

Fuzzy support vector machine based on hyperbolas optimized by the quantum-inspired gravitational search algorithm

Feng NI^{1,*}, Yuzhu HE¹, Fei JIANG²

¹Department of Instrument and Meter Engineering, Faculty of Instrumentation Science and Opto-electronics Engineering, Beihang University, Beijing, P.R. China

²PLA No. 5715 Factory, Luoyang, China

Received: 21.04.2016

Accepted/Published Online: 07.12.2016

Final Version: 30.07.2017

Abstract: Fuzzy support vector machines (FSVMs) are known for their excellent antinoise performance, but there is no general rule when the fuzzy membership function (FMF) is set up. A novel FSVM based on hyperbolas optimized by the quantum-inspired gravitational search algorithm (QGSH-FSVM) is proposed to handle this question. In the proposed QGSH-FSVM, the FMF is defined by two disparate hyperbolas, whose eccentricities are optimized by the quantum-inspired gravitational search algorithm. A variable called diversity, revealing the percentage of a sample in different classes, is proposed to distinguish outliers or noises from valid samples. Experimental results confirm that the QGSH-FSVM is able to provide the best solutions to different situations by optimizing its eccentricities. The traditional support vector machine and the FSVM based on affinity or the distance between a sample and its cluster center, however, can only succeed in some particular problems while failing in others.

Key words: Fuzzy support vector machine, fuzzy membership function, hyperbolas, eccentricities, diversity

1. Introduction

A support vector machine (SVM) is an effective machine learning method, powerful for problems characterized by small samples, nonlinearity, high dimensions, and local minima [1,2]. However, when the optimal classification plan is constructed, all the samples have the same effect. Since the noises and outliers are often located near the classification surface, the optimal classification plan usually is not the real one [3]. To make SVMs more robust, Lin et al. [4] presented the fuzzy SVM (FSVM), applying fuzzy technology to the SVM. The performance of FSVMs is critically dependent on the fuzzy membership function (FMF) [5].

At present, there are many kinds of methods constructing the FMF without a unified criterion. When dealing with actual data, we need to establish reasonable FMFs according to the experience for the specific problem. The authors in [6–8] introduced a design for FMFs based on the distance between a sample and its cluster center, but this method has to depend on the geometry of samples. The support vectors and noise will all be assigned small membership for their long distance from the cluster center. It can obtain ideal results for the samples distributed in the spherical area rather than the samples distributed along the planar area. To reduce the dependence on the geometry, the authors in [9] used the hyperplane of the cluster instead of its center. This method, however, fails in effectively distinguishing outliers from valid samples. In [10], the authors proposed a FSVM based on border vector extraction. They identified noises and outliers at first, and then they

*Correspondence: 13161105445@163.com

made the FMF increase with the distance of a sample to its cluster center. This method is able to enhance the importance of support vectors. However, the noises and outliers are difficult but crucial to identify, and a wrong decision will reduce the classification accuracy. The authors in [11] presented a fast FSVM. By removing the data that are considered as nonsupport vectors based on the geometric distribution of the dataset, it can improve the training efficiency more or less. However, the efficiency is at the expense of the training accuracy. In [12,13], the FMF was defined by not only the relation between a sample and its cluster center, but also those among samples. Before the FMF is worked out, a smallest hypersphere must be obtained. The memberships of samples outside the hypersphere are smaller than those inside it, but when the noises and outliers exclude the hypersphere, the support vectors are also ruled out, which decreases the classification precision. In addition, this algorithm relies on the samples' sparsity degree, and so it is not suitable for the situation of sparseness.

Aiming at providing a generally applicable method, we propose a novel FSVM based on hyperbolas optimized by the quantum-inspired gravitational search algorithm (QGSH-FSVM). In addition, we define diversity to make this algorithm more robust. This paper is organized as follows. In Section 2, the theory of FMF in FSVM is described. The proposed QGSH-FSVM is introduced in Section 3. The experimentation and the results obtained are given in Section 4. Finally, the paper is concluded in Section 5.

