International Journal of Control and Automation
Vol. 9, No. 12 (2016), pp.235-248
http//dx.doi.org/10.14257/ijca.2016.9.12.20

Parameter Optimizations of Multi-class Support Vector Machine Based on Seeker Optimization Algorithm

Huimin Ge, Long Chen and Jun Liang

School of Automotive and Traffic Engineering, Jiangsu University
hmge@mail.ujs.edu.cn

Abstract

In the traditional model of fault diagnosis, neural network classification requires high demand for the number and completeness of samples with a problem cannot be overcome--"the curse of dimensionality". While the actual bearing failure is a typical case of small sample with few samples and the number of different types of samples is asymmetrical and even not complete. And the pattern classification effects of the support rector machine in case of small sample are better. Therefore, according to the above comparative analysis, combined with the character of small samples of actual bearing failure mode, this paper selects to build classification model based on the support vector machines, and after researching, the model proved to be feasible.

***Keywords**: Seeker optimization algorithm; SVM; Parameter optimization*

1. A Non-Linear Support Vector Machine

Support vector machine is a machine learning method put forward by the Vapnik based on statistical learning theory [1], which is widely used in the field of classification and regression [2]. The advantage of this model is the minimization of structural risks, rather than the experiential risk of traditional statistics. With complete theoretical basis, SVM has been widely used in the field of face recognition, voice recognition, image and signal processing [3].

1.1. Constructions of Nonlinear SVM

In the face of solving such a nonlinear problem, we should make the nonlinear problem converted into a linear problem first by means of high-dimensional space conversion and then figure out the optimal hyperplane. The basic principles are as follows:

Suppose $\Phi : R^d \to H$ to be a nonlinear mapping, and the purpose of training is to find a function K. The conditions are as follows:

$$K(x_i, y_i) = \Phi(x_i).\Phi(x_j) \tag{1}$$

When the function F satisfies Mercer condition, it is called kernel function, and the objective function becomes:

$$Q(a) = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i,j=1}^{n} a_i a_j y_i y_j K(x_i.x_j) \tag{2}$$

While the corresponding classification function also becomes:

$$f(x) = \mathrm{sgn}\{\sum_{j=1}^{n} a_i^* y_i K(x_i.x) - b^*\} \tag{3}$$

1.2. Kernel Function Selection of SVM

Kernel function is essentially a kind of function with mapping function, which can achieve the mapping from non-linear to linear and map complex issues of the low-dimensional space to simple low-dimensional space. Kernel functions commonly used early are the following:

1. Liner Kernel Function : $K(x,y)=(x.y)^d$
2. Polynomial Kernel Function: $K(x,y)=((x.y)+c)^d, c\geq 0$
3. Gaussian radial basis Function: $K(x,y)=\exp(-\|x-y\|^2/\sigma^2)$
4. Sigmoid Kernel Function: $K(x,y)=\tanh(\rho(x.y)+c)$

After selecting kernel function, support vector machines require to choose two parameters: penalty factor C and kernel parameter σ^2 , in which the kernel parameter σ^2 represents the distribution or range character of the training sample data; the penalty factor C can control the smoothness of the curve and compromise experiential risks, which determines the strength of the size of the training error and generalization capabilities. The lower the value of C is, the contribution of each support vector in the model is more approximate, the curve is more smooth and it is better for the performance popularize of the subsequent data; on the contrary, the higher the value of C is, the contribution gap of each support vector in the model is more wider so that the greater support vector plays a leading role in the decision-making function.

In summary, as to the support vector machine SVM, the choice the characteristic parameters are critical. We use SOA search algorithm to do the parameter optimization of the SVM model.

1.3. Classification SVM

Commonly used multi-class support vector machine mainly includes one-to-many, one-to-one, directed acyclic graph, error correction coding SVM, binary SVM and so on. Elaborations on them are as follows.

1. One-to-many SVM

In one-to-many SVM, we need to solve k (k> 1) class classification problems and build k binary classifiers through the model. The main advantages and disadvantages of one-to-many SVM model are as follows:

Advantages: Since the model constructs k binary classifiers to solve k class problems, so the number of classification function is small, and the classification speed is faster.

Disadvantages: While building k binary classifiers, the construction of each two classifiers needs to train sample, so the training time of the model is longer.

2. One-to-one SVM

In one-to-one SVM, in order to build all of the two classifiers, it is necessary to take any two sample for training in the k class training samples, and the number of two classifiers constructed is k (k-1) 2. The main advantages and disadvantages of one-to-one SVM model are as follows:

Advantages: It is not sensitive to the imbalance on the amount of classification data.

Disadvantages: The larger k is, the more the number of classifiers is, and training is slower.

3. Directed acyclic graph SVM

Directed acyclic graph SVM is a multi-value classification algorithm with the most perfect theory. Training of this model is similar to one-to-one SVM. In the test, it chooses the decision tree method to determine the category space. For the directed acyclic graph SVM, its advantages and disadvantages are obvious. Compared with the aforementioned two models, the classification speed of this model is faster, but the disadvantage is that the result of classification is quite sensitive to the selection of root node.

