Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 1797 – 1814

doi:10.3906/elk-1310-253

A new fuzzy membership assignment and model selection approach based on dynamic class centers for fuzzy SVM family using the firefly algorithm

Omid NAGHASH ALMASI[1,*], **Modjtaba ROUHANI**[2]
[1]Young Researchers and Elite Club, Mashhad Branch, Islamic Azad University, Mashhad, Iran
[2]Department of Computer Engineering, Faculty of Engineering, Ferdowsi University, Mashhad, Iran

Received: 01.11.2013 • **Accepted/Published Online:** 30.06.2014 • **Final Version:** 23.03.2016

Abstract: The support vector machine (SVM) is a powerful tool for classification problems. Unfortunately, the training phase of the SVM is highly sensitive to noises in the training set. Noises are inevitable in real-world applications. To overcome this problem, the SVM was extended to a fuzzy SVM by assigning an appropriate fuzzy membership to each data point. However, suitable choice of fuzzy memberships and an accurate model selection raise fundamental issues. In this paper, we propose a new method based on optimization methods to simultaneously generate appropriate fuzzy membership and solve the model selection problem for the SVM family in linear/nonlinear and separable/nonseparable classification problems. Both the SVM and least square SVM are included in the study. The fuzzy memberships are built based on dynamic class centers. The firefly algorithm (FA), a recently developed nature-inspired optimization algorithm, provides variation in the position of class centers by changing their attributes' values. Hence, adjusting the place of the class center can properly generate accurate fuzzy memberships to cope with both attribute and class noises. Furthermore, through the process of generating fuzzy memberships, the FA can choose the best parameters for the SVM family. A set of experiments is conducted on nine benchmarking data sets of the UCI data base. The experimental results show the effectiveness of the proposed method in comparison to the seven well-known methods of the SVM literature.

Key words: Support vector machines, fuzzy support vector machine, fuzzy membership function, model selection problem, firefly algorithm, classification, noise

1. Introduction

The support vector machine (SVM) was developed by Vapnik et al. [1,2]. The SVM minimizes the structural risk instead of the empirical risk. Vapnik demonstrated that the generalization error is bounded by the sum of the empirical error and a confidence interval term, which depends on the Vapnik–Chervonenkis dimension [3], and he proved that the SVM achieves better generalization performance by minimizing that bound.

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

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

*Correspondence: o_almasi@yahoo.com

Noises are irrelevant or meaningless data points in a training data set [6]. Noise will confuse the machine-learning algorithm in the training phase. Accordingly, accuracy and generalization ability are noticeably reduced [7–9]. Thus, an important phase associated with using machine-learning algorithms, such as SVMs, is to reduce the effect of noisy data on the training data set [7,10].

Generally, noisy data in the classification problems can be organized into three groups [10–14]: 1) data whose corresponding labels include noise (paradoxical labeling error for a data point or misclassifications errors), 2) data points whose attribute values get noisy, and 3) data that simultaneously have noise in their class labels and in their attributes.

Lin et al. reformulated the SVM to the fuzzy SVM (FSVM) by associating a fuzzy membership to each data point [15]. The fuzzy membership of each data point is specified by the distance between the point and its class center. However, the class center is sensitive to noise. It was proven that the FSVM achieved a better performance in comparison with the SVM-encountered noise. However, there exists a problem in FSVM about how to generate appropriate fuzzy membership functions to cope with all the classes of noise. In [16], two factors, named 'confident' and 'trashy', were introduced for the automatic assignment 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 for implementation. Jiang et al. designed a new fuzzy membership function with a kernel extension of the FSVM formulation of Lin [17]. In [18], based on the class centers, a fuzzy membership function was developed for separable and nonseparable data sets in input space and feature space, respectively. In SVMs, the optimal hyperplane is constructed with a small portion of data called support vectors (SVs), which are laid in the convex hull of each class in the feature space, similarly to the outlier and third group of noise. Therefore, in those approaches, SVs and outliers could not be distinguished accurately. This will reduce the generalization performance and accuracy of the FSVM.

Moreover, attribute noises have a tendency to occur more often in real-world data sets, and there exists some risk of discarding the meaningful data points as noise or outliers. This 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 that it needs to be removed from the other data.

Another problem is the lack of a certain method to select the most suitable parameters in the family of SVMs. SVMs have two adjustable sets of parameters: the kernel parameter(s) and the regularization parameter (C in SVM and γ in LSSVM). SVM generalization ability depends on the proper choosing of these parameters. The best performance of a SVM is realized with an optimal choice of the kernel parameter(s) and the regularization parameter. The optimal choice of these parameters is called the SVM model selection problem [19–21].

Various model selection methods exist considering different criteria, such as the Opper–Winther bound [22], span bound [23], radius/margin bound [24], distance between two classes [25], and generalization performance (K-fold cross-validation (CV)) [26]. According to the solvers applied to those criteria, the methods for model selection of the SVM can be classified into two groups. The first contains classical (analytical) approaches and the second contains population-based optimization algorithms. The classical approaches, such as [19,23] and [25,27], use a gradient descent method to optimize the model selection criteria.

The optimization methods based on the gradient methods are fast, although for smooth cost functions the optimization algorithm may get stuck in local minima. Moreover, for nonsmooth cost functions (i.e. not differentiable ones), gradient methods are not applicable. Additional disadvantages of the gradient-based methods are memory usage and the need to invert the modified Gram matrix, as well as requiring a solution to the additional quadratic programming problem [19,21,27,28]. To overcome the drawbacks of the first group, a second group, based on population-based algorithms, has been introduced. In this group, global optimization techniques, such as PSO [29,30,31], adaptive chaos PSO [32], quantum PSO [33], simulated annealing [34], ant colony [35], and the GA [36–39], were employed for finding the optimal solution of the cost function so as to achieve the best model selection for the SVM.

Many studies have been performed on how to assign new fuzzy memberships and how to solve the model selection problem of the SVM in isolation; however, their combined effects have not been addressed before. The main contribution of this paper is to propose a novel method to solve both problems. In this method, generating appropriate fuzzy memberships and solving the model selection problem of the SVM family in linear/nonlinear and separable/nonseparable classification problems are defined as optimization problems. Moreover, the fuzzy memberships are generated based on dynamic class centers in contrast to previous works [15–18], which had fixed class centers. The firefly algorithm (FA), which is a recently developed nature-inspired swarm-based algorithm on the behavior of social insects (fireflies) and the phenomenon of bioluminescent communication [40], is used to solve the optimization problem. Preliminary studies on solving multimodal optimization problems indicate that the FA is superior to GA and PSO [41]. The FA provides a variation in the position of both positive and negative class centers by changing their attributes' values. Hence, adjusting the place of class centers can properly generate accurate fuzzy memberships to cope with attribute and class noises. Moreover, the FA, through the process of generating fuzzy memberships, chooses the best parameters for the SVM family. A set of experiments is conducted on nine benchmarking data sets of the UCI database, and the results are compared with the seven well-known methods for the SVM family in the literature.