2. Theory of fuzzy membership function

In this section, according to [14], we briefly summarize the theory of FMF in FSVM. Compared with the traditional SVM, the FSVM adds a fuzzy membership to every training sample. Assume the dataset takes the form of:

$$(y_1, x_1, \mu(x_1)), (y_2, x_2, \mu(x_2)), \dots, (y_n, x_n, \mu(x_n)), \tag{1}$$

where $x_i \in R^d$ denotes the characteristic of a training point, $y_i \in \{+1, -1\}$ is the class label, and $0 < \mu(x_i) \leq 1$ is the FMF. The final optimal classification plan of the SVM is:

$$f(x) = \operatorname{sgn} \left\{ \sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i \cdot \mathbf{x}) + b \right\}, \tag{2}$$

$$s.t. 0 < \alpha_i \leq C$$

where C represents the error penalty, $\alpha_i \geq 0$ is a Lagrange multiplier, and $K(x_i, x_j)$ is the kernel function. The final optimal classification plan of the FSVM is:

$$f(x) = \operatorname{sgn} \left\{ \sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i \cdot \mathbf{x}) + b \right\}. \tag{3}$$

$$s.t. 0 < \alpha_i \leq \mu(\mathbf{x}_i)C$$

Comparing the results, we can find that the only difference is the constraint condition. In Eq. (2), α_i is constrained to the condition of $0 \leq \alpha_i \leq C$ and α^* is the optimal solution. If α^* meets the condition of $0 < \alpha^* \leq C$, the corresponding vector is a support vector, and if α^* meets the condition of $\alpha^* = 0$, the corresponding vector is not a support vector. The support vectors can be further divided into two kinds: boundary support vectors and general support vectors. The former are the samples near the classification line, meeting the condition of $0 < \alpha^* < C$. The latter are the inseparable samples, meeting the condition of $\alpha^* = C$. The classification plan of the SVM is only related to the boundary support vectors.

When C is given a large value, more samples meet the condition of $0 < \alpha^* < C$. That is to say, only a small part of the samples are wrongly classified. On the contrary, if C is given a small value, more samples will

meet the condition of $\alpha^* = C$. This means that more vectors can be misclassified and the classification risk will increase, resulting in more samples being overlooked by the SVM. All the samples are assigned the same C in the C-SVM, so the classes of all the support vectors will be adjusted if C is altered.

However, in Eq. (3), each sample of the FSVM can be assigned a different $\mu(\mathbf{x}_i)C$ according to its contribution to classification. If x_i is a support vector, it will be assigned a larger membership, meeting the condition of $0 < \alpha_i < \mu(\mathbf{x}_i)C$, but if x_i is an outlier or noise, a small membership will be allocated by using some membership function, reducing its impact on SVM training. In this way, the outliers or noises are effectively suppressed. Thus, the FMF is the key of the FSVM and it must be able to reflect the uncertainty of a sample in the system objectively and precisely.

3. Designing the fuzzy membership function

In order to put the FSVM to wide use in various conditions and improve its antinoise performance, we have to analyze the distribution characteristics of support vectors, noises, and outliers. This is because of the similar distribution characteristics of locating at the edge of the sample set such that the noises can decrease the classification accuracy. The samples distributed on the edge are not only close to their own class but are also close to the opposite class.

3.1. The disparate hyperbolas

The first step of this work is to obtain the class centers. We take the average value of two kinds of samples as the centers, which can reduce the dependence on the geometry. Assume that $\phi(\bullet)$ is the mapping function, N_+ is the number of positive samples, and N_- is the number of negative samples.

Definition 3.1: The class centers $m_i, i \in \{+, -\}$ are:

$$m_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \phi(x_j). \tag{4}$$

Then we define the straight line including m_+ and m_- as the x-axis. m_+ is defined as the right focus of positive hyperbola, and the straight line that is parallel to the y-axis and includes the left-most point is defined as the right hyperbolic line. According to the second definition of hyperbola, eccentricity e_+ is the distance from a point in the hyperbola to the focus point over the distance from the point to the hyperbolic line. Then we can get positive hyperbola $g(x)$. In a similar way, m_- is defined as the left focus of negative hyperbola, and the straight line that is parallel to the y-axis and includes the right-most point is defined as the left hyperbolic line. Then we can get negative hyperbola $h(x)$.

As shown in Figure 1, the two hyperbolas are disparate from each other with different eccentricities. By optimizing the eccentricities, the hyperbolas can achieve the best result of various kinds of samples. The FMF of inner points are decided by the minimum distance between them and the related hyperbola, and the outer points of hyperbolas are abandoned to save computing space.