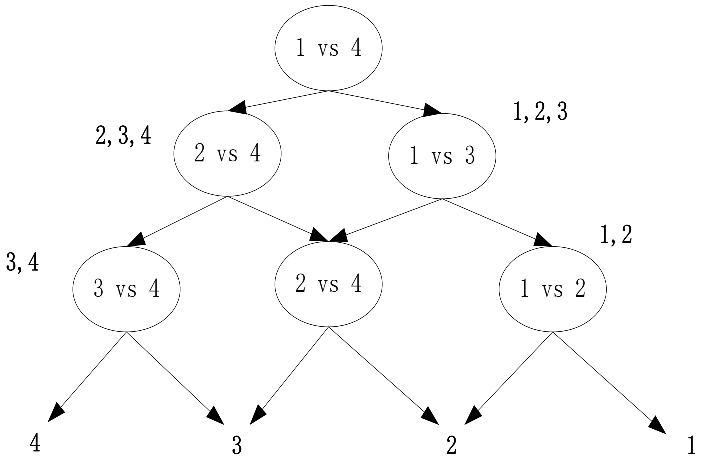


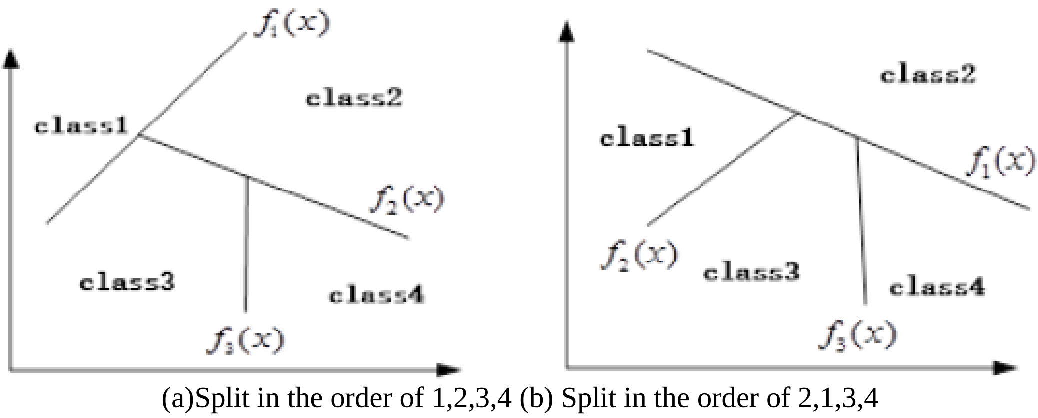
Figure 1. The Classification Effect of DAGSVM

4. Error correction coding SVM

Error correction coding SVM can solve multi-class classification problems effectively. When the categories needed to classify is k, use the purchase dichotomous method to resolve, and assign each category with binary error correcting code sequence, the length of which is L. Thus, the entire classification process can form and construct a matrix with k rows and L columns. The speed of this type of classifiers depends on the codebook's design.

5. Multi-class support SVM based on binary tree

In multi-class SVM based on binary tree, first of all, divide the categories into two subclasses and then divide the subclasses. Do the partition in the same way until a single type. The model is able to overcome the inseparable phenomenon, and when there is a need for k-class classification, the number of SVM classifier constructed is only k-1, so the test time is shorter.



(a)Split in the order of 1,2,3,4 (b) Split in the order of 2,1,3,4

Figure 2. Different Segmentation Order for Four Kinds of Samples

However, the classification accuracy of multi-class support SVM based on binary tree is affected by the tree structure [4]. As shown in Figure 2, if there was a node classification error, it will be passed down.

2. Typical Optimization Algorithm Analysis

2.1. Simulated Annealing Algorithm

Simulated annealing algorithm (SA) was proposed in 1953, and its core is doing simulation and abstraction on the annealing process of solid materials to solve optimization problems. Simulated annealing algorithm can effectively solve problems of combinatorial optimization and minimization optimization of continuous function with a better result. However, this method still exist some shortcomings.

Firstly, only at the condition of a sufficiently high initial temperature and a declining coefficient close enough to 1 can this method achieve optimal results; Moreover, the additional computational algorithm is large so the search speed is low.

2.2. Genetic Algorithms

Genetic algorithm (GA) is fundamentally a heuristic algorithm based on multipoint random search. Genetic algorithm can effectively solve complex nonlinear problems, and the advantages of it are having a strong capability of global optimization, a high speed of optimizing and being able to process parallelly.

2.3. Ant Colony Algorithm

Ant colony algorithm (ACO), put forward by Italian scholars, achieve the optimal value seeking problem mainly by simulating the behavior of real ants characteristics. The algorithm is widely used in solving scheduling problems, quadratic assignment problems and knapsack problems.