This paper is organized as follows. The basic SVM, FSVM, LSSVM, and FLSSVM formulations for binary classification are reviewed in Section 2. In Section 3, the FA as an optimization method is introduced for realizing the proposed approach. In Section 4.1, the model selection problem is summarized, and the details of the new fuzzy membership function assignment for the SVM family are discussed in Section 4.2. In Section 5, experimental results are presented and discussed to illustrate the effectiveness of the proposed method. Concluding remarks are given in Section 6.

2. Review of the formulations of the SVM family

The necessary mathematical formulation of SVM, FSVM, LSSVM, and FLSSVM for classification problems is reviewed in this section.

2.1. SVM

Assume that a two-class set Ω of labeled training points $(x_i y_i)$ is given. Each training point $x_i \in R^n$ belongs to either of the two classes, as determined by the corresponding label $y_i \in \{-1, 1\}$ for $i = 1, \ldots, n$. The optimal hyperplane is obtained by solving the quadratic optimization problem Eq. (1) (known as primal form), whose number of variables is as large as the training data size n.

$$\begin{aligned}
&Min\,\varphi(w,\xi) = \tfrac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \\
&st \\
&y_i \left(w^T .\, x_i + b\right) \geq 1 - \xi_i, \qquad i = 1, 2, \ldots, n \\
&\xi_i \geq 0, \qquad i = 1, 2, \ldots, n
\end{aligned} \tag{1}$$

Here, ξ_is are slack variables that represent the violation of the pattern separation condition. The user-defined parameter C is regarded as a regularization parameter that controls the model complexity. This is one of the model selection parameters in the SVM formulation. For nonlinear separable data, a kernel trick is utilized to map the input space into a high-dimensional space named feature space. The optimal hyperplane is obtained in the feature space. The primal optimal problem Eq. (1) can be transformed into dual form as follows.

$$\begin{aligned} &Max\, Q(\alpha) = \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j k(x_i, x_j) - \sum_{j=1}^{n}\alpha_j \\ &st \\ &\sum_{j=1}^{l}\alpha_i y_i = 0 \\ &0 \le \alpha_i \le C, \qquad i = 1, \ldots, n \end{aligned} \tag{2}$$

Here, $K(.,.)$ is a kernel function. In practical applications of SVMs, there are several frequent substitutions for selecting the kernel function $K(.,.)$. Some of the conventional kernel functions are listed in Table 1. In this table, σ and d are constants, and those parameters must be set by a user. For the MLP kernel, a suitable choice for β_0 and β_1 is needed to enable the kernel function to satisfy the Mercer condition [42,43]. Those kernel parameters have an enormous effect on generalization performance. Therefore, kernel parameters are selected as the other model selection parameters. Furthermore, in Eq. (2) $\alpha = (\alpha_1, \ldots, \alpha_n)$ is the vector of nonnegative Lagrange multipliers. The solution vector $\alpha = (\alpha_1, \ldots, \alpha_n)$ 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 correspond to nonzero α_i and are called support vectors. Therefore, the points x_i that correspond to $a_i = 0$ have no contribution to the construction of the optimal hyperplane, and only part of the training data and support vectors construct the optimal hyperplane. Letting v be the index set of support vectors, the optimal hyperplane is then:

Table 1. 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	$k(x_i, x_j) = exp?(-\lVert x_i - x_j\rVert^2/\sigma^2)$
MLP (*) kernel	$k(x_i, x_j) = tanh?(\beta_0 x_i^T x_j + \beta_1)$

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

and the resulting classifier is:

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

where b is easily determined by KKT conditions.

2.2. Fuzzy SVM

In many real-world applications, each data point is not fully classified into one of the two classes. Based on this fact, Lin extended the theory of classical SVM to FSVM [15]. In FSVM, each data point can make a different contribution to the construction of the optimal hyperplane in contrast to SVM, where all data points have 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 weights. Assume the training data in the following form:

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

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

$$\begin{aligned} &Min\,\varphi(w,\xi) = \tfrac{1}{2}w^T w + C\sum_{i=1}^{n} s_i\xi_i \\ &st \\ &y_i\left(w^T.\,x_i + b\right) \geq 1 - \xi_i, \qquad i = 1,2,\ldots,n \\ &\xi_i \geq 0, \qquad\qquad\qquad\quad i = 1,2,\ldots,n \end{aligned} \tag{6}$$

Here, C is the regularization parameter. The main difference between the SVM and FSVM arises from the $s_i\xi_i$ term. Since ξ_i is known as a measure of error in the SVM, the fuzzy membership s_i has different weights for error measurement 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 the FSVM is obtained. Additionally, the FSVM can be solved by its dual form. This issue was discussed in more detail in [15,16].

2.3. Least square SVM and FLSSVM

The least square SVM (LSSVM) was first proposed by Suykens et al. by modifying the formulation of the standard SVM [44]. The LSSVM was modified in two points: first, instead of inequality constraints, it took equality constraints and changed the quadratic programming to a linear programming. Second, a squared loss function was taken from the error variable [44,45]. These modifications greatly simplified the problem and can be described in detail as follows.

Consider a given training set of n data points $\{(x_i y_i)|i = 1,2,\ldots,n,\ x_i \in R^n, y_k \in R$, where x_i is the ith input pattern and y_i is the ith output pattern. The nonlinear classification function modeling takes the form in Eq. (7):

$$y(x) = w^T\varphi(x) + b \tag{7}$$

where $\varphi(x)$ denotes the high-dimensional feature space, w is the weight vector, and b is the bias term. Then the problem is resolved to minimize the empirical risk cost function in Eq. (8).

$$\begin{aligned} &\min J(w,e) = \tfrac{1}{2}w^T w + \gamma\tfrac{1}{2}\sum_{i=1}^{n} e_i^2 \\ &st \\ &y_i = w^T\varphi(x_i) + b + e_i\,,\ i = 1,\ldots,n \end{aligned} \tag{8}$$

Here, e_i are error variables that play a similar role as the slack variables ξ_i in the Vapnik SVM formulation, and γ is a regularization parameter in determining the trade-off between minimizing the training errors and minimizing the model complexity. The Lagrangian corresponding to Eq. (8) can be defined as:

$$L(w,b,e,\alpha) = J(w,e) - \sum_{i=1}^{n} \alpha_i \{w^T \varphi(x_i) + b + e_i - y_i \tag{9}$$

where $\alpha_i \in R$ are the Lagrange multipliers. The KKT optimality conditions for a solution can be obtained by partially differentiating with respect to w, b, e_i, and α_i.

$$\begin{cases} \frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{n} a_i \varphi(x_i) \\ \frac{\partial L}{\partial w} = 0 \rightarrow -\sum_{i=1}^{n} a_i y_i = 0 \\ \frac{\partial L}{\partial e_i} = 0 \rightarrow a_i = \gamma e_i, \qquad i = 1,\dots,n \\ \frac{\partial L}{\partial \alpha_i} = 0 \rightarrow w^T \varphi(x_i) + b + e_i - y_i = 0, \end{cases} \tag{10}$$

