \xi_i, & i = 1, 2, \dots, n \\ \xi_i &\geq 0, & i = 1, 2, \dots, n \end{aligned} \tag{6}$$

where C is a constant. The main difference between SVM and FSVM arises from $s_i \xi_i$ term. Since ξ_i is known as a measure of error in the SVM, then the fuzzy membership s_i makes a measure of error which has different weights in FSVM. Note that by adjusting the value of fuzzy membership s_i can reduce or increase the effect of each training data point. Similar to the SVM formulation, by some manipulating, the solution of FSVM is obtained. Also, the FSVM can be solved by its dual form. For meticulous detail see Ref. [15, 16].

4. Adaptive Particle Swarm Optimization

PSO is a population-based optimization technique. It uses swarm of particles to find a global optimum solution in search space [24]. Each particle represents a candidate solution to the cost function and it has its own position and velocity. Assume particle swarms are in D -dimensional search space. Let the i -th particle in a D -dimensional space be represented as $x_i = (x_{i1}, \dots, x_{id}, \dots, x_{id})$. The best previous position of the i -th particle is recorded and represented as $p_{bi} = (p_{bi1}, \dots, p_{bid}, \dots, p_{bid})$, which gives the best value in the cost function and also called p_{best} . General best position, G_{best} , denoted by p_{gb} is the best value of the P_{best} among all the particles in the cost function. The velocity for the i -th particle is represented as $v_i = (v_{i1}, \dots, v_{id}, \dots, v_{id})$. In each of the iterations, the velocity and the position of each particle are updated according to Eq.(7) and Eq.(8), respectively.

$$v_{id} = wv_{id} + c_1r_1(p_{bid} - x_{id}) + c_2r_2(p_{gb} - x_{id}) \tag{7}$$

$$x_{id} = x_{id} + v_{id} \tag{8}$$

where w is called the inertia coefficient and it is in the interval $[0,1]$. We reduce the inertia weight, w , linearly during the iterations to enhance the search ability of the PSO.

$$w(ite\text{r}) = w_{max} - \frac{w_{max} - w_{min}}{Iter_{max}} \times ite\text{r} \tag{9}$$

where $ite\text{r}$ denoted the iteration of the algorithm and $ite\text{r}_{max}$ is the maximum number of algorithm's iteration [24,25]; C_1 and C_2 are non-negative constants of acceleration; r_1 and r_2 are generated randomly in the interval $[0,1]$; $v_{id} \in [-v_{max}, v_{max}]$ and v_{max} is the maximum velocity. The termination criterion of the PSO is determined by reaching the maximum iterations number. After the maximum number of iterations reached, global best particle represents the optimal solution consisting of the optimal positive and negative class centers.

In order to control the trade-off between exploitation and exploration property of the PSO algorithm the beginning of the optimization procedure, a large value for the cognitive weight and a small value for the social weight should be chosen. Hence, the exploration property of the PSO is enhanced. By contrast, near the final stage of the PSO algorithm, a small cognitive weight and a large social weight should be selected for the convergence of the algorithm to the global optimum point [26]. Therefore, it is necessary to change the cognitive weight and social weight during the optimization process adaptively. The following formula for the APSO is proposed [27, 28]:

$$\begin{aligned} &\text{If } C_1^{final} < C_1^{initial} \\ C_1 &= (C_1^{final} - C_1^{initial}) \left(\frac{ite\text{r}}{ite\text{r}_{max}} \right) + C_1^{initial} \end{aligned} \tag{10}$$

$$\begin{aligned} &\text{If } C_2^{final} > C_2^{initial} \\ C_2 &= (C_2^{final} - C_2^{initial}) \left(\frac{ite\text{r}}{ite\text{r}_{max}} \right) + C_2^{initial} \end{aligned} \tag{11}$$

where the superscripts "initial" and "final" indicate the initial and final values of the cognition weight and the social weight factor, respectively. The parameters of the APSO set as follows: $C_1^{initial} = 0.7$, $C_1^{final} = 1.6$, $C_2^{initial} = 0.9$, $C_2^{final} = 1.7$. In this study, the maximum number of iterations and the number of particles are selected to be 100 and 25, respectively.

