Parameter Selection of a Support Vector Machine, Based on a Chaotic Particle Swarm Optimization Algorithm

Huang Dong[1]*, Gao Jian*[2]

[1] *College of Mathematics and Computers Science, QinZhou University, QinZhou, China*
[2] *Panjin Vocational and Technical College, Panjin, China*
Emails: qzxyhd@163.com   gao-jian-21@163.com

*Abstract: This paper proposes a SVM (Support Vector Machine) parameter selection based on CPSO (Chaotic Particle Swarm Optimization), in order to determine the optimal parameters of the support vector machine quickly and efficiently. SVMs are new methods being developed, based on statistical learning theory. Training a SVM can be formulated as a quadratic programming problem. The parameter selection of SVMs must be done before solving the QP (Quadratic Programming) problem. The PSO (Particle Swarm Optimization) algorithm is applied in the course of SVM parameter selection. Due to the sensitivity and **frequency** of the initial value of the chaotic motion, the PSO algorithm is also applied to improve the particle swarm optimization, so as to improve the global search ability of the particles. The simulation results show that the improved CPSO can find more easily the global optimum and reduce the number of iterations, which also makes the search for a group of optimal parameters of SVM quicker and more efficient.*

Keywords: Support vector machine, parameter selection, particle swarm optimization, chaotic optimization.

1. Introduction

In the 1990ies, Vapnik and other researchers proposed a new and efficient method of machine learning using support vector machines [1-2], and the study of statistical

learning theory. Machine learning methods generally have the shortcomings of being small-sample, nonlinear, over-learning and of high dimension, as well as having local minima. The Support Vector Machine (SVM), with a high actual promotional value, has the advantage of minimizing the structure risk to solve the problem in insufficient presence of the above-mentioned algorithms [3]. With the development of artificial intelligence and machine learning, SVMs play a very important role. They have been applied in classification, time prediction function estimation and other areas and have attracted the attention of many researchers.

In general, a SVM algorithm is a constrained convex quadratic optimization problem solving tool [4]. Before solving optimization problems, it is necessary to select certain parameters, such as the regularization coefficient C, the core width parameter a and the insensitive coefficient. SVM parameter selection criteria include the two aspects of optimization and parameter adjustment approach [5]. Usually, several experiments are used for the determination of SVM parameters, in order to collect large amounts of data from a number of experiments and to therefore select from optimal parameters. However, this method is not efficient and it is not able to select the optimal parameters. While determining the parameters of SVM, the gradient descent method can be used to save time. However, this method can easily fall into the problems of local optimal algorithms [6], with a high sensitivity initial value and linear search process. Compared with the above-mentioned method, using a genetic algorithm to determine the parameters of a SVM not only saves time in parameter selection, but also weakens the negative impact of selecting the initial value [7]. Nevertheless, this algorithm has also the obvious shortcoming that it is not able to easily achieve its goal, because a crossover or a mutation in a corresponding manner should be designed according to some optimization problems [8].

This paper uses the improved chaotic particle swarm optimization to optimize the parameter selection of SVMs. A SVM training algorithm is a quadratic programming tool to solve the problem [9]. Firstly, the nuclear width parameter and the regularization coefficient of the optimization problem and other parameters need to be determined. The appropriate selection of the parameters is of great importance to the performance of SVMs. In this paper the chaotic particle swarm optimization algorithm is used to select the parameters of SVMs, while improving the algorithm by the initial value of the chaotic motion sensitivity and ergodicity. The experimental results show that a set of optimal parameters of SVMs can be quickly found by using the improved chaos particle swarm optimization algorithm. Compared with a genetic algorithm, this algorithm has, among the other properties, the following advantages: it requires only a small amount of calculation, it is easy to implement and highly efficient.

2. Support vector machine analysis

Support vector machines were first proposed in the 1990-ies by V a p n i k, L e v i n and L e C u n [10], as a method based on statistical theory. They are new, machine learning algorithms, based on structural risk minimization principle. This results in

a hyper plane, thereby enabling the spacing between the two sides to maximize data. This method has been widely used in practice, such as pattern recognition and intelligent modeling and has also good generalization ability; thus it is receiving increasingly widespread attention from scholars [11].

SVM algorithms are achieved by the following formula. Here $\phi(\bullet)$ represents a non-linear transformation. That is, the input space is mapped to a corresponding high-dimensional space. At the same time, in this high-dimensional space, a linear function of $f(x) = w^{\mathrm{T}}\phi(x) + b$ is used to fit the sample data.