After elimination of the variable w and e_i, the following linear equation can be obtained:

$$\begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 & \vec{1}_n^T \\ \vec{1}_n & \Omega + \gamma^{-1} I_n \end{bmatrix} \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{11}$$

where $y = [y_1; \dots; y_n]$, $\vec{1}_n = [1; \dots; 1]$, and $\alpha = [\alpha_1, \dots, a_n]$. The kernel trick is applied here as follows:

$$\begin{aligned} \Omega_{ij} &= \varphi(x_i)^T \varphi(x_j) \\ &= K(x_i x_j), ij = 1,\dots,n \end{aligned} \tag{12}$$

where $K(\cdot,\cdot)$ is the kernel function fulfilling the Mercer condition. Similarly, b and α are obtained by the following.

$$b = \frac{\vec{1}_n^T (\Omega + \gamma^{-1} I_n)^{-1} y}{\vec{1}_n^T (\Omega + \gamma^{-1} I_n)^{-1} \vec{1}_n^T} \tag{13}$$

$$\alpha = \left(\Omega + \gamma^{-1} I_n\right)^{-1} (y - \vec{1}_n^T b) \tag{14}$$

Finally, the resulting LSSVM model for the classification problem can be expressed as follows.

$$y(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x_j) + b \tag{15}$$

2.4. Fuzzy LSSVM

A fuzzy extension of LSSVM for classification problems is formulated as follows.

$$\begin{aligned} &\min J_f(w,e) = \tfrac{1}{2} w^T w + \gamma \tfrac{1}{2} \sum_{i=1}^{n} s_i e_i^2 \\ &st \\ &y_i = w^T \varphi(x_i) + b + e_i, \ i = 1,\dots,n \end{aligned} \tag{16}$$

The Lagrangian function is

$$L\left(w,b,e,\alpha\right)=J_f\left(w,e\right)-\sum_{i=1}^{n}\alpha_i\{w^T\varphi\left(x_i\right)+\ b+e_i-y_i \tag{17}$$

where $\alpha_i \in R$ are the Lagrange multipliers and could have either positive or negative values due to the equality constraints. The optimal conditions are obtained by differentiating Eq. (17).

$$\begin{cases} \frac{\partial L}{\partial w}=0\rightarrow w=\sum\limits_{i=1}^{n}a_i\varphi\left(x_i\right) \\ \frac{\partial L}{\partial w}=0\rightarrow -\sum\limits_{i=1}^{n}a_iy_i=0 \\ \frac{\partial L}{\partial e_i}=0\rightarrow a_i=\gamma s_ie_i\,, \qquad i=1,\ldots,n \\ \frac{\partial L}{\partial \alpha_i}=0\rightarrow w^T\varphi\left(x_i\right)+b+e_i-y_i=0, \end{cases} \tag{18}$$

Similar to the normal LSSVM, the matrix formulation can be obtained as follows:

$$\begin{bmatrix} b \\ \alpha \end{bmatrix}=\begin{bmatrix} 0 & \vec{1}_n^T \\ \vec{1}_n & \Omega+\gamma^{-1}I_n \end{bmatrix}\begin{bmatrix} 0 \\ y \end{bmatrix}, \tag{19}$$

where $y=[y_1;\ldots;y_n]$, $\vec{1}_n=[1;\ldots;1]$, and $\alpha=[\alpha_1,\ldots,a_n]$. The kernel trick is applied here as follows:

$$\Omega_{ij}=y_iy_j\varphi\left(x_i\right)^T\varphi\left(x_j\right)+\left(s_i\gamma\right)^{-1}I \tag{20}$$

$$=K\left(x_i,x_j\right),i\,,\,j=1,\ldots,n$$

where $K(\cdot,\cdot)$ is the kernel function satisfying Mercer's condition. The resulting FLSSVM model and its parameters, b and α, are similarly obtained by the normal LSSVM formulation. For further details, see [44,46,47].

3. Firefly algorithm

The FA is a population-based algorithm proposed by Yang [40,41]. It is based on the following three idealized rules:

- All fireflies are unisex, so that a firefly is attracted to other fireflies regardless of their sex.
- Attractiveness is proportional to their brightness or light intensity; thus, for any two flashing fireflies, the less bright one will move towards the brighter one. Attractiveness is proportional to brightness and they both decrease as their distance increases. If no firefly is brighter than any other, then it moves randomly.
- The brightness of a firefly is affected or determined by the landscape of the objective function to be optimized.

Initially, all the fireflies are randomly dispersed across the search space. The FAs can be summarized simply in the two following stages:

1) Variation of light intensity: Light intensity depends on the value of the objective function. Therefore, it can be suggested that in the maximization/minimization problem, a firefly with high/low intensity will attract another firefly with high/low intensity.

For all the fireflies, x_i brightness I_i corresponds to its value of the objective function and is represented as follows [48]:

$$I_i = f(x_i)\ i = 1, 2, \ldots, n. \tag{21}$$

2) Movement toward a brighter firefly: The movement of a firefly x_i is absorbed into another brighter (more attractive) firefly x_j and is formulated by:

$$x_i^{t+1} = x_i^t + \beta(r) \times \left(x_i^t - x_j^t\right) + \alpha\varepsilon_i^t \tag{22}$$

where $\beta(r)$ is the attractiveness function of the firefly and is determined by:

$$\beta(r) = \beta_0 e^{-\gamma r_{ij}^2} \tag{23}$$

where β_0 is the attractiveness at $r = 0$ and γ is the light absorption coefficient. r represents the distance between any two fireflies i and j at x_i, and x_j can be formulated by any norm, such as the l_2-norm, as follows:

$$r_{ij} = \|x_i - x_j\|_2 \tag{24}$$

Finally, the third term is randomization, with the vector of random variables ε_i being drawn from a Gaussian distribution. In essence, the parameter γ characterizes the variation of the attractiveness and partly controls how the algorithm behaves. It is also possible to adjust γ so that multiple optima can be found at the same during iterations [49]. A meticulous detailed description of the FA was given in [41]. All parameters required to implement the FA are presented in the Appendix. The pseudocode of this algorithm is given below.

Algorithm 1. Firefly algorithm.

Objective function f(x), $x = (x_1, \ldots, x_d)^T$
Initialize a population of fireflies $x_i(i = 1, 2, \ldots, n)$. Define light absorption coefficient γ
while (t < MaxGeneration)
for i = 1: n all n fireflies
  for j = 1: i all n fireflies
    Light intensity I_i at x_i is determined by $f(x_i)$
    if $(I_j > I_i)$
      Move firefly i toward j in all d dimensions
    end if
    Attractiveness varies with distance r via $\exp[-\gamma r^2]$
    Evaluate new solutions and update light intensity
  end for j
end for i
Rank the fireflies and find the current best
end while
Postprocess results and visualization.

4. New fuzzy membership assignment and model selection problem

4.1. Model selection problem

The optimal parameter selection of the SVM family is an important step in achieving a high generalization performance. As mentioned earlier, the SVM family has two adjustable parameters known as the regularization parameter and kernel parameter(s). The kernel parameter(s) implicitly characterize the geometric structure of data in the high-dimensional space known as feature space. In the feature space, the data become linearly separable in such a way that the maximal margin of separation between two classes is achieved. The selection of kernel parameter(s) will change the shape of the separating surface in the input space. Selecting improperly large or small values for kernel parameter(s) is the cause of overfitting or underfitting in the model surface of the SVM family. Consequently, the model would be unable to accurately separate data [28,42,50].