5. Fuzzy membership assignments

How to determine appropriate fuzzy memberships to be used in FSVM is a significant problem. Basically, the lower bounds of fuzzy memberships are defined, and then the main property of each data point is chosen and a connection is made between this property and fuzzy memberships function. As a result, FSVM can achieve good performance and therefore discards the effect of noise

Table 2- The structure of each particle of APSO

	Par. 1	...	Par. m	...	Par. m+1	...	Par. n+2
Min value	$MeanPos_{i1} - \alpha$...	$MeanPos_{im} - \alpha$...	$MeanNeg_{im+1} - \alpha$...	$MeanNeg_{in+2} - \alpha$
	x_{i1}	...	x_{im}	...	x_{im+1}	...	x_{in+2}
Max value	$MeanPos_{i1} + \alpha$...	$MeanPos_{im} + \alpha$...	$MeanNeg_{im+1} + \alpha$...	$MeanNeg_{in+2} + \alpha$
	Positive class center			Negative class center			

and outliers, if the fuzzy memberships function prepares the fuzzy memberships accurately and appropriately.

Generally, fuzzy memberships are generated by setting the fuzzy membership as a function of the distance between the data point and its fixed class center [15,16,17,18]. Many fuzzy membership functions have been proposed based on this idea. Although those methods could cope with outliers or misclassification noise in classification problem, they could not deal with attribute-noise.

To overcome this problem and generate fuzzy membership function so that enhances the performance of FSVM, APSO is used to change the attributes of positive and negative class centers in such way that minimizes the generalization performance error. After the APSO is converged, the fuzzy membership functions are generated based on the positions of purified class centers. Generally, each data set is divided into two classes and each of them contains some attributes, so the class center is a vector of average attributes. Define the mean of positive class label as x_+ and the mean of negative class as x_- as follow:

$$x_+ = \frac{1}{n_+} \sum_{x_i \in C^+} x_i \tag{12}$$

and

$$x_- = \frac{1}{n_-} \sum_{x_i \in C^-} x_i \tag{13}$$

where n_+ and n_- are the numbers of data points in class C^+ and C^- , respectively. In all previous studies, researchers focused on extending the class center-based fuzzy membership assignment in which the positions of the class centers are fixed. Thus, the noise data and outliers are discarded and eliminated by their distance from their corresponding class centers. It is transparent that the class centers of a noisy data set will be noisy and not be adequately pure to be used as reference measure for producing fuzzy memberships. In this study we propose to vary the position of the class centers by using APSO method to de-noise the class centers attributes. Moreover, by modifying the position of the class centers, their data points within each class could achieve different weights relevant to its class center. For this aim, we considered a lower and upper bound for class centers attributes and APSO used for this goal to change the mean of the class center attributes within those bounds. In other words, by altering attributes the class center place is replaced and the class centers will have dynamic behaviors. The structure of each particle of APSO is shown in Table 2.

As shown in Table 2, each particle consists of two parts. The first part of dimensions of each particle is considered for finding the optimal mean attribute of positive class center and the rest dimensions of each particle are used for finding the optimal mean attribute of negative class center. First, mean of each positive and negative class centers are calculated, then a small positive constant, which is so-called α , is added and subtracted from these means to obtain the upper and lower bounds of each features in positive and negative class centers. The value of α is 0.1 in this study.

Define the maximum radius of positive class C^+ by

$$r_+ = \max|x_+ - x_i| \text{ for } x_i \in C^+ \quad (14)$$

where x_i belongs to C^+ and the maximum radius of negative class defined as follows:

$$r_- = \max|x_- - x_i| \text{ for } x_i \in C^- \quad (15)$$

where $x_i \in C^-$. Consequently, fuzzy membership s_i defined to be a function of the mean and radius of each class in the following form:

$$s_i = \begin{cases} 1 - \|x_+ - x_i\|/(r_+ + \delta) & \text{if } y = 1 \\ 1 - \|x_- - x_i\|/(r_- + \delta), & \text{if } y = -1 \end{cases} \quad (16)$$