Suppose that

$$x_i \in R^n, \; y_i \in R, i = 1, ..., l.$$

Considering the ease of the observed sample, in the above formula R^n represents the input space. This allows a linear regression SVM to be simply treated as a constrained quadratic programming optimization problem:

(1)
$$\min \left[\frac{1}{2}\|w\|^2 + \frac{C}{l}\sum_{i=1}^{l}\left(\xi_i + \xi_i^*\right) \right],$$

$$\text{s.t.} \begin{cases} \left((w \cdot x_i) + b\right) - y_i \leq \varepsilon + \xi_i, \\ y_i - \left((w \cdot x_i) + b\right) \leq \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0. \end{cases}$$

In the formula, $C > 0$ is a function of the balance between complexity and loss of errors. The partial derivative of the optimization problem (1) $\omega, b, \xi_i, \xi_i^*$ Lagrange function, with respect to the variable of 0: the available optimization problem (1) dual problem has as its solution a linear combination of kernel functions. It has the following form:

(2)
$$f(x) = \sum_{i=1}^{l}\left(\alpha_i - \alpha_i^*\right) k(x_i, x) + b^*.$$

The formula is the learning machine (SVM) of a regression function, that is, the training samples, where the expression coefficient can never be zero.

3. Particle swarm optimization and chaos optimization

The Particle Swarm Optimization (PSO) algorithm is proposed by scientists on the basis of the results of bird population studies. It depends on the biological population model proposed in 1995 [12], which is currently of widespread concern to scholars. The basic idea is that, among a group of flying birds, each bird can be regarded as a volume and quality of excluding particles. The particles have a certain speed in the search space flight, while they also, based on their experience and collective flight, continuously change their speed and position. It is assumed that the particle swarm in a dimension D space search will have an iterative formula, [13], such as

$$(3) \quad \begin{cases} v_{i,d}^{k+1} = \omega v_{i,d}^k + c_1 \cdot \mathrm{rand}\left(p_{i,d}^k - x_{i,d}^k\right) + c_2 \cdot \mathrm{rand}\left(p_{g,d}^k - x_{i,d}^k\right), \\ x_{i,d}^{(k+1)} = x_{i,d}^{(k)} + v_{i,d}^{(k+1)}. \end{cases}$$

In it, the position of the particle i is denoted by $x_i = (x_{i,1}, x_{i,2}, ..., x_{i,D})$, the speed is denoted by $v_i = (v_{i,1}, v_{i,2}, ..., v_{i,D})$, the best position is denoted as $p_g = (p_{g,1}, p_{g,2}, ..., p_{g,D})$, $i = 1, 2, ..., m$ represents the particle label, $d = 1, 2, ..., D$ represents the vector dimension, k represents the k iteration, and ω is the inertia weight, which is a constant; rand denotes a random number between 0 and 1, and C_1 and C_2 are constants between 0 and 2 and represent the acceleration factor. $x_{i,d} \in [x_{\min}, x_{\max}]$, $x_{\min}, x_{\max}$ depending on the objective function and the search space, is set to a different constant value. Simultaneously, to ensure that the particles can be carried out within the search space flight, where necessary, $v_{i,d}^k \in [-v_d^{\max}, v_d^{\max}]$, $v_d^{\max}$ represents the maximum distance between which the particles can fly, $v_d^{\max} = \bar{k} \cdot x_d^{\max}, 0.1 \le \bar{k} \le 0.5$.

Keeping (3) in the same position iterative formula, then improving the speed iterative formula, leads to the compression factor model of PSO algorithm convergence:

$$(4) \quad \begin{cases} v_{i,d}^{k+1} = \chi\left(v_{i,d}^k + \varphi_1 \cdot \mathrm{rand}\left(p_{i,d}^k - x_{i,d}^k\right) + \varphi_2 \cdot \mathrm{rand}\left(p_{g,d}^k - x_{i,d}^k\right)\right), \\ x_{i,d}^{(k+1)} = x_{i,d}^{(k)} + v_{i,d}^{(k+1)}, \end{cases}$$

where χ represents the convergence factor. If $\varphi = \varphi_1 + \varphi_2$ and $\varphi_1 = \varphi_2 = 2.05$, it follows that $\varphi = 4.1, \chi = 0.729$. Compared with the speed iteration, the formula has not been improved. This model is the same as the parameter value of $\omega = \chi = 0.729, c_1 = c_2 = \chi \cdot \varphi_1 = 1.49445$ before the improvement.

Since PSO primary particles are randomly selected, it is difficult to ensure that the selected particles are evenly distributed throughout the solution space. The method is therefore prone to local optimum conditions, greatly reducing the efficiency of the algorithm. Some particles may fly to outside the solution space, so this part of the particle boundary constraints is set, which defaults to the border in the solution space. Furthermore, when there is a part of the particles that may reach the boundary position [14], its speed is very small, so it has no capacity optimization. Considering the population of PSO algorithm, for a single particle, there is a lack of variation mechanism. It is therefore possible to consider the use of the characteristics of chaotic motion to improve the PSO.