In nonseparable problems, noisy training data will introduce slack variables to measure their violation of the margin. Therefore, a penalty factor, named C in SVM and γ in LSSVM, is considered for controlling the amount of margin violation. In other words, the penalty factor is defined to determine the trade-off between minimizing empirical error and structural risk and guaranteeing the accuracy of the classifier outcome in the presence of noisy training data. A higher penalty factor value causes the margin to be hard and the cost of violation to become too high, and hence the separating model surface overfits the training data. By contrast, lower penalty factor values allow the margin to be soft, which results in underfitting separating the model surface. In both cases, the generalization performance of the classifier is unsatisfactory, and this renders the SVM family's model useless [28,51].

4.2. New fuzzy membership assignments

How to determine appropriate fuzzy memberships for use in the FSVM and FLSSVM 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 the fuzzy memberships function. As a result, the FSVM and FLSSVM can achieve good performance and discard the effects of noise and outliers if the fuzzy membership functions prepare/provide 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 class center [15–18]. Many fuzzy membership functions have been proposed based on this idea. Although those methods could cope with outliers or misclassification noise in the classification problem, they were not capable of dealing with attribute noise accurately.

To overcome this problem, a new fuzzy membership function based on dynamic class centers is proposed so that the performance of the FSVM and FLSSVM is enhanced in both misclassification and attribute noise. In this approach, determining the appropriate position of class centers is defined as an optimization problem. Whereas the fuzzy membership functions are generated based on the position of their class centers, the optimal place of the class centers will improve the performance of the FSVM family. Each data set is divided into two positive and negative classes. Each class center is a mean vector of positive and negative class-label data points. We can define the mean of the positive class label as x_+ and the mean of the negative class as x_- as follows:

$$x_+ = \frac{1}{n_+} \sum_{x_i \in C^+} x_i \tag{25}$$

and

$$x_- = \frac{1}{n_-} \sum_{x_i \in C^-} x_i \tag{26}$$

where n_+ and n_- are the number of data points in class C^+ and C^-, respectively. In previous studies suggesting new fuzzy memberships, researchers focused on extending the class center-based fuzzy membership assignment in which the position of the class centers were fixed. Thus, the noise data and outliers are discarded and eliminated by their distance from their corresponding class centers. It is obvious that the class center of a noisy data set will be noisy and not adequately pure to use as a reference measure for producing fuzzy memberships. In this study, we propose to vary the position of the class centers by using FA to denoise class center attributes. Moreover, by modifying the position of the class centers, their data points within each class could have different weights, relevant to their class center. For this aim, we considered a lower and upper bound for class center attributes, and the FA was used 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 firefly is shown in Table 2.

Table 2. Structure of each firefly.

Par. n + 2		Par. m + 1	Par. m		Par. 3	Par. 2	Par. 1	
$MeanNeg_{in+2} - \alpha$	$\cdots$	$MeanNeg_{im+1} - \alpha$	$MeanPos_{im} - \alpha$	$\cdots$	$MeanPos_{i3} - \alpha$	log (0.001)	log (0.01)	Min. value
x_{in+2}	$\cdots$	x_{im+1}	x_{im}	$\cdots$	x_{i3}	x_{i2}	x_{i1}	
$MeanNeg_{in+2} + \alpha$	$\cdots$	$MeanNeg_{im+1} + \alpha$	$MeanPos_{im} + \alpha$	$\cdots$	$MeanPos_{i3} + \alpha$	log (150)	log (1500)	Max. value
Negative class center			Positive class center			σ	C or γ	

As shown in Table 2, the first two dimensions of each firefly are considered for solving the model selection problem in the SVM family, and the remaining dimension of each firefly is used for finding the optimal mean attribute in each dimension of class centers. The logarithmic range is considered for finding the regularization parameter and the kernel parameter in the model selection problem. Logarithmic range is used because this gives a more stable solution to the optimization problem. This range is log(0.01) to log(1500) for the regularization parameter and log(0.001) to log(150) for the kernel parameter. First, the mean of each positive and negative class center is calculated, and then a small positive constant, the so-called α, is added or subtracted from these means to obtain the upper and lower bounds of each feature in positive and negative class centers. In this study, the value of α is 0.1.

We define the maximum radius of positive class C^+ by:

$$r_+ = max\,|x_+ - x_i| \;\, for\, x_i \in C^+ \tag{27}$$

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

$$r_- = max\,|x_- - x_i| \;\, for\, x_i \in C^- \tag{28}$$

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

$$s_i = \begin{cases} 1 - \|x_+ - x_i\| / (r_+ + \delta) & if\, y = 1 \\ 1 - \|x_- - x_i\| / (r_- + \delta), & if\, y = -1 \end{cases} \tag{29}$$

where δ is a small positive constant that is 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:

$$CostFunction = K\!-\!fold\,CV\ Error \tag{30}$$

The FA is used to determine the optimal class centers and simultaneously solve the model selection problem based on the cost function. Many researchers have proposed different criteria, such as the leave-one-out (LOO) [52] and K-fold CV methods, for evaluating the generalization performance of the SVM [53]. In the K-fold CV method the training data set is randomly divided into K equal subsets, and then for the K iteration each subset is defined as a testing data set. Retain subsets, K-1, are used as a training data set. After K iterations, the overall generalization performance is averaged over K calculated performances. Generally, each part of the training data set is separately considered as a testing data set. Therefore, K-fold CV is a robust criterion for evaluating generalization performance. Algorithm 2 describes the proposed method.

Algorithm 2. Proposed method.

1: Initialize FA population, i.e. initial model parameters and positive and negative class centers.
2: Generate fuzzy memberships based on initial class centers.
3: Train FSVM/FLSSVM with initial model parameters.
4: Evaluate the cost function.
5: Update the initial population based on the procedure of FA until the optimization approach terminates.
6: Use the best firefly of the FA containing the best model selection parameters, and the optimal positive and negative class centers where fuzzy memberships are generated based on them.

5. Computational experiments

5.1. Experimental conditions

To demonstrate the effectiveness of the proposed cost function, a PC with MATLAB R2008b software was utilized. Experiments on nine different real-world data sets, which are frequently used in the literature, were carried out to compare the performance of the proposed approach with seven well-known methods of the SVM family in the literature. The data set descriptions are presented in Table 3 [54]. In order to convert three-class data sets to two-class data sets, two classes of each data set that are not labeled +1 are merged.

Table 3. Description of data sets.

Data set name	# Data	# Features	# Classes
Banana	5300	2	2
Diabetes	768	8	2
German	1000	20	2
Image	2086	18	2
Ring norm	7400	20	2
Splice	2991	60	2
Thyroid	215	5	2
Two-norm	7400	20	3
Waveform	5000	21	3

Although the proposed method could be applied to all existing kernel functions, the RBF kernel function is used. The RBF kernel has a superior performance in comparison to the other kernels, as explained below.