where δ is a small positive constant which used to avoid the case $s_i = 0$. The value of this parameter is selected to be 0.01, in this paper. The generalization performance error is regarded as the objective function as follows:

$$\text{CostFunction} = K\text{-fold CV Error} \quad (17)$$

Various criteria proposed by many researchers such as Leave One Out (LOO) [21], and K -fold Cross Validation (CV) method for evaluating the generalization performance of SVM [22,23]. In K -fold CV method, the training data set randomly is divided into K equal subsets and then for K iteration each subset is defined as a testing data set and retain subsets, $K-1$, are used as a training data set. After K iterations, the overall generalization performance is averaged over K calculated performances. Totally, each part of the entire training data set is separately considered as testing data set. Hence, K -fold CV is a robust criterion for evaluating generalization performance. The K value is equal to 10 in this study. Algorithm 1 describes the proposed method.

Algorithm 1: Proposed method

- 1: Initialize FSVM model parameters, i.e. C and kernel parameters
 - 2: Initialize APSO population, i.e., positive and negative class centers
 - 3: Generate fuzzy memberships based on initial class centers
 - 4: Train FSVM with initial model parameters
 - 5: Evaluate the cost function Eq. (17)
 - 6: Update the initial population based on the procedure of APSO until the optimization approach terminate
 - 7: Use the Gbest of the APSO containing the optimal positive and negative class centers which fuzzy memberships are generated based on them
-

6. Computational Experiments

6.1 Experimental Conditions

To perform the effectiveness of the proposed cost function, a PC with MATLAB software R2008b is utilized. Experiments on artificial data and real data sets will be carried out to compare the performance of FSVM using the fuzzy memberships calculated in the input space as that in Ref. [15] (FSVM-1), FSVM-2 which is computed fuzzy memberships in the feature space [17], FSVM-3, which is calculated the fuzzy memberships in both input space and feature space for linear and nonlinear cases [18], and our proposed method FSVM-4.

In order to provide the same condition to compare the results, similar to Ref. [18] only RBF and Polynomial kernel functions are used. The experiment on synthetic data set reporting here is implemented using RBF kernel and the other test on real data set using both RBF and Polynomial kernel functions. However, the proposed method could be applied to all the existing kernel functions [29,30,31]. Similar to Ref. [18] different selections of the regularization parameters are used for the training to compare the performance of the FSVM-4 with the FSVM-1, the FSVM-2, and FSVM-3.

6.2. Experiments on Artificial Data Set

We use the same artificial data set in Ref. [18]. Ref. [18] generates 200 points with some given outlier points in $R^6 \times \{-1, +1\}$. 120 points are selected as training points and the remaining 80 points are selected as testing set. In the training set, 60 points including six outliers belong to one class and the other points including three outliers belong to another.

The experiment results for the FSVM-4 are compared with FSVM-1, FSVM-2, and FSVM-3 (feature space form) in Table 3. Results show that the FSVM-4 outperforms the same generalization performance in comparison to the FSVM-3 and more better accuracy from the others. But the number of Support Vectors (SVs) is smaller than the others FSVMs in FSVM-4. Although the accuracy rate is of primary importance to the number of SVs in evaluation of the FSVM methods, less number of SVs means FSVM-4 has a smaller model size. By a small model size, the time consumed in testing phase is reduced and the memory storage required is decreased. Therefore, it could be concluded that FSVM-4 has a better performance in reducing the effect of the outliers and de-noising the attributes noise.

Figure 1 shows the variation attributes of the positive and negative class centers in FSVM-4 for both fixed and variable class centers ideas with $C=100$ and $\sigma=2$. The figure consists of four subplots. In the figure, left-upper and right-upper subplots respectively show the fixed positive and negative class centers and left-lower and right-lower subplots illustrate the positive and the negative class centers after applying the proposed method. The optimal cost function is obtained in FSVM-4 based on the purified class centers in comparison with the fixed class centers. As shown in this figure, the proposed method properly de-noise the new positive and negative class centers from attribute-noise and also the fuzzy memberships which is generated based on the new class centers could appropriately discard the boundary data points from class-noise. The convergence of the objective function versus the numbers of iterations during the procedure of APSO is shown in Figure 2.