3.2. The quantum-inspired gravitational search algorithm

To obtain the optimal $e_i, i \in \{+, -\}$, we use the quantum-inspired gravitational search algorithm (QIGSA) to optimize them. The QIGSA is a novel intelligent optimization algorithm [15], inspired by the classical gravitational search algorithm. In the quantum time-space framework, the quantum state of an individual

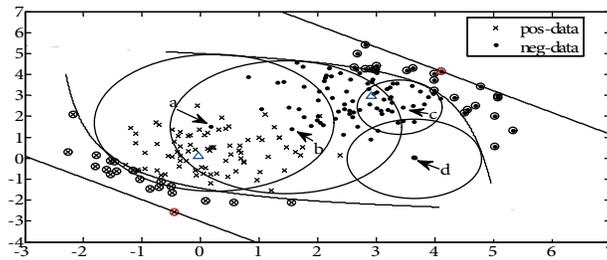


Figure 1. Situation of the hyperbolas.

is depicted by wave function (ψ) instead of position. According to [15], the procedure for implementing the QIGSA is given by the following steps:

Step 1. Randomly initialize a population in the D-dimensional problem space, where the position is defined as follows:

$$X_i = (x_i^1, x_i^2, \dots, x_i^k, \dots, x_i^d); i = 1, 2, \dots, N. \tag{5}$$

Step 2. Evaluate all agents' fitness.

Step 3. Select $kbest$ and update $pbest$. $kbest$, a function of time, is the set of first K agents with the best fitness value and biggest mass [16]. $pbest$ represents each agent's best fitness. In order to update it, we compare each agent's fitness with the agent's $pbest$. If the current value is better than $pbest$, then change the $pbest$ value equal to the current value and the $pbest$ location equal to the current location in D-dimensional space.

Step 4. Update $Mbest_i$ using Eq. (6):

$$Mbest_i = \frac{\sum_{j=1}^K \frac{1}{d_{i,j}} \cdot kbest_j}{\sum_{l=1}^K \frac{1}{d_{i,l}}}, \tag{6}$$

where $d_{i,l} = \|X_i - kbest_l\|$ represents the Euclidean distance between two masses.

Step 5. Update the masses' position. Change the position of the mass using Eq. (7), where c_1 and c_2 are two random numbers distributed in the range $[0, 1]$:

$$Best_i^d = \frac{c_1 Mbest_i^d + c_2 pbest_i^d}{c_1 + c_2}. \tag{7}$$

By using the Monte Carlo method, we get the iterative equation for the masses moving:

$$\begin{cases} X_i(t+1) = Best_i + \sigma \cdot |Best_i - X_i(t)| \cdot \ln\left(\frac{1}{rand}\right) & S \geq 0.5 \\ X_i(t+1) = Best_i - \sigma \cdot |Best_i - X_i(t)| \cdot \ln\left(\frac{1}{rand}\right) & S \leq 0.5 \end{cases}, \tag{8}$$

where σ is a design parameter called the contraction-expansion coefficient, and $rand$ and S are random values in the range $[0, 1]$.

Step 6. Loop Steps 2 to 5 until a stop criterion is reached.

3.3. The variable of diversity

In order to improve the QGSH-FSVM antinoise performance, we propose diversity η , revealing the percentage of a sample in different classes. The vectors x_i and x_j represent positive and negative samples and they are

mapped to the feature space Z by $\phi(\bullet)$. Then the Euclidean distance between the two vectors is:

$$d(x_i, x_j) = \sqrt{\phi(x_i, x_i) - 2\phi(x_i, x_j) + \phi(x_j, x_j)}, \quad (9)$$

and the Euclidean distance between x_i and the positive hyperbola's point x_g is:

$$d(x_i, x_g) = \sqrt{\phi(x_i, x_i) - 2\phi(x_i, x_g) + \phi(x_g, x_g)}. \quad (10)$$

The Euclidean distance between x_j and the negative hyperbola's point x_h is:

$$d(x_j, x_h) = \sqrt{\phi(x_j, x_j) - 2\phi(x_j, x_h) + \phi(x_h, x_h)}. \quad (11)$$

Definition 3.2: d_{\min}^+ is the minimum distance between a positive sample and the positive hyperbola. It can be obtained by solving:

$$\begin{cases} \min \sqrt{\phi(x_i, x_i) - 2\phi(x_i, x_g) + \phi(x_g, x_g)} \\ \text{s.t. } g(x_g) = 0 \end{cases} . \quad (12)$$

d_{\min}^- is the minimum distance between a negative sample and the negative hyperbola. It can be obtained by solving:

$$\begin{cases} \min \sqrt{\phi(x_j, x_j) - 2\phi(x_j, x_h) + \phi(x_h, x_h)} \\ \text{s.t. } h(x_h) = 0 \end{cases} . \quad (13)$$