The RBF kernel maps data sets nonlinearly into the feature space, and thus it can handle the data sets when the relation between desired output and input attributes is nonlinear. The second reason is the number of hyperparameters influencing the complexity of the model selection. The polynomial kernel has more hyperparameters than the RBF kernel. Finally, the RBF kernel has less numerical difficulties [25,28,42,43]. Consequently, the model selection parameters consist of the regularization parameter C in SVM and γ in LSSVM as well as the parameter σ, which is the only parameter of the RBF kernel. Moreover, the K value in K-fold CV is selected to be 100, as in [55] and [56]. For this purpose, each data set is split into 100 independent sample sets of training and test sets.

5.2. Experimental results and discussion

In order to emphasize the efficiency of the dynamic class center idea, FSVM-1 and FLSSVM-1 are developed. The fuzzy memberships of FSVM-1 and FLSSVM-1 are calculated based on the fixed class centers idea, proposed in [15], and the FA is only used to solve their model selection problems. The proposed approach is applied to the SVM and LSSVM to develop FSVM-2 and FLSSVM-2, respectively.

Table 4 shows the comparison results for a single RBF classifier (RBF), AdaBoost (AB), and regularized AdaBoost (ABR), which are obtained from Rätsch et al. [55]; the results for LOO-SVM, obtained from Weston and Herbrich [56]; the results for FSVM using kernel target alignment (KT) strategy; FSVM using the k nearest neighbors (k-NN) strategy, obtained from [16]; the results for FSVM-1 and FLSSVM-1 using only the FA for solving their model selection problems; and FSVM-2 and FLSSVM-2 using the proposed approach.

Table 4. Comparison of test error of seven well-known methods in the literature on 9 benchmarking data sets. The best results are shown in bold and the second best are underlined.

Data set	RBF	AB	AB_R	SVMs	LOOS	KT	k-NN	FA-FSVM-1	FA-FSVM-2	FA-FLSSVM-1	FA-FLSSVM-2
Banana	10.8	12.3	10.9	11.5	10.6	10.4	11.4	10.7	**9.5**	11.1	<u>9.6</u>
Diabetes	24.3	26.5	23.8	23.5	23.4	23.3	23.5	23.6	**22.7**	23.6	<u>23.0</u>
German	24.7	27.5	24.3	23.6	N/A	<u>23.3</u>	23.6	23.9	**23.2**	24.0	<u>23.3</u>
Image	3.3	2.7	2.7	3.0	N/A	2.9	3.0	3.2	<u>2.6</u>	3.1	**2.5**
Ring norm	1.7	1.9	1.6	1.7	N/A	1.7	1.7	1.7	**1.4**	1.8	<u>1.5</u>
Splice	10.0	10.1	9.5	10.9	N/A	10.9	10.9	10.9	**8.9**	10.8	<u>9.0</u>
Thyroid	4.5	4.4	4.6	4.8	5.0	4.7	4.8	4.7	**1.8**	4.8	<u>2.1</u>
Two-norm	2.9	3.0	2.7	3.0	N/A	2.4	2.9	2.5	<u>2.3</u>	2.4	**2.2**
Waveform	10.7	10.8	9.8	9.9	N/A	9.9	9.9	9.8	**8.4**	9.9	<u>8.7</u>

The results showed that the proposed method outperformed the smaller generalization error in the data set with noise. In contrast to the k-NN and KT strategies, there is no need for additional information about noise in data sets or the distribution of data sets to show which strategy should be used in each case. Moreover, the results demonstrate that our method can considerably improve the performance of SVM and LSSVM when the data sets contain class noise or even attribute noise. Figures 1 and 2 show the variation attributes of the positive and negative class centers in FSVM-2 and FLSSVM-2 for both fixed and dynamic class centers' ideas. Both figures consist of nine sections for the entire data set of experiment, and each section is divided into four subplots. In both figures, right-upper and left-upper subplots show the fixed positive and negative class centers, and right-lower and left-lower subplots illustrate the positive and negative class centers after applying the proposed method. The optimal cost functions are obtained in FSVM-2 and FLSSVM-2, based on the different class centers in comparison to fixed class centers. As shown in these figures, the proposed method properly denoises the new positive and negative class centers from attribute noise. Furthermore, the fuzzy memberships, which are generated based on the new class centers, could appropriately discard the boundary data points from class noise. Moreover, Figure 3 illustrates the positive and negative class center position in the Banana data set. Figures 3a and 3b display the position of the fixed class centers and the dynamic class center, which were obtained from the proposed method in FSVM-2 and FLSSVM-2, respectively. In these figures, positive and negative class-label data points are shown in blue and red colors, respectively. Both fixed class centers of positive and negative data points are presented with blue- and red-filled squares, respectively. The blue- and red-filled circles show the new class centers for positive and negative data points, respectively. On the other hand, fuzzy memberships are assigned to data points regarding their distance to the corresponding class centers. The movements of class centers play an important role in this matter. The optimal solution for the model selection problem using the FA for FSVM-1, FLSSVM-1, FSVM-2, and FLSSVM-2 is shown in Table 5.