Table 3- Experimental results of the FSVM-1 [15], the FSVM-2 [17], the FSVM-3 (feature space form) [18], and the FSVM-4 on the artificial data set with Gaussian RBF kernel function (width=2)

C	FSVM-1		FSVM-2		FSVM-3*		FSVM-4	
	SVs	Testing Acc. (%)						
10	43(35.8%)	93.8	44(36.7%)	93.8	43(35.8%)	93.8	15(12.2%)	93.8
50	42(35.0%)	93.8	45(37.5%)	93.8	44(36.7%)	93.8	17(13.9%)	93.8
100	40(33.3%)	92.5	44(36.7%)	93.8	44(36.7%)	93.8	17(13.9%)	93.8
300	38(31.7%)	91.3	39(32.5%)	92.5	44(36.7%)	93.8	14(11.4%)	93.8
400	38(31.7%)	91.3	38(31.7%)	91.3	44(36.7%)	93.8	15(12.2%)	93.8
500	38(31.7%)	91.3	38(31.7%)	91.3	45(37.5%)	93.8	16(13.1%)	93.8
600	38(31.7%)	91.3	38(31.7%)	91.3	45(37.5%)	93.8	16(13.1%)	93.8
1000	38(31.7%)	91.3	38(31.7%)	91.3	43(35.8%)	93.8	17(13.9%)	92.5

*FSVM-3 is in the input space from

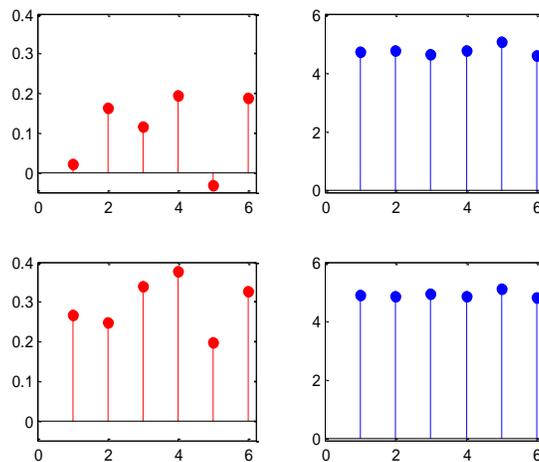


Figure 1- Comparison between the variation of the fixed positive class center (left-upper column) and the new positive class center after applying the proposed approach (left-lower column) and the fixed negative class center (right-upper column) and the new negative class center from the proposed approach (right-lower column) for FLSSVM-4 in artificial data set.