For each sample inside the hyperbolas, it acts as the center of the sphere and its minimum distance d_{\min}^+ or d_{\min}^- acts as the radius. Then we can get a hypersphere. Inside the hypersphere, the number of positive samples is sum_+ and the number of negative samples is sum_- . We define diversity η as:

$$\eta_{\pm} = \begin{cases} \frac{sum_+}{N_+} + \frac{sum_-}{N_-} & \text{if } \frac{sum_+}{N_+} \geq \frac{sum_-}{N_-} \\ \frac{sum_+}{N_+} - \frac{sum_-}{N_-} & \text{if } \frac{sum_+}{N_+} < \frac{sum_-}{N_-} \\ \frac{sum_-}{N_-} + \frac{sum_+}{N_+} & \text{if } \frac{sum_-}{N_-} \geq \frac{sum_+}{N_+} \\ \frac{sum_-}{N_-} - \frac{sum_+}{N_+} & \text{if } \frac{sum_-}{N_-} < \frac{sum_+}{N_+} \end{cases} . \quad (14)$$

As shown in Figure 2, sum_-/N_- of noise point a is smaller than sum_+/N_+ , so η_- is a negative number. The η_- of boundary support vector point b is a large positive one because its sum_-/N_- and sum_+/N_+ are both large values and sum_-/N_- is larger than sum_+/N_+ . The sum_-/N_- of point c is very small and its sum_+/N_+ equals zero. Thus, η_- is a small positive number, and point d is most likely an outlier. Its sum_-/N_- is very, very small and its sum_+/N_+ is zero. Thus, the η_- of the outlier point is almost zero. In conclusion, we can see that η well reveals the sample's importance for classification.

3.4. The fuzzy membership function

According to the basic idea of the SVM, the sample closest to the classification hyperplane is the support vector, and the samples subclose to the classification hyperplane are also of importance. Fuzzy membership shows the importance of obtaining support vectors, so we should make the support vector's membership be close to 1.

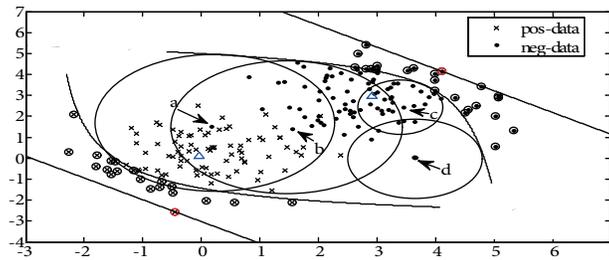


Figure 2. Theory of the affinity.

The smaller role a sample plays in solving classification, the smaller its membership is. In this way, the FMF is given as follows:

$$\mu(\mathbf{x}_i) = \begin{cases} \frac{d_{\min}^+ + 5 \times \eta_+}{\max(d_{\min}^+) + \max(5 \times \eta_+)} & y_i = +1 \\ \frac{d_{\min}^- + 5 \times \eta_-}{\max(d_{\min}^-) + \max(5 \times \eta_-)} & y_i = -1 \end{cases} \quad (15)$$

In conclusion, the procedure for implementing the QGSH-FSVM is given by the following steps:

Step 1. Work out the class centers by using Eq. (4).

Step 2. Optimize the parameters of $e_i, i \in \{+, -\}$.

- (a) Initialize a population of masses with random positions. Every mass represents a group of parameters (e_+, e_-) .
- (b) Solve the range of the masses' position iterations and step-size according to the characteristics of the hyperbola.
- (c) Evaluate the fitness value.
- (d) Select $kbest$ and update $pbest$.
- (e) Update the masses' position by using Eq. (6).
- (f) Loop to (b) until a stop criterion is met.

Step 3. Get the hyperbolas $g(x)$ and $h(x)$ according to the second definition, and map them to the space Z .

Step 4. Calculate the minimum distance of every sample inside the hyperbolas.

Step 5. Make a hypersphere with radius of d_{\min} and count out \sum_+ and \sum_- .

Step 6. Calculate the diversity η by using Eq. (14).

Step 7. Calculate the membership by using Eq. (15).

4. Experimental results

To evaluate the performance of the proposed algorithm, a series of experiments are conducted. The obtained results are compared with the traditional SVM, the FSVM algorithm based on the distance between a sample and its cluster center [4], the FSVM algorithm based on border vector extraction (BVE-FSVM) [10], and the FSVM based on affinity (AAS-FSVM) [12]. For comparison, the four algorithms use the same Gaussian kernel

function: $K(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|^2 / 2\sigma^2)$. The penalty parameter C and the Gaussian kernel parameter σ are optimized by 10-fold cross-validation method. All the experiments are performed in MATLAB (R2008a) on a personal computer with a 2.26 GHz Celeron CPU, 1 GB memory, and Microsoft Windows XP.