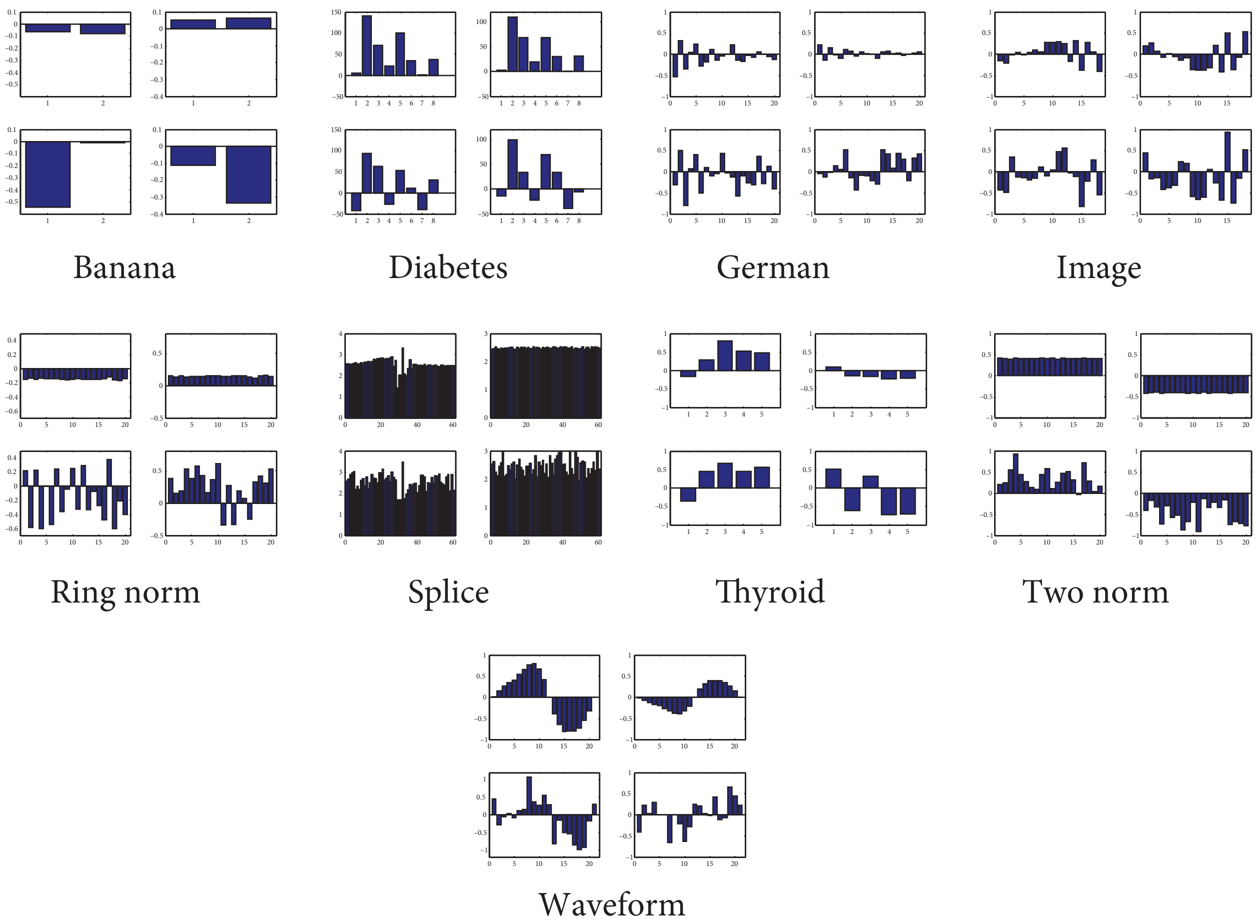

Figure 1. Comparison between the variation of the fixed positive class center (left-upper column) and the new positive class center after applying the proposed approach (left-lower column), and the fixed negative class center (right-upper column) and the new negative class center after applying the proposed approach (right-lower column) for FSVM-2 in nine data sets.

Table 5. Solution of model selection problem for all the experiments.

Data set name	FSVM-1		FLSSVM-1		FSVM-2		FLSSVM-2	
	log(C)	log(σ)	Log(γ)	Log(σ)	Log(C)	Log(σ)	Log(γ)	Log(σ)
Banana	–0.4685	0.3016	0.2917	–0.4963	3.9764	–0.0082	2.5396	–0.8400
Diabetes	–0.6649	1.4401	5.3623	4.7953	4.1815	4.4852	–3.8401	3.9511
German	0.5650	0.8735	–0.6563	1.9206	2.9125	2.5629	–0.3235	1.8367
Image	3.6401	–0.0599	4.5441	1.5783	7.2264	1.1537	3.2086	0.4181
Ring norm	0.3278	–0.4002	–0.8223	1.3019	–0.4791	0.9915	–0.7487	1.1872
Splice	3.0938	1.4657	0.7980	3.5811	5.1435	1.4551	4.6986	3.0734
Thyroid	4.6921	1.0884	0.4293	0.5250	5.7720	1.3054	–0.7317	0.4040
Two norm	1.0193	2.2679	–0.6090	3.9299	5.2730	3.9446	–3.6330	4.4739
Waveform	–0.2541	0.8417	0.2338	4.3971	–0.1392	0.9313	3.0969	5.0031

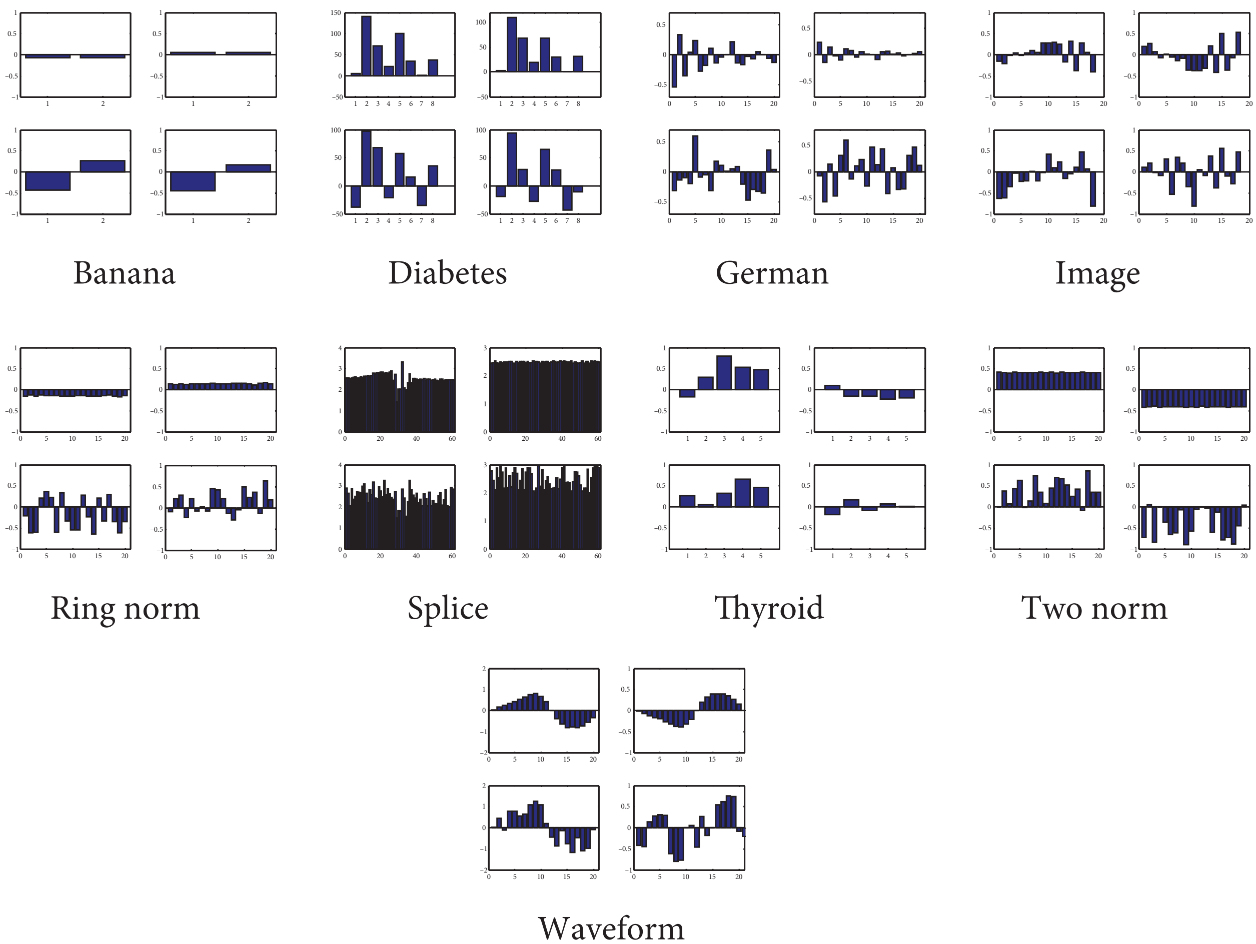

Figure 2. Comparison between the variation of the fixed positive class center (left-upper column) and the new positive class center after applying the proposed approach (left-lower column), and the fixed negative class center (right-upper column) and the new negative class center from the proposed approach (right-lower column) for FLSSVM-2 in nine data sets.