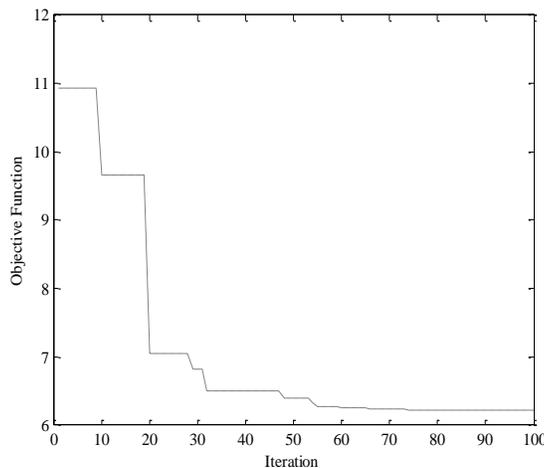


Figure 2- The convergence of objective function via APSO in artificial data set

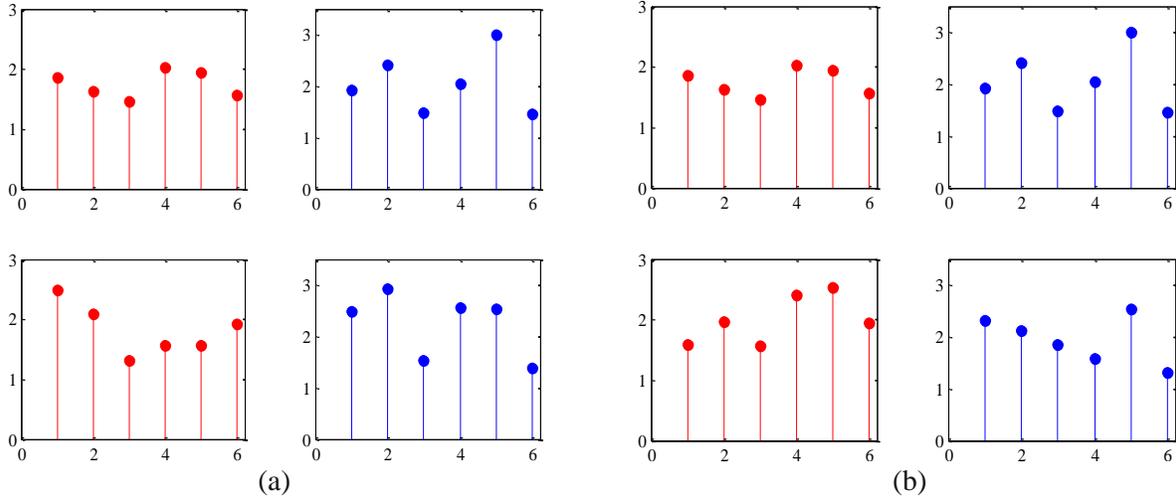


Figure 3- Comparison between the variation of the fixed positive class center (left-upper column) and the new positive class center after applying the proposed approach (left-lower column) and the fixed negative class center (right-upper column) and the new negative class center from the proposed approach (right-lower column) for FSVM-4 in Monks data set. a) RBF kernel with $C=100$ and $\sigma=2$. b) Polynomial kernel with $C=120$, $t=1$, and $d=2$.

6.3. Experiments on Real Data Set

The MONK problem was the basis of the first international comparison of learning algorithms. There are three MONK problems. In this paper, the third training and testing data sets of MONK problem which contains 5% random noise among the samples are chosen from the UCI Repository [32]. Each sample is represented by six discrete-valued attributes. This is a two-class data set. The training set consists of 122 patterns (60 patterns belonging to the positive class and 62 patterns belonging to the negative class), and the testing set consists of 432 patterns (228 patterns belonging to the positive class and 204 patterns belonging to the negative class). Table 4 shows the experimental results of the algorithm of the FSVM-1, FSVM-2, FSVM-3 (both input and feature space forms), and the FSVM-4 using RBF kernel and polynomial kernel.

The results demonstrate that the proposed FSVM-4 has a considerable better generalization performance in comparison with the other FSVM methods. FSVM-3 which has input and feature.

Table 4- Experimental results of the FSVM-1 [15], the FSVM-2 [17], the FSVM-3 (both input and feature space forms) [18], and the FSVM-4 on the Monks data set

Kernel	C	FSVM-1		FSVM-2		FSVM-3*		FSVM-3**		FSVM-4	
		SVs	Testing Acc. (%)								
RBF ($\sigma = 2$)	10	40(32.8%)	77.7	32(26.2%)	80.3	40(32.8%)	85.2	39(32.0%)	91.9	55(45.0%)	97.2
	50	31(25.4%)	84.8	30(24.6%)	85.2	41(33.6%)	91.4	34(27.9%)	94.7	47(38.5%)	97.2
	100	25(20.5%)	84.3	26(21.3%)	84.5	35(28.7%)	93.3	35(28.7%)	94.4	48(39.3%)	97.2
	500	39(32.0%)	83.5	34(27.9%)	84.8	35(28.7%)	95.6	35(28.7%)	93.5	46(37.7%)	96.5
Poly ($t = 1$, $d = 2$)	10	48(39.3%)	78.1	41(33.6%)	83.0	41(33.6%)	94.2	26(21.3%)	94.2	26(21.3%)	96.7
	120	27(22.1%)	80.3	25(20.5%)	80.3	12(9.8%)	49.5	19(15.6%)	94.0	24(19.6%)	95.1
	250	33(27.0%)	81.2	35(28.7%)	83.9	12(9.8%)	46.3	19(15.6%)	89.1	25(20.4%)	94.2
	800	42(34.4%)	79.6	44(36.1%)	80.6	15(12.3%)	91.0	21(17.2%)	89.8	25(20.4%)	94.2

*FSVM-3 is in the input space from

** FSVM-3 is in the feature space form

forms, reaches to the higher accuracy in its feature space forms compared to its inputs form.