4.1. Statistic of memberships

First we record the memberships of the key samples shown in the Figure 1 with the four FSVM methods. The randomly generated dataset belongs to a two-class classification problem, and it contains 196 samples (98 samples belonging to each class). Statistical results are given in Table 1.

Table 1. Membership values of key points in Figure 1.

| Algorithm | a | b | c | d | e |
|-----------|------|------|------|------|------|
| FSVM | 0.03 | 0.36 | 0.71 | 0.14 | 0.11 |
| AAS-FSVM | 0.30 | 0.50 | 0.73 | 0.41 | 0.27 |
| BVE-FSVM | 0.51 | 0.85 | 0.64 | 0.98 | 0 |
| QGSH-FSVM | 0.12 | 0.97 | 0.31 | 0.16 | 0 |

As can be seen from the results, the fuzzy membership of noise point a is the smallest among the recorded samples by all the methods, so the four algorithms can suppress noise effectively. In the QGSH-FSVM, although a is far from the negative hyperbola, its membership is still very small owing to the negative η_- . Point b is the boundary support vector. Its membership is nearly equal to 1 because its d_{\min}^- and η_- are both close to the largest value in the proposed method. The BVE-FSVM can also assign it a large value while the AAS-FSVM and FSVM fail. Although point c is close to the cluster center, it plays a minor role in classification, but only the proposed method can allot it a small membership. Point d is most probably outliers. It gets a small membership because of the short distance to the negative hyperbola and the small diversity. The outside points like e are abandoned because they make no difference for classification.

Sometimes the samples are distributed as in Figure 3, and the memberships are recorded in Table 2. Point a is nearest to the class center, but it is most probably outliers. When the FMF is defined by the FSVM and AAS-FSVM, it gets the largest value. Points b , c , d , and e get the same membership by using the present algorithms for their same distance to the class center, but apparently b and d are more important than c and e . When we use the algorithm proposed in this work, c is abandoned, a gets a small membership, b is assigned a large one, and the rest of the samples get their memberships according to their importance.

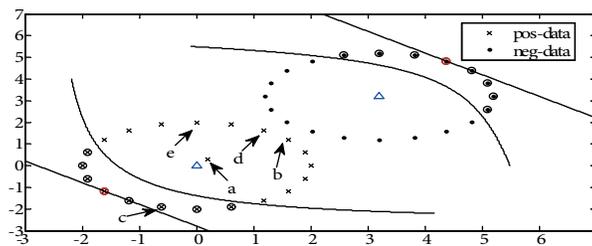


Figure 3. Specific sample situation with outliers.

The memberships are also of importance for fuzzy clustering algorithms. We use the classical fuzzy c-means algorithm (FCM) [17], possibilistic c-means (PCM) [18], and possibilistic fuzzy c-means (PFCM) [19] to

Table 2. Membership values of key points in Figure 3.

| Algorithm | a | b | c | d | e |
|-----------|------|------|------|------|------|
| FSVM | 0.76 | 0.44 | 0.44 | 0.44 | 0.44 |
| AAS-FSVM | 0.64 | 0.40 | 0.40 | 0.40 | 0.40 |
| BVE-FSVM | 0.72 | 1 | 0 | 1 | 1 |
| QGSF-FSVM | 0.34 | 0.98 | 0 | 0.85 | 0.47 |

calculate the memberships of the key points to the negative cluster shown in the Figure 1. The computational protocols are as follows: $\varepsilon = 0.00001$, $T_{\max} = 100$, $m = \eta = c = 2$, and $a = b = K = 1$. The membership value and clustering error rates are recorded in Table 3. It reveals that the three algorithms give memberships to points inversely related to the relative distance to the cluster centers. Point c gets the largest membership but plays a small role in classification. The proposed algorithm, however, assigns memberships according to their importance for classification, which is quite useful for the FSVM.

Table 3. Membership values and error rates by fuzzy clustering algorithms.

| Algorithm | a | b | c | d | e | Error rates |
|-----------|------|------|------|------|------|-------------|
| FCM | 0.09 | 0.51 | 0.96 | 0.77 | 0.81 | 11.61 |
| PCM | 0.36 | 0.52 | 0.85 | 0.45 | 0.33 | 9.28 |
| PFCM | 0.08 | 0.53 | 0.96 | 0.75 | 0.80 | 8.07 |

Since the BVE-FSVM has to determine the noises and outliers in advance, we do not compare it with the other algorithms in the next experiments.