3. Seeker Optimization Algorithm

Seeker optimization algorithm (SOA), in essence, is a heuristic random search algorithm based on population [5]. Now it has made successful applications in function optimization, filter design, power system [6] and other fields.

Based on the fundamental behavior of the crowd, SOA can effectively simulate the four acts in the evolution of human species and calculate the seeking step of the next generation, according to seeking rules and based on fuzzy inference, in order to achieve the location update of seekers between different generations of people. The algorithm has several features of high speed of convergence, high precision of convergence and strong global seek capability. In this paper, we choose seekers optimization algorithm to do the parameter optimization of SVM model.

3.1. Basic Behavior

3.1.1. Self-Serving Behavior

In order to live in nature, it is necessary for social groups to cooperate with each other, in which the crowd is the most intelligent group. Human development is inseparable from the mutual cooperation between each other, which reflects in two forms: self-serving behavior and altruistic behavior. As the name suggests, self-serving behavior takes priority of the self as the basic principle of development; In contrast, altruistic behavior takes community development as the basic principle. In human populations, self-serving behavior is an objective reality, the performance of which is that each individual continues to do self-development by means of learning and makes himself move toward the optimum position through knowledge.

3.1.2. Altruistic Behavior

As described above, in the human population, collaborative behavior between each other comprises another form, the altruistic behavior, which is also an objective existence. The performance of it is that each individual of population groups takes the development of the group as the principle and communicates and shares with each other constantly to promote overall development. In SOA algorithm, this altruistic behaviors how up as a search strategy of community priority to guide the individual moving to the historical neighborhood of or current best position.

3.1.3. Pre-Moving Behavior

In the human population, the pre-movement behavior is a predictive behavior whose intelligent individuals can perform an operation initiatively by means of special ability to adjust the direction. In SOA algorithm, the pre-movement behavior is a comprehensive result, and intelligent individual can guide the completion of the pre-movement behavior by past behaviors. The performance is during the search process, according to influencing factors and the predicted behavior results, to adjust their own strategies timely to achieve optimal forward guide.

3.1.4. Uncertainty Behavior

Some related researches suggest that the optimal solution is often around the initial solutions while targeting the optimization of specific problems. And in the Seeker optimization algorithm, there is also a concrete manifestation of this behavior, which called deterministic behavior. The performances are some guidelines as follows: when the individual of the population is excellent, search range should be shrunk to nearby in order to increase the probability of finding the optimal solution; Conversely, when the fitness of individual is poor, the crowd should expand the scope of the search to improve the search efficiency of the optimal solution, since the optimal solution may be distributed in a larger context.

As we can see from the description of the basic principles of optimization algorithm, the search step length and search direction of population groups are the key parameters to complete its evolution and search. Next the two issues will be discussed in detail.

3.2. Calculations on Search Step Length and Search Direction

As we can see from the description of the basic principles of the crowd search algorithm, the search step length and search direction of population groups are the key parameters to complete its evolution and search. Next the two issues will be discussed in detail.

3.2.1. The Determination of the Search Step

Based on fuzzy mathematic theory, SOA algorithm uses fuzzy theory to achieve the uncertainty reasoning process, in which the fuzzy rules can be described as follows: The smaller the target function is, the smaller size of search step will be. Therefore, the definition of the search step fuzzy variables is shown below, in which the function is a Gaussian membership function.

$$u_A(x) = \exp[-(x-u)^2 / 2\delta^2] \tag{4}$$

Linear membership function can be represented by the following formula:

$$u_i = u_{\max} - \frac{s - I_i}{s - I}(u_{\max} - u_{\min}), i = 1, 2, ..., s \tag{5}$$

$$u_{ij} = rand(u_i, 1), \ j = 1, 2, ...D \tag{6}$$

Where, u_i is the degree of membership of the objective function value i, and u_{ij} is the desired degree of membership; D is the dimension of the search space.

After calculating the degree of membership from the above formula, the step length of behaviors can be obtained by the following formula:

$$\begin{aligned} &\alpha_{ij} = \delta_{ij}\sqrt{-\ln(u_{ij})} \\ &\delta_{ij} = \omega.abs(x_{\min} - x_{\max}) \\ &\omega = (T_{\max} - t)/T_{\max} \end{aligned} \tag{7}$$

3.2.2. Determination of the Search Direction

Three basic directions of seeker optimization algorithm can be calculated as follows:

$$\begin{aligned} &\vec{d}_{i,ego}(t) = \vec{p}_{i,best} - \vec{x}_i(t) \\ &\vec{d}_{i,alt}(t) = \vec{g}_{i,best} - \vec{x}_i(t) \\ &\vec{d}_{i,pro}(t) = x_i(t_1) - x_i(t_2) \end{aligned} \tag{8}$$

Considering various factors, take three directions weighting average randomly to determine the final set of search direction:

$$\vec{d}_i(t) = sign(\omega\vec{d}_{i,pro} + \varphi_1\vec{d}_{i,ego} + \varphi_2\vec{d}_{i,alt}) \tag{9}$$

Where, $t_1, t_2 \in \{t, t-1, t-2\}$; $\vec{x}_i(t_1)$ and $\vec{x}_i(t_2)$ is respectively the best position of $\{\vec{x}_i(t-2), \vec{x}_i(t-1), \vec{x}_i(t)\}$; Sign is the sign function.