FSVM-1, FSVM-2, especially FSVM-3 (input form), and FSVM-3 (feature form) are degenerated whereas the Polynomial kernel is utilized. FSVM-4 is less sensitive to the kernel type and the value variation of the parameter C . Figures 3(a) and 3(b) show the variation attributes of the positive and negative class centers in FSVM-4 for both fixed and variable class centers ideas with RBF and Polynomial Kernels, respectively.

7. Conclusions

In FSVMs, different training points can make different contributions to the learning of the decision surface. Choosing a proper fuzzy membership function is the main issue in the FSVM for the classification problems. This paper propose a new fuzzy membership function for FSVM linear/nonlinear and separable/non-separable classification problem based on purified positive and negative class centers. To this end, an APSO is used to vary the attributes of both positive and negative class centers. The results demonstrate that the effect of attribute-noise and the outlier data can be reduced. In addition, the proposed FSVM can achieve higher generalization performance with lower model size (#SVs) in comparison with the other FSVMs. It makes FSVMs more feasible for the real applications, containing noise in attribute and the class of data points, in which there is not enough information to choose appropriate fuzzy memberships.

Further work on the extension of the proposed fuzzy membership assignment function to feature space and thorough the APSO making feature/data reduction and solving the model selection problem of FSVM are under the author's investigation.

References

- [1] V.N. Vapnik, *Statistical learning theory*, New York: Wiley, (1998).
- [2] I. Guyon, B. Boser, V. Vapnik, “*Automatic capacity tuning of very large VC-dimension classifier*”, *Adv. Neural Inform. Process. Systems*. 5 (1993) 147–155.
- [3] V.N. Vapnik, “*An overview of statistical learning theory*”, *IEEE Transaction on Neural Networks*. 10(5) (1999) 988–999.
- [4] C.J.C. Burges, “*A tutorial on support vector machines for pattern recognition*”, *Data Mining Knowl. Discovery*. 2(2) (1998) 955–974.
- [5] X.G. Zhang, “*Using class-center vectors to build support vector machines*”, In *Proc. IEEE Signal Process. Soc. Workshop*. NewYork: IEEE Press, (1999) 3–11.
- [6] H. Xiong, G. Pandey, M. Steinbach, V. Kumar, “*Enhancing Data Analysis with Noise Removal*”, *IEEE Transaction on Knowledge and Data Engineering*. 18(3) (2006) 304–319.
- [7] J.V. Hulse, T. Khoshgoftaar, “*Knowledge discovery from imbalanced and noisy data*”, *Data & Knowledge Engineering*. 68 (2009) 1513–1542.
- [8] M.E. Mavroforakis, S. Theodoridis, “*A Geometric Approach to Support Vector Machine (SVM) Classification*”, *IEEE Transaction on Neural Networks*. 17(3) (2006) 671–682.
- [9] A. Angelova, Y. Abu-Mostafa, P. Perona, “*Pruning training sets for learning of object categories*”, *IEEE Conference on Computer Vision and Pattern Recognition*. (2005) 494–501.
- [10] X. Zhu, X. Wu, “*Class noise vs. attribute noise: a quantitative study of their impacts*”, *Artificial Intelligence Review*. 22 (3–4) (2004) 177–210.
- [11] C.E. Brodley, M.A. Friedl, “*Identifying mislabeled training data*, *Journal of Artificial Intelligence Research*”. 11 (1999) 131–167.
- [12] J.V. Hulse, T.M. Khoshgoftaar, H. Huang, “*The pairwise attribute noise detection algorithm*”, *Knowledge and Information Systems Journal, Special Issue on Mining Low Quality Data*. 11(2) (2007) 171–190.