4.2. Classification experiments

In order to test the classification performance of the proposed algorithm, we randomly generate five sets of two-dimensional data with different geometry and sparsity degrees as Figure 4 shows. They all belong to two-class classification problems. The first four sets of experiments are linear cases and the last set is a nonlinear case. The datasets adopted in the experiments are described in detail as follows:

1. This dataset is spherical-shape closely distributed, and it is made up of 184 samples (92 samples belonging to each class).
2. This dataset is also spherical-shape distributed, but it is much more sparse than the first set. It consists of 65 samples with 35 samples belonging to the positive class and 30 samples belonging to the negative class.
3. This dataset is long-shape closely distributed. In order to obtain the desired shape, the samples are randomly generated two times. The positive class contains 101 samples and the negative class contains 94 samples.
4. This dataset is long-shape sparsely distributed, and it is also generated two times. It is made up of 56 samples with 30 samples belonging to the positive class and 26 samples belonging to the negative class.
5. This dataset is produced for the nonlinear classification model. The positive class consists of 20 samples distributed along an ellipse and a noise sample near the center. The negative class contains 16 randomly generated samples and two artificially added noise samples.

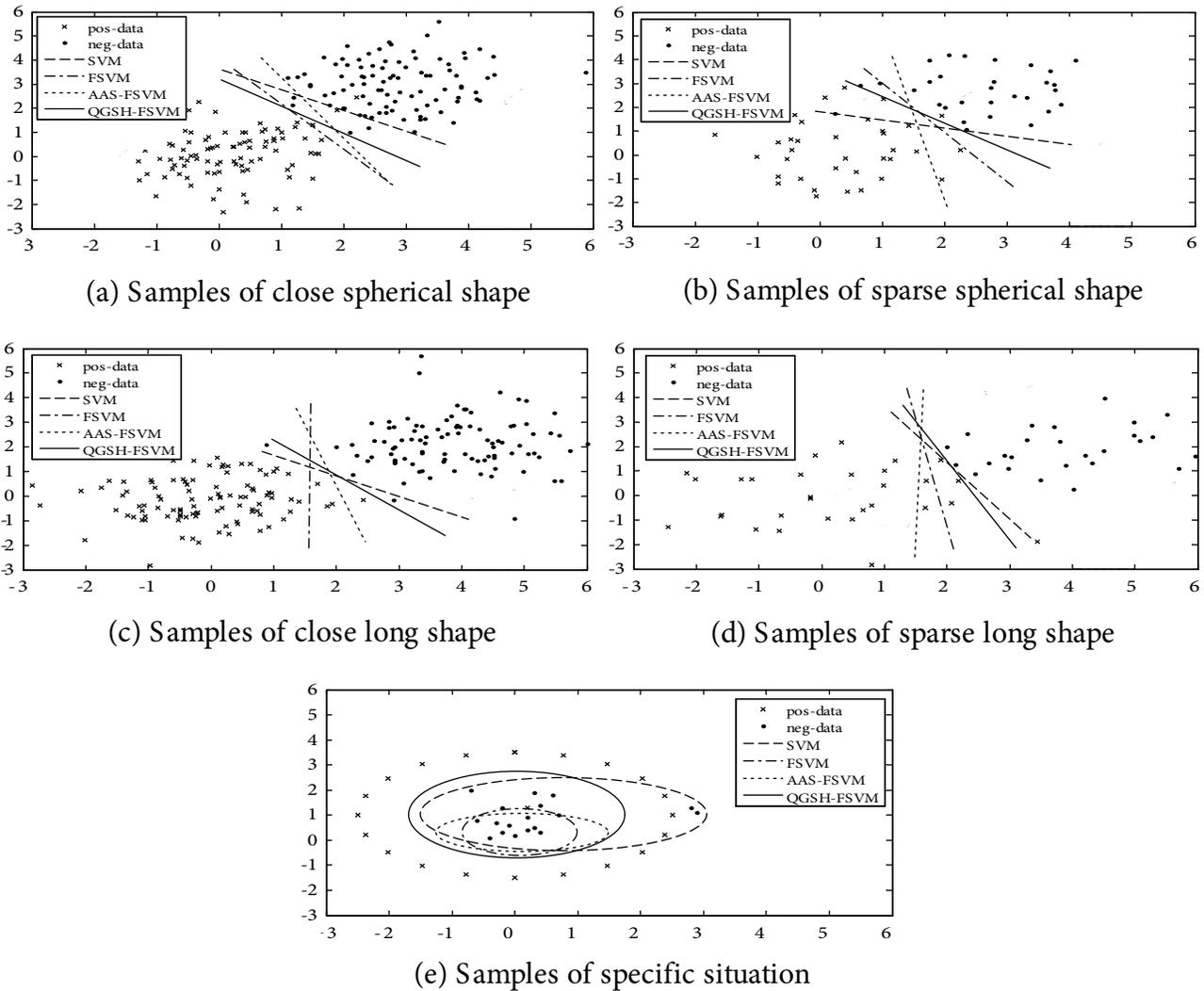


Figure 4. Experimental results of the four algorithms: a) samples of close spherical shape, b) Samples of sparse spherical shape, c) samples of close long shape, d) samples of sparse long shape, e) samples of specific situation.