3.3. Simulated Annealing Algorithm

The main ideas and key steps of seeker optimization algorithm have been introduced in detail above and based on it, to introduce the basic steps of seeker optimization algorithm:

Step 1: Setting algorithm parameters. Based on experience and the actual problem, determine search parameters like the population size, algebraic algorithm and so on.

Step 2: Initialization of population groups. In this step, complete initialization of each individual, and the Initialization algorithm is decided on the practical problems. After the initialization, each individual has an initial value to provide the basis of the next seeker. The initial position of population crowd is shown in the following formula.

$$\left\{\vec{x}_i(t) \mid \vec{x}_i(t) = (x_{i1}, x_{i2}, ..., x_{iM})\right\} \tag{10}$$

Where, $i = 1, 2, 3, ..., sizepop, t = 0$

Step 3: Evaluation on the initial crowd. Based on the problem needs to be solved, make it specific and calculate the fitness value of each individual of initialization crowd and select the best one.

Step 4: According to the seeking rules and basic principles, to calculate the search step length and search direction of each individual.

Step 5: Updating crowd group location. According to the following formula, update crowd the location of crowd group.

$$\begin{aligned} &\Delta x_{ij}(t+1) = \alpha_{ij}(t)d_{ij}(t) \\ &x_{ij}(t+1) = x_{ij}(t) + \Delta x_{ij}(t+1) \end{aligned} \tag{11}$$

Step 6: Do once evolution, the evolution generation increase 1, t = t + 1.

Step 7: Judge whether seeker optimization algorithm has reached the condition of termination. When the evolution generations of seeker optimization have reached the peak or the optimal fitness has reached the initial setup requirement, then the termination condition is reached. The algorithm stops going on, jumping out of the program. Otherwise, the algorithm will not terminate and go to Step3 continuing.

4. Analysis on the Performance of SOA

As mentioned above, in the actual application process of SVM, the classification performance depends directly on the selection of kernel function and penalty factor. In studies on the parameter optimization of SVM, different algorithms have been used, in which some common ones like GA, PSO, *etc*. The studies show: PSO is simple in structure and easy to control, with more stable performance in the moving objective function.

With the issue of parameter optimization of SVM, SOA, with the better performance, should be applied to optimize the parameters of SVM model. In order to convince the effect of SOA, three typical test functions are selected to do the simulation in this section, comparing SOA with PSO, an algorithm with better performance, and analyzing. During the simulations, taking convergence accuracy and computation time as the comparative indicators, each function optimizates 10 times, and take the average of them. Hardware and software environment of the experiments are as follows: Intel (R) Core (TM) i5 CPU, 2.00 GB of memory, Matlab R2010b.

4.1. Simulation Analysis of Sphere Function

The expression of Sphere function is as follow：

$$\min f(x)=\sum_{i=1}^{n} x_i^{2}, \left|x_i\right| \le 15, n=10 \tag{12}$$

Three-dimensional graphics of Sphere function is shown in Figure 3.

Use PSO optimization algorithm to do minimum function optimization, and take the function value as the fitness value of PSO. In order to compare the two algorithms in the same condition, set evolution generation of PSO as 100, the population size as 50, and according to the optimization problem to be solved, select the optimal parameters of global seeking parameter c_1 and local seeking parameter c_2. Through experimental compares, determine $c_1 = 1.5$, $c_2 = 1.5$. The data of the 10 simulation groups are shown in the following table, and we can see that the average minimal optimization value of the algorithm is 0.2327, and the time is 0.1456 seconds.

Use SOA optimization algorithm to do minimum function optimization, and take the function value as the fitness value of SOA. Set evolution generation of SOA as 100, the population size as 50. Set the maximum degree of membership of the algorithm Umax = 0.95, the minimum degree of membership Umin = 0.011, the maximum weight Wmax = 0.9, the minimum weight Wmin = 0.1. The data of the 10 simulation groups are shown in the following table, and we can see that the average minimal optimization value of the algorithm is 0.0137, and the time is 0.098 seconds.

Figure 4 compares the optimization curves of the two typical algorithms, and after integrated quantitative comparison, we can be see that SOA algorithm has higher calculating speed and better accuracy. So SOA algorithm is superior to PSO.