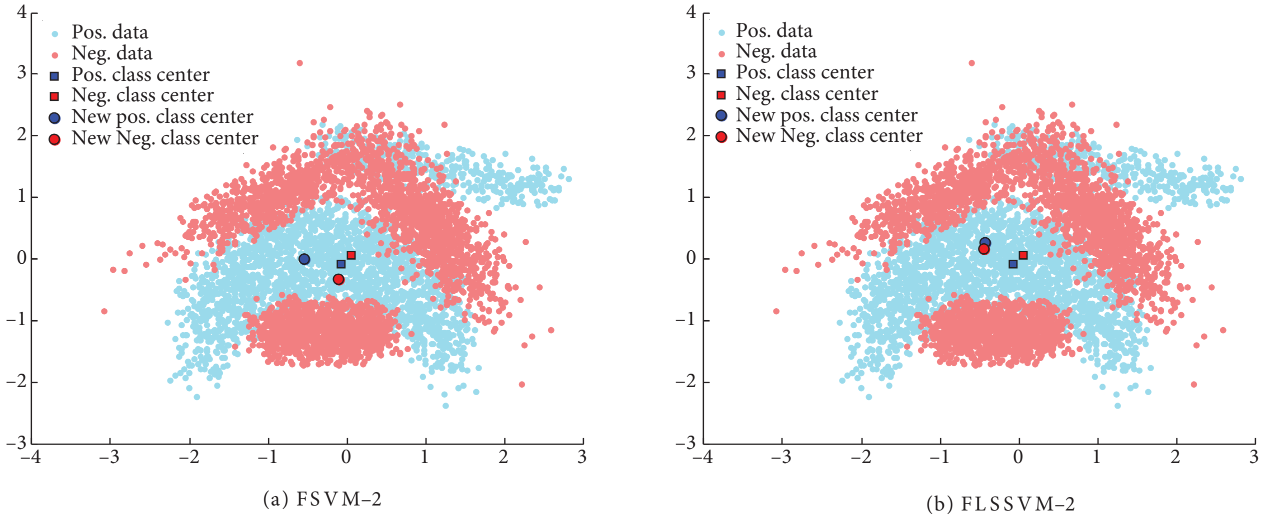

Figure 3. The positive and negative class center position in the Banana data set: a) FSVM-2, b) FLSSVM-2.

6. Conclusion

In the FSVM family, different training points can make different contributions to the learning of the decision surface. Both choosing proper fuzzy memberships and selecting the optimal parameters play a significant role in the SVM family. This paper proposed a new method based on the FA for assigning appropriate fuzzy memberships and solving the model selection problem for linear/nonlinear and separable/nonseparable classification problems.

The results demonstrate that by using this fuzzy membership assignment and model selection simultaneously, the effect of the attribute noise and outlier data can be considerably reduced. In addition to the FSVM and FLSSVM, it was established that this method can achieve higher generalization performance in comparison to other methods in previous works. It makes FSVM and FLSSVM more feasible for real applications, containing noise in the attribute and class label of the data points, although not enough information exists to choose appropriate fuzzy memberships and their optimal parameters.

Further work is necessary on the extension of the proposed approach to feature space, as well as on the optimization method for feature/data reduction for large noisy data sets.

References

[1] Guyon I, Boser B, Vapnik V. Automatic capacity tuning of very large VC-dimension classifier. Adv Neur In 1993; 5: 147-155.

[2] Vapnik VN. Statistical Learning Theory. New York, NY, USA: Wiley, 1998.

[3] Vapnik VN. An overview of statistical learning theory. IEEE T Neural Networ 1999; 10: 988-999.

[4] Burges CJC. A tutorial on support vector machines for pattern recognition. Data Min Knowl Disc 1998; 2: 955-974.

[5] Zhang XG. Using class-center vectors to build support vector machines. In: IEEE 1999 Workshop on Signal Processing Society; August 1999; Madison, MI, USA. New York, NY, USA: IEEE. pp. 3-11.

[6] Xiong H, Pandey G, Steinbach M, Kumar V. Enhancing data analysis with noise removal. IEEE T Knowl Data Eng 2006; 18: 304-319.

[7] Hulse JV, Khoshgoftaar TM. Knowledge discovery from imbalanced and noisy data. Data Knowl Eng 2009; 68: 1513-1542.

[8] Mavroforakis ME, Theodoridis S. A geometric approach to support vector machine (SVM) classification. IEEE T Neural Networ 2006; 17: 671-682.

[9] Angelova A, Abu-Mostafa Y, Perona P. Pruning training sets for learning of object categories. In: IEEE 2005 Conference on Computer Vision and Pattern Recognition; 20–25 June 2005; San Diego, CA, USA. New York, NY, USA: IEEE. pp. 494-501.

[10] Zhu X, Wu X. Class noise vs. attribute noise: a quantitative study of their impacts. Artif Intell Rev 2004; 22: 177-210.

[11] Brodley CE, Friedl MA. Identifying mislabeled training data. J Artif Intell Res 1999; 11: 131-167.

[12] Hulse JV, Khoshgoftaar TM, Huang H. The pairwise attribute noise detection algorithm. Knowl Inf Syst 2007; 11: 171-190.

[13] Khoshgoftaar TM, Zhong S, Joshi V. Enhancing software quality estimation using ensemble-classifier based noise filtering. Intell Data Anal 2005; 9: 3-27.

[14] Wu X, Zhu X. Mining with noise knowledge: error-aware data mining. IEEE T Syst Man Cy 2008; 38: 917-932.

[15] Lin CF, Wang SD. Fuzzy support vector machines. IEEE T Neural Networ 2002; 13: 464-471.

[16] Lin CF, Wang SD. Training algorithms for fuzzy support vector machines with noisy data. Pattern Recogn Lett 2004; 25: 1647-1656.

[17] Jiang XF, Zhang Y, Lv JC. Fuzzy SVM with a new fuzzy membership function. Neural Comput Appl 2006; 15: 268-276.

[18] Tang WM. Fuzzy SVM with a new fuzzy membership function to solve the two-class problems. Neural Process Lett 2011; 34: 209-219.

[19] Peng X, Wang Y. A geometric method for model selection in support vector machine. Expert Syst Appl 2009; 36: 5745-5749.

[20] Wang S, Meng B. Parameter selection algorithm for support vector machine. Procedia Environ Sci 2011; 11: 538-544.

[21] Chapelle O, Vapnik VN, Bousquet O, Mukherjee S. Choosing multiple parameters for support vector machines. Mach Learn 2002; 46: 131-159.

[22] Opper M, Winther O. Gaussian processes and SVM: mean field and leave-one-out. In: Smola AJ, Bartlett PL, Scholkopf B, Schuurmans D, editors. Advances in Large Margin Classifiers. Cambridge, MA, USA: MIT Press, 2000. pp. 311-326.