According to the classification results, as Figures 4a–4e show, we can see that the FSVM improves the antinoise performance of the SVM algorithm to a certain degree. However, if the FMF is chosen improperly, the FSVM may also reduce the classification accuracy. The traditional FSVM suits samples with spherical-shape distribution, such as those shown in Figures 4a and 4b. The AAS-FSVM suits samples closely distributed, as Figures 4a and 4c show. However, the QGSH-FSVM can provide the best solutions for all situations, particularly for the nonlinear samples distributed as in Figure 4e.

4.3. Application experiments

To verify the performance of the QGSH-FSVM further, we simulate a circuit model by using OrCAD 10.5 software. The tolerance of the resistors and capacitors is set to 5% and 10%, respectively. After analyzing sensitivities of all components, we select R1, R7, R10, C1, and C2 as the potential faulty components. The fault kinds are shown in Table 4, where \uparrow and \downarrow mean that the fault value is more or less than the normal value.

Table 4. Nominal and fault values of 11 faults.

| Fault ID | Faults | Normal values | Fault values |
|----------|--------|---------------|--------------|
| F1 | NF | – | – |
| F2 | R1↑ | 10k | 14k |
| F3 | R1↓ | 10k | 6k |
| F4 | R7↑ | 280k | 370k |
| F5 | R7↓ | 280k | 180k |
| F6 | R10↑ | 150k | 200 k |
| F7 | R10↓ | 150k | 100 k |
| F8 | C1↑ | 0.18u | 0.25u |
| F9 | C1↓ | 0.18u | 0.1u |
| F10 | C2↑ | 1.5u | 2.3u |
| F11 | C2↓ | 1.5u | 1u |

In order to generate the simulation fault data, a Monte Carlo analysis method is used. In the Monte Carlo analysis, the number of runs and use distribution are set to 500 and Gaussian, respectively. Then we add 10% Gaussian noise to the collected data. Therefore, we get 550 sets of data. The target feature extraction adopts five layers of wavelet packet decomposing method. We increase the training samples from 20 to 160, and the remaining samples are used for testing.

Figure 5 displays the experimental results of the four methods. In general, the three FSVM algorithms can achieve higher classification accuracy than the traditional SVM when dealing with samples with noises. We can also see that the traditional FSVM and the AAS-FSVM have very similar performances. The AAS-FSVM is better than the traditional FSVM in most cases but worse when the number of training samples is 50 or 60. However, the QGSH-FSVM has a satisfactory advantage over the other three algorithms on classification accuracy and convergence speed, especially when the number of training samples is less than 90.

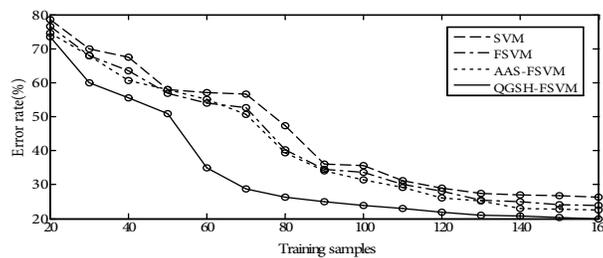


Figure 5. Error rate of the four algorithms.

Table 5 gives the training time statistics. Except for the traditional SVM, the time for the other algorithms consists of two parts. The first is the cost for training and the second is the cost for setting fuzzy memberships. Comparing the training time, the training speed of the QGSH-FSVM is much faster than the other methods due to its abandoning some samples [11]. Because the AAS-FSVM has to solve the quadratic problem twice and the QGSH-FSVM has to optimize the eccentricities, the two algorithms spend much time on setting fuzzy memberships. Although the QGSH-FSVM needs a few steps, it is still much easier to achieve than solving the quadratic problem after the eccentricities have been optimized.