- [13] T.M. Khoshgoftaar, S. Zhong, V. Joshi, “*Enhancing software quality estimation using ensemble-classifier based noise filtering*”, *Intelligent Data Analysis: An International Journal*. 9(1) (2005) 3–27.
- [14] X. Wu, X. Zhu, “*Mining With Noise Knowledge: Error-Aware Data Mining*”, *IEEE Transaction on Man, and Cybernetics—PART A: Systems and Humans*. 38(4) (2008) 917–932.
- [15] C.F. Lin, S.D. Wang, “*Fuzzy support vector machines*”, *IEEE Transaction on Neural Networks*. 13(2) (2002) 464–471.
- [16] C.F. Lin, S.D. Wang, “*Training algorithms for fuzzy support vector machines with noisy data*”, *Pattern Recognition Letters*. 25(14) (2004) 1647–1656.
- [17] X.F. Jiang, Y. Zhang, J.C. Lv, “*Fuzzy SVM with a new fuzzy membership function*”, *Neural Comput. Appl.* 15 (2006) 268–276.
- [18] W.M. Tang, “*Fuzzy SVM with a new fuzzy membership function to solve the two-class problems*”, *Neural Processing Letters*. 34(3) (2011) 209–219.
- [19] K.M. Lin, C.J. Lin, “*A study on reduced support vector machines*”, *IEEE Transactions on Neural Networks*, 14(6) (2003) 1449–1459.
- [20] A. Bordes, S. Ertekin, J. Weston, L. Bottou, “*Fast kernel classifiers with online and active learning*”, *The Journal of Machine Learning Research*. 6 (2005) 1579–1619.
- [21] J. Weston, “*Leave-one-out support vector machines*”, In: Dean, T. (Ed.), *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99*. Morgan Kaufmann, (1999) 727–733.
- [22] M.W. Browne, “*Cross-validation methods*”, *Journal of Mathematical Psychology*. 44(1) (2000) 108–132.
- [23] D.C. Li, Y.H. Fang, Y.M. Fang, “*The data complexity index to construct an efficient cross-validation method*”, *Decision Support Systems*. 50(1) (2010) 93–102.
- [24] J. Kennedy, R.C. Eberhart, “*Swarm Intelligence*”, USA: Academic Press, (2001).
- [25] Y.H. Shi, R.C. Eberhart, “*A modified particle swarm optimizer*”, *IEEE World congress on computer intelligence*. (1998) 69–73.
- [26] A. Nickabadi, M.M. Ebadzadeh, R. Safabakhsh, “*A novel particle swarm optimization algorithm with adaptive inertia weight*”, *Applied Soft Computing*. 11(4) (2011) 3658–3670.
- [27] K.T. Chaturvedi, M. Pandit, L. Srivastava, “*Self-organizing hierarchical particle swarm optimization for non-convex economic dispatch*”, *IEEE Transaction on Power Systems*. 23 (3) (2008) 1079–1087.
- [28] A. Ratnaweera, S.K. Halgamuge, H.C. Watson, “*Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients*”, *IEEE Transaction on Evol. Comput.* 8(3) (2004) 240–255.
- [29] S.S. Keerthi, C.J. Lin, “*Asymptotic behavior of support vector machines with gaussian kernel*”, *Neural Computation*. 15(7) (2003) 1667–1689.
- [30] H.T. Lin, C.J. Lin, “*A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods*”, *Technical report, Department of computer science national Taiwan university*. (2003) 1–32.
- [31] S. Li, M. Tan, “*Tuning SVM parameters by using a hybrid CLPSO–BFGS algorithm*”, *Neurocomputing*. 73(10) (2010) 2089–2096.
- [32] C.L. Blake, C.J. Merz, “*UCI Repository of Machine Learning Databases*”, *Tech. Rep. Univ. California, Dept, Information and Computer Science, Irvine, CA*, (1998).