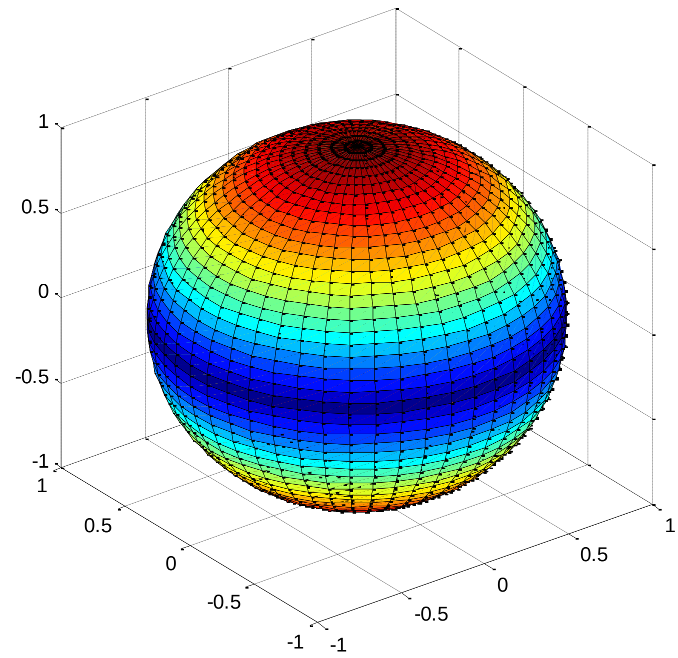


Figure 3. The Graph of Sphere Function

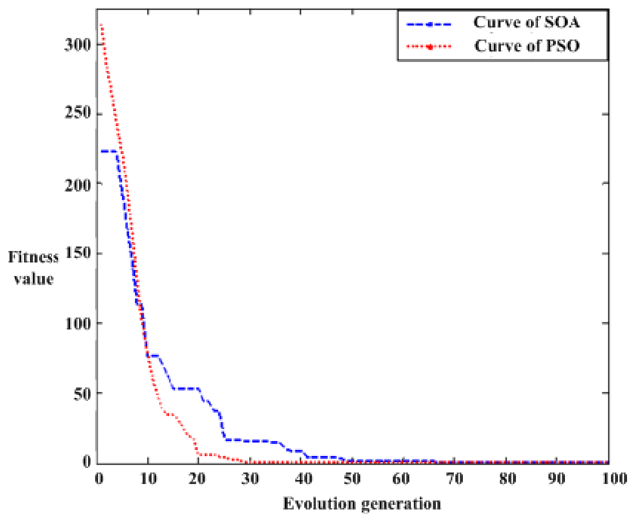


Figure 4. The Optimization Compare for Sphere Function by SOA and PSO

Table 1. The Optimization Result of Sphere Function by PSO

Number	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	f_{min}	Time
1	-0.02	0.00	0.05	-0.01	0.02	-0.02	0.10	0.00	-0.02	-0.04	0.0169	0.41
2	0.02	-0.04	0.01	0.03	0.08	-0.03	-0.01	0.08	-0.00	0.08	0.0280	0.02
3	-0.02	0.01	-0.01	-0.01	-0.06	0.13	-0.08	-0.00	-0.05	-0.01	0.0346	0.22
4	-0.07	0.01	0.06	-0.01	-0.02	-0.08	0.00	0.02	-0.10	-0.01	0.0163	0.24
5	-0.05	-0.00	-0.01	-0.00	-0.05	-0.08	0.05	0.02	-0.00	0.02	0.0175	0.23
6	-0.01	-0.00	0.01	0.02	0.08	-0.01	-0.12	0.11	0.03	0.01	0.0389	0.02
7	0.00	-0.00	-0.04	0.08	0.03	-0.01	0.03	-0.04	0.07	-0.02	0.0214	0.02
8	0.00	-0.00	-0.03	0.00	0.12	-0.02	0.00	0.03	-0.01	0.02	0.0182	0.02
9	-0.05	-0.04	0.06	-0.00	0.03	0.05	0.00	-0.00	0.02	0.07	0.0185	0.02
10	-0.02	-0.04	-0.02	0.06	-0.09	-0.02	-0.03	-0.00	-0.05	-0.01	0.0224	0.256
Average											0.02327	0.1456