[23] Vapnik V, Chapelle O. Bounds on error expectation for support vector machines. Neural Comput 2000; 12: 2013-2036.

[24] Keerthi SS. Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms. IEEE T Neural Networ 2002; 13: 1225-1229.

[25] Sun J, Zheng C, Li X, Zhou Y. Analysis of the distance between two classes for tuning SVM hyperparameters. IEEE T Neural Networ 2010; 21: 305-318.

[26] Guo XC, Yang JH, Wu CG, Wang CY, Liang YC. A novel LS-SVMs hyper-parameter selection based on particle swarm optimization. Neurocomputing 2008; 71: 3211-3215.

[27] Glasmachers T, Igel C. Gradient-based adaptation of general Gaussian kernels. Neural Comput 2005; 17: 2099-2105.

[28] Keerthi SS, Lin CJ. Asymptotic behavior of support vector machines with Gaussian kernel. Neural Comput 2003; 15: 1667-1689.

[29] Wang S, Meng B. PSO algorithm for support vector machine. In: IEEE 2010 Conference on Electronic Commerce and Security; 29–31 July 2010; Guangzhou, Hong Kong. New York, NY, USA: IEEE. pp. 163-167.

[30] Lei P, Yi L. Parameter selection of support vector machine using an improved PSO algorithm. In: IEEE 2010 2nd International Conference on Intelligent Human–Machine Systems and Cybernetics; 26–28 August 2010; Nanjing, China. New York, NY, USA: IEEE. pp. 221-225.

[31] Lin SW, Ying KC, Chen SC, Lee ZJ. Particle swarm optimization for parameter determination and feature selection of support vector machines. Expert Syst Appl 2008; 35: 1817-1824.

[32] Cheng W, Ding J, Kong W, Chai T, Qin SJ. An adaptive chaotic PSO for parameter optimization and feature extraction of LS-SVM based modeling. In: 2011 American Control Conference; 29 June–1 July 2011; San Francisco, CA, USA. New York, NY, USA: IEEE. pp. 3263-3268.

[33] Luo Z, Zhang W, Li Y, Xiang M. SVM parameters tuning with quantum particles swarm optimization. In: IEEE 2008 Conference on Cybernetics and Intelligent Systems; 21–24 September 2008; Chengdu, China. New York, NY, USA: IEEE. pp. 324-329.

[34] Zhang W, Niu P. LS-SVM based on chaotic particle swarm optimization with simulated annealing and application. In: IEEE 2011 2nd International Conference on Intelligent Control and Information Processing; 25–28 July 2011; Harbin, China. New York, NY, USA: IEEE. pp. 931-935.

[35] Blondin J, Saad A. Metaheuristic techniques for support vector machine model selection. In: IEEE 2010 10th International Conference on Hybrid Intelligent Systems; 23–25 August 2010; Atlanta, GA, USA. New York, NY, USA: IEEE. pp. 197-200.

[36] Wu CH, Tzeng GH, Goo YJ, Fang WC. A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy. Expert Syst Appl 2007; 32: 397-408.

[37] Frohlich H, Chapelle O, Scholkopf B. Feature selection for support vector machines by means of genetic algorithms. In: IEEE 2003 15th International Conference on Tools with Artificial Intelligence; 3–5 November 2013; Sacramento, CA, USA. New York, NY, USA: IEEE. pp. 142-148.

[38] Huang CL, Wang CJ. A GA-based feature selection and parameters optimization for support vector machines. Expert Syst Appl 2006; 31: 231-240.

[39] Lihu A, Holban Ş. Real-valued genetic algorithms with disagreements. In: Pelta D, Krasnogor N, Dumitrescu D, Chira C, Lung R, editors. Nature Inspired Cooperative Strategies for Optimization. Berlin, Germany: Springer, 2012. pp. 333-346.

[40] Yang XS. Nature-Inspired Metaheuristic Algorithms. 1st ed. Bristol, UK: Luniver Press, 2008.

[41] Yang XS. Firefly algorithms for multimodal optimization. In: Watanabe O, Zeugmann T, editors. Stochastic Algorithms: Foundations and Applications. Berlin, Germany: Springer-Verlag, 2009. pp. 169-178.

[42] Lin HT, Lin CJ. A Study on Sigmoid Kernels for SVM and the Training of Non-PSD Kernels by SMO-Type Methods. Taipei, Taiwan: Taiwan University, 2003.

[43] Bordes A, Ertekin S, Weston J, Bottou L. Fast kernel classifiers with online and active learning. J Mach Learn Res 2005; 6: 1579-1619.

[44] Suykens JAK, Gestel TV, De Brabanter J, De Moor B, Vandewalle J. Least Squares Support Vector Machines. Singapore: World Scientific Publishing, 2002.

[45] Suykens JAK, Vandewalle J, De Moor B. Optimal control by least squares support vector machines. Neural Networks 2001; 14: 23-35.

[46] Chuang CC. Fuzzy weighted support vector regression with a fuzzy partition. IEEE T Syst Man Cy 2007; 37: 630-640.

[47] Yu L, Lai K, Wang S, Zhou L. A least squares fuzzy SVM approach to credit risk assessment. In: Cao B, editor. Fuzzy Information and Engineering. Berlin, Germany: Springer-Verlag, 2007. pp. 865-874.

[48] Senthilnath J, Omkar SN, Mani V. Clustering using firefly algorithm: performance study. Swarm Evol Comput 2011; 1: 164-171.

[49] Yang XS, Hosseini SS, Gandomi AH. Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. Appl Soft Comput 2011; 12: 1180-1186.

[50] Williams P, Li S, Feng J, Wu S. A geometrical method to improve performance of the support vector machine. IEEE T Neural Networ 2007; 18: 942-947.

[51] Ding S, Liu X. Evolutionary computing optimization for parameter determination and feature selection of support vector machines. In: IEEE 2009 Conference on Computational Intelligence and Software Engineering; 11–13 December 2009; Wuhan, China. New York, NY, USA: IEEE. pp. 1-5.

[52] Weston J. Leave-one-out support vector machines. In: Dean T, editor. Proceedings of the 16th International Joint Conference on Artificial Intelligence. San Francisco, CA, USA: Morgan Kaufmann, 1999. pp. 727–733.

[53] Browne MW. Cross-validation methods. J Math Psychol 2000; 44: 108-132.

[54] Murphy PM, Aha DW. UCI repository of machine learning databases. Irvine, CA, USA: UCI. Available online at www.ics.uci.edu/~mlearn/MLRepository.html.

[55] Ratsch G, Onoda T, Meuller K-R. Soft margins for AdaBoost. Mach Learn 2001; 42: 287-320.

[56] Weston J, Herbrich R. Adaptive margin support vector machines. In: Smola A, Bartlett P, Scholkopf B, Schuurmans D, editors. Advances in Large Margin Classifiers. Cambridge, MA, USA: MIT Press, 2000. pp. 281-295.

Appendix. FA algorithm parameters.

FA algorithm parameters

Swarm size	25
Max. iteration	40
β_0	1
γ	1
α	0.2
ε_i	$N \sim [-1, 1]$