Table 5. Training time of the four algorithms.

| Training sets | SVM | FSVM | AAS-FSVM | QGSF-FSVM |
|---------------|------|-----------|-----------|-----------|
| 40 | 0.37 | 0.34+0.01 | 0.32+0.38 | 0.22+0.53 |
| 80 | 0.92 | 0.87+0.01 | 0.88+0.52 | 0.74+0.77 |
| 120 | 2.14 | 2.03+0.02 | 2.02+0.88 | 1.72+1.04 |
| 160 | 2.86 | 2.77+0.02 | 2.74+1.01 | 2.06+1.18 |

5. Conclusion

In this paper, we introduce a design for the FMF by using hyperbolas and diversity to suppress outliers or noises. It can be seen from the statistics of memberships results that only the QGSF-FSVM can assign memberships to the samples according to their contribution to the classification without knowing the noises and outliers in advance. The noises and outliers will get small memberships and the support vectors can obtain the maximum memberships in the proposed method. It shows encouraging performances further in all the classification situations tested while the traditional FSVM and the AAS-FSVM can only succeed in some particular problems. Compared with other FSVM algorithms, this algorithm is independent of the geometry and sparsity degree of samples. The proposed method also performs best in terms of classification accuracy and convergence speed in the application experiment.

When it comes to the training times of all the algorithms, the QGSF-FSVM still has priority if the time of setting the fuzzy membership is not taken into account. However, much time is spent on optimizing the eccentricities. In future work, it would be interesting to investigate the methods of optimizing the eccentricities to improve the training efficiency and classification accuracy. In addition, there is no support vector whose membership is 1 and we will continue to research this. We hope that the QGSF-FSVM can be applied to many real-world applications.

References

- [1] Vapnik V. The Nature of Statistical Learning Theory. New York, NY, USA: Springer-Verlag, 1995.
- [2] Cristianini N, Schawe TJ. An Introduction to Support Vector Machines. Cambridge, UK: Cambridge University Press, 2000.
- [3] Zhang XG. Using class-center vectors to build support vector machine. In: Proceedings of the 1999 IEEE signal processing society workshop; 23–25 August 1999; Madison, WI, USA. New York, USA: IEEE. pp. 3-11.
- [4] Lin CF, Wang SD. Fuzzy support vector machines. IEEE T Neural Network 2002; 13: 464-471.
- [5] Li ST, Member IEEE, Chen CC. A regularized monotonic fuzzy support vector machine model for data mining with prior knowledge. IEEE T Fuzzy Syst 2015; 23: 1713-1727.
- [6] Huang HP, Liu YH. Fuzzy support vector machines for pattern recognition and data mining. Int J Fuzzy Syst 2002; 4: 826-835.
- [7] Chiang JH, Hao PY. A new kernel-based fuzzy clustering approach: support vector clustering with cell growing. IEEE T Fuzzy Syst 2003; 11: 518-527.
- [8] Tang J, Shi Y. New method of analog circuit fault diagnosis using fuzzy support vector machine. J Electron Meas Instrum 2009; 6: 7-12.
- [9] Zhang GX, Fei L, Du Z, Liu SY. Fuzzy support vector machine based on cluster hyperplane. Computer Engineering and Design 2008; 29: 3177-3178 (in Chinese with an abstract in English).

- [10] Wu Q, Liu SY, Du Z. Fuzzy support vector machine method based on border vector extraction. *PR & AI* 2008; 21: 332-337 (in Chinese with an abstract in English).
- [11] Ding SF, Han YZ, Yu JZ. A fast fuzzy support vector machine based on information granulation. *Neural Comput Appl* 2013; 23: 139-144.
- [12] Zhang X, Xiao XL, Xu GY. Fuzzy support vector machine based on affinity among samples. *Journal of Software* 2006; 17: 951-958 (in Chinese with an abstract in English).
- [13] An WJ, Liang MG. Fuzzy support vector machine based on within-class scatter for classification problems with outliers or noises. *Neurocomputing* 2013; 110: 101-110.
- [14] Shigeo A. Fuzzy support vector machines for multi label classification. *Pattern Recogn* 2015; 48: 2110-2117.
- [15] Mohadeseh S, Hossein N, Malihe M. A quantum inspired gravitational search algorithm for numerical function optimization. *Inform Sciences* 2014; 267: 83-100.
- [16] Esmat R, Hossein N, Saeid S. GSA: A gravitational search algorithm. *Inform Sciences* 2009; 179: 2232-2248.
- [17] Bezdek JC, Hathaway RJ. Convergence and theory for fuzzy c-means clustering: counterexamples and repairs. *IEEE T Pattern Anal* 1987; 17: 873-877.
- [18] Krishnapuram R, Keller JM. The possibilistic c-means algorithm: insights and recommendations. *IEEE T Fuzzy Syst* 1996; 4: 385-393.
- [19] Pal NR, Pal K, Keller JM, Bezdek JC. A possibilistic fuzzy c-means clustering algorithm. *IEEE T Fuzzy Syst* 2005; 13: 517-530.