Table 2. Result of Sphere function by SOA

Number	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	f_{min}	Time
1	−0.00	-0.02	0.000	0.029	0.001	-0.004	-0.055	0.004	-0.083	-0.065	0.0160	0.19
2	-0.06	0.01	-0.00	-0.003	-0.021	-0.016	0.045	-0.016	0.015	0.038	0.0091	0.01
3	-0.02	0.03	0.011	0.074	-0.011	-0.003	-0.008	0.063	0.0196	-0.017	0.0126	0.09
4	0.01	-0.01	0.018	-0.033	-0.022	-0.004	0.008	0.031	-0.066	-0.019	0.0081	0.20
5	-0.03	-0.03	-0.028	0.005	-0.032	-0.046	-0.012	0.010	-0.068	0.016	0.011	0.18
6	-0.03	0.02	0.009	-0.012	0.060	0.030	-0.049	0.076	-0.041	-0.051	0.0198	0.01
7	-0.00	0.05	-0.004	-0.024	0.032	-0.016	0.061	0.041	0.008	-0.053	0.0132	0.03
8	-0.01	-0.02	0.016	0.044	0.037	0.054	0.015	0.055	0.036	-0.034	0.0131	0.01
9	0.00	0.03	-0.012	0.068	0.077	-0.006	-0.01	0.019	0.022	0.02	0.0137	0.07
10	0.00	-0.01	0.025	-0.026	0.023	0.105	0.046	0.006	0.036	-0.06	0.0204	0.19
Average											0.0137	0.098

4.2. Simulation Analysis of Schaffer Function

The expression of Schaffer function is as follow:

$$\min f(x,y) = 0.5 + \frac{\sin^2\sqrt{x^2+y^2} - 0.5}{[1+0.001(x^2+y^2)]^2}, \quad -10 \le x, y \le 10 \tag{13}$$

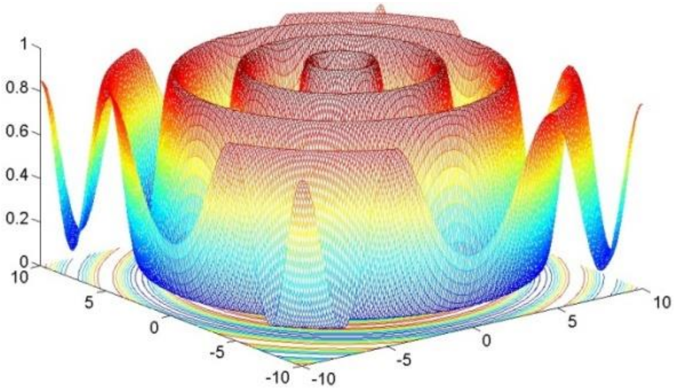


Figure 5. The Graph of Schaffer Function

Table 3. The Optimization Result of Schaffer Function by PSO

Times of experiments	x_1 (10^{-3})	x_2 (10^{-3})	Optimal value(10^{-5})	Time
1	1.4	-0.6	4.658	0.0175
2	3.7	-0.4	3.260	0.0160
3	1.3	-0.77	0.237	0.1880
4	7.1	3.2	6.100	0.1648
5	5.0	-10.8	1.420	0.1548
6	-5.9	9.1	1.165	0.1621
7	-0.1	9.3	9.612	0.1586
8	-0.7	2.1	1.501	0.1751
9	0.7	-0.8	3.516	0.1552
10	-8.4	99.0	16.92	0.1597
Average			8.000	0.1351

Table 4. The Optimization Result of Schaffer Function by SOA

Times of experiments	x_1 (10^{-3})	x_2 (10^{-3})	Minimum value(10^{-5})	Time
1	0.0019	-0.0013	0.5171	0.0108
2	0.5078	0.1151	0.0271	0.0111
3	4.8000	1.2000	2.4146	0.1520
4	-0.6000	4.0000	1.6338	0.1290
5	-0.5747	-0.7139	0.0839	0.1116
6	-2.9000	0.1000	0.8344	0.1329
7	-1.6000	3.1000	1.2432	0.1432
8	0.2617	0.0590	0.0071	0.1598
9	0.9185	-0.1922	0.0880	0.1320
10	0.7000	-1.6	0.3148	0.1120
Average			0.7164	0.0975

Three-dimensional graphics of Schaffer function are shown in Figure 5.

Firstly, use PSO algorithm to do the function optimization. The settings are same to the above. Simulation data of 10 groups are as shown in table 3.

Use SOA algorithm to do the function optimization. The settings are same to the above. Simulation data of 10 groups are shown in table 4.

4.3. Simulation Analysis of Rastrigin Function

The expression of Rastrigin function is as follow：

$$\min f(x_1,x_2)=20+x_1^{2}+x_2^{2}-10[\cos(2\pi x1)+\cos(2\pi x^2)], x_1,x_2\in[-5,5] \tag{14}$$

Three-dimensional graphics of Rastrigin function is shown in Figure 7.

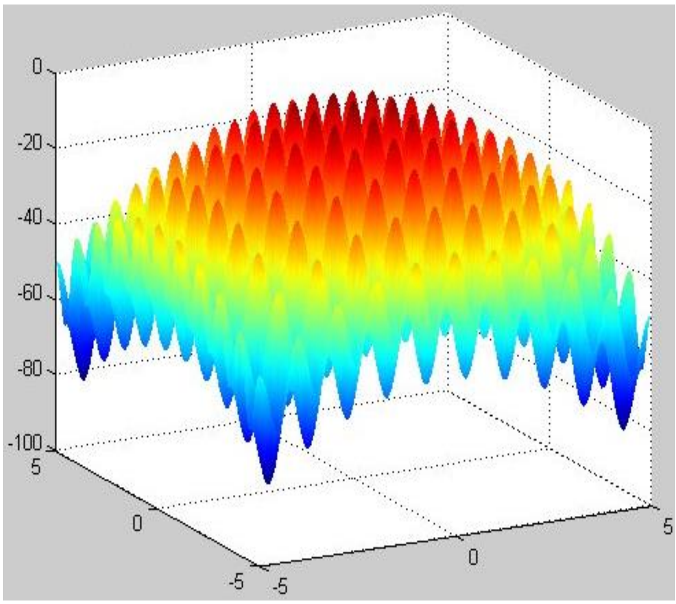


Figure 6. The Graph of Rastrigin Function

Table 5. The Optimization Result of Rastrigin Function By PSO

Times of experiments	x1	x2	Optimal value	Time
1	0.0011	0.0032	0.0022	0.1322
2	0.0020	0.0077	0.0126	0.2373
3	0.0001	-0.0027	0.0014	0.3064
4	0.0037	0.0003	0.0027	0.2346
5	0.0014	0.0023	0.0014	0.2529
6	0.0039	0.0054	0.0088	0.2424
7	0.0104	-0.0031	0.0234	0.2422
8	0.0051	0.0013	0.0054	0.2630
9	-0.0016	0.0043	0.0042	0.2474
10	-0.0035	0.0048	0.0071	0.2441
Average			0.00692	0.2414

Table 6. The Optimization Result of Rastrigin Function by SOA

Times of experiments	x1	x2	Optimal value	Time
1	0.0054	0.0020	0.0065	0.1109
2	-0.0024	0.0011	0.0014	0.1926
3	-0.00014	0.00086	0.000152	0.2983
4	-0.0049	0.0014	0.0052	0.2012
5	-0.00075	0.00051	0.00016	0.2329
6	-0.0008	-0.0030	0.0019	0.2015
7	-0.0039	-0.0022	0.0040	0.2109
8	0.0011	0.0047	0.0047	0.2342
9	-0.0009	-0.0011	0.00038	0.2198
10	0.0036	-0.0003	0.0026	0.2065
Average			0.0027	0.2108

Combining the three typical functions mentioned above, Sphere, Schaffer and Rastrigin, and after the comparative study between PSO and SOA, it can be seen that compared to PSO, SOA can obtain the smaller function value and reflect the better convergence accuracy, which can effectively avoid the prematurity phenomenon. Moreover, SOA has a higher calculation speed and a higher comprehensive efficiency. Therefore, we choose SOA, with higher accuracy, to do the parameters optimization of SVM model.

5. Parameters Optimization Process of Multi-Class SVM Based On SOA

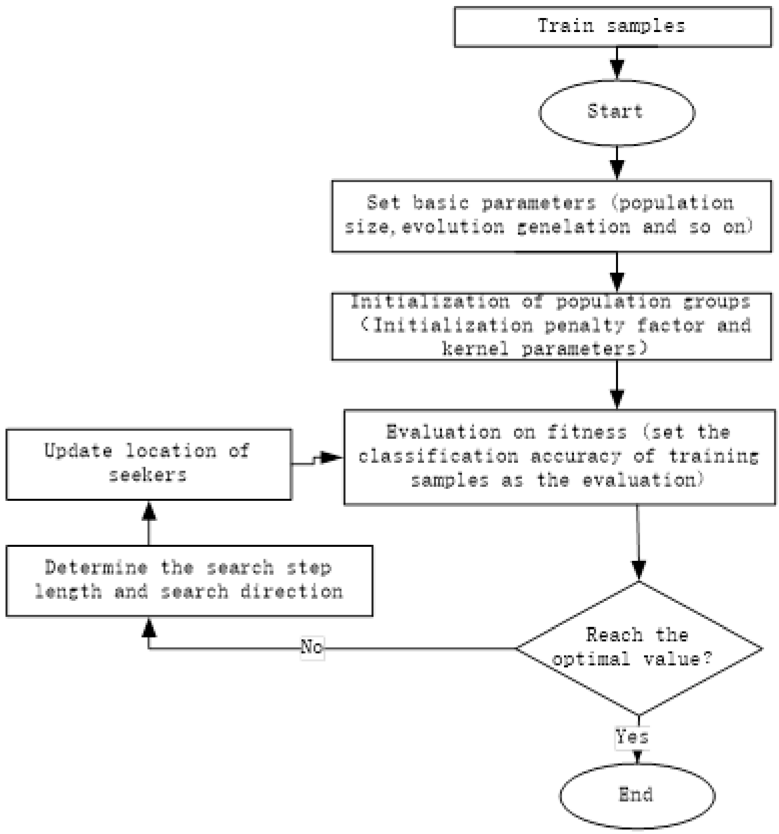


Figure 7. The Process of Multi-Class SVM

With the issue of parameter optimization of SVM, parameters optimization method of multi-class SVM based on SOA is put forward, the main flow of which is shown below. At the beginning, do the partition of the samples, and take the training accuracy of training samples of SVM as the fitness. At first, initialize the parameters, which include the evolution generation, population size and so on, and then initiate the seekers. Since the two parameters were optimized, so seeker dimension is 2. Do the fitness evaluation of the initialized sample, calculate the accuracy of training samples while using initial parameters. And then judge the accuracy, when it reach the optimal value, stop the optimization algorithms. If not, calculate the seeking step and seeking direction, and update the location of seeker individual, do the evaluation analysis again until it reaches the optimal value or the maximum evolution algebra.

6. Conclusions

This paper does some researches on support vector machine model and its parameter optimization methods. Firstly, analyze the basic theories of support vector machine, and point out the importance and influence of the issue of parameter optimization. Based on it, discuss the optimization issue and seeker optimization algorithm respectively. The simulation shows that SOA algorithm has good convergence rate and convergence precision, suitable for the parameter optimization of multi-class support vector machine.

During the application process of SVM, the actual application results are often effected by the selection of parameters. SOA algorithm has advantages in the convergence rate and convergence precision between the relevant optimization algorithms. Focusing on parameter optimization issues in building the multi-class support vector machine model, this paper establishes an optimization method of multi-class support vector machine based on the seeker optimization algorithm and do the simulation analysis on the performance of this algorithm.

References

[1] A. Colorni, M. Dorigo and V. Maniezze, “Distributed Optimization by Ant Colonies”, Proe 1st European conf artificial life.Pans, Franee: Elsevier, **(2009)**, pp. 134-142.

[2] A. R. Webb, “Statistical Pattern Recognition Second Edition’’, **(2004)**, pp. 240-263.

[3] C. Dai, W. Chen, Y. Song and Y. Zhu, “Seeker optimization algorithm: A novel stochastic search algorithm for global numerical optimization”, Journal of Systems Engineering and Electronics (English Edition), vol. 21, no. 2, **(2010)**, pp. 300-311.

[4] W. Yao and S. Qin, “Research of fire alarm system based on extension neural network”, Review of computer engineering studies, , China vol. 2, no. 1, **(2015)**, pp.1-8.

[5] P. Shen, H. Wang, Z. Meng, Z. Yang, Z. Zhi, R. Jin and A. Yang, “An improved parallel bayesian text classification algorithm”, Review of computer engineering studies Ningbo, China, vol.3, no. 1,**(2016)**, , pp. 6-10.

[6] R. Jin, C. Kou and R. Liu, “Biclustering algorithm of differential co-expression for gene data”, Review of computer engineering studies, Ningbo, China, vol.1, no.1, **(2014)**, pp.7-12.

[7] Y. T. Sheen, “A complex filter for vibration signal de-modulation in bearing defect diagnosis”, Journal of Sound and Vibration, vol. 276, **(2004)**, pp. 105-119.

[8] The Case Western Reserve University. Bearing data center [EB/OL].[2009-08-06]. http://www.eecs.cwru.edu/laboratory/bearing.

[9] C. J. C.A. Burges, “Tutorial on Support Vector Machines for PaRem Recognition”, Data Mining and Knowledge Discovery, vol. 2, no. 2, **(1998)**, pp. 121-167.

[10] V. N. Vapnik and A. Y. Chervonenkis, “On the uniform convergence of relative frequencies of events to their probabilities”, Theory of Probability & Its Applications, vol. 16, no. 2, **(2011)**, pp. 264-280.

[11] K.R. Muller, S. Mika and G.R. Atsch, “An Introduction to Kernel—based Learning Algorithms”, IEEE Transactions on Neural Networks, vol. 12, no. 2, **(2001)**, pp. 181—201.

[12] Z. Zhang, D. Y. Yeung and J. T. Kwok, “Wishart processes: A statistical view of reproducing kernels”, Technical report, Department of Computer Science, Hong Kong University of Science and Technology, **(2004)**.

[13] V.Popovicia and J.P. Thiran, “Face Detection Using an SVM Trained in Eigenfaces Space”, Lecture Notes in Computer)Science, vol. 2688, **(2003)**, pp. 190-198.

[14] T. G. Dietterich and G. Bakiri, “Solving Multiclass Learning Problemsvia Error—correcting Output Codes”, J.Artif.Intell.Res., vol. 2, **(1995)**, pp. 263-286.

[15] L. Zhang, J. Xu and J. Yang, “Multi scale morphology analysis and its application to fault diagnosis”, Mechanical Systems and Signal Processing, vol. 22, no. 3, **(2008)**, pp. 597-610.

[16] B. Li, P. Zhang and Z. Wang, “A weighted multi-scale morphological gradient filter for rolling element bearing fault detection”, ISA transactions, vol. 50, no. 4, **(2011)**, pp. 599-608.