Chaos is a nonlinear deterministic system. It may not be repeated through all states within the agreed space. In this paper it is applied to reset the initial and boundary particle position of PSO algorithm. Here the logistic is chosen to generate the chaos variable, which is mapped to

$$(5) \quad c_j^{r+1} = \mu c_j^r \left(1 - c_j^r\right), \quad r = 1, 2, ...,$$

where μ is a control parameter, c_j is chaotic variable and $0 \le c_j \le 1, j = 1, 2, \ldots, D$, D represents the particle position dimension, and r represents the serial number of the position of the particle which will be reset.

In the above formula, if $\mu = 4$, then in a completely chaotic state, at the same time, c_j between 0 and 1, to traverse. According to the initial value of the sensitivity of the chaotic motion, D has been randomly selected from different initial values between 0 and 1 (0, 0.25, 0.5, 0.75 and 1 cannot be selected) and then substituted into the formula. This gives the first variation of the particle position $C_1^1, \ldots, C_D^1$, the position of the second variation of the particles $C_1^2, \ldots, C_D^2$, and so on. The chaotic variable C_j^r is mapped into the corresponding optimization space variable x_j^r, as shown in the following formula:

(6)
$$x_j^r = x_j^{\min} + (x_j^{\max} - x_j^{\min})c_j^r.$$

Here ($x_j^{\min}, x_j^{\max}$) is the range of variation of the boundary values of x_j^r.

4. Improved CPSO algorithm and SVM parameter selection

The basic idea of the improved Chaotic Particle Swarm Optimization (CPSO) algorithm is that, in order to combine with the characteristics of the chaotic motion characteristic, the chaotic motion is used to select the initial particle position. Then, for particles in the search space to reach the border, the same chaos method is used to initialize. The algorithm can have the characteristics of randomness of a PSO algorithm, and can improve the population diversity and search characteristics [15].

These improvements of the CPSO algorithm are used to determine the parameters of the SVM. Among them, parameter ε supports the amount of vector sparsity control. Selecting the parameter will affect the accuracy of the estimated regression model. Depending on the reality of the situation, this can be set, so that all that is needed is the kernel width parameter a. The regularization coefficient c is then set to the optimization target amount. This leaves the particle as x_i, its dimensionality as $D = 2$, and the particle swarm location as $x_i = [a_i, c_i]$.

From the above description, information can be obtained for each particle, which corresponds to a coordinate position of SVM parameters. Due to the use of real-value coding, the only solution region can be random initialization. The k-fold cross-validation error is then regarded as a fitness function of the PSO [16], wherein j times give the generalization error by the following formula:

(7)
$$e_j = \sqrt{\frac{1}{l_j} \sum_{i=1}^{l_j} (\hat{f}(x_i) - f(x_i))^2}.$$

144

In the formula, $\hat{f}(x_i)$ is the output of the SVM prediction, $f(x_i)$ is the target output value, and l_j is the number of the j sample set, $j = 1, \ldots, k$.

The above process is repeated k times. Then after k times iterations, the generalization error obtained is subjected to averaging, resulting in a k-fold cross-validation error. The number of iterations, k, is selected as 5, then after the end of the iteration, a set of optimal parameters of the SVM can be obtained.

A CPSO algorithm flowchart using SVM parameter selection is shown in Fig. 1.

The specific steps are as follows:

Step 1. m is the number of the particles in a particle group; $x_{min}, x_{max}, v_{min}, v_{max}$ are particle position and velocity boundary values, M are the iterations; and v_i^0 is the random initialization particle velocity. Formula (6) is initialized with the initial position of the particle swarm chaos variable:

$$x_i^0 = \left\{ a_i^0, C_i^0 \right\}, i = 1, \ldots, m.$$

Step 2. The training samples are divided into five subsets that do not intersect $S_1, \ldots, S_5$ and $k = 0$.

Step 3. The current optimal parameter $x_i^k = \begin{bmatrix} a_i^k & C_i^k \end{bmatrix}$ is taken for training on SVM, while the 5-fold cross-validation error is an e^{-k} calculation.

Step 3.1. $j = 1$.

Step 3.2. S_j is taken as a test set; the other sub-sets as a training set.

Step 3.3. Formula (7): The generalization error e_j^k of S_j can be calculated, then $j = j+1$. The steps are followed up to Step 3.2 until $j = 5$.

Step 3.4. The five generalization errors obtained in the previous steps are averaged, by which the 5-fold cross-validation error e^{-k} can be drawn.

Step 4. e^{-k} represents the fitness value, when individuals and groups are under the best conditions: e^{-k}, p_i and p_g respectively of the position. $k = k+1$ is taken, according to Formula (4), the position $x_i^k = \begin{bmatrix} a_i^k & C_i^k \end{bmatrix}$ is arrived at, and the particle velocity v_i^k is updated.

Step 5. If some particles reach the border position x_{min}, x_{max} it must be re-initialized, with a predetermined speed in $\begin{bmatrix} v_{min}, v_{max} \end{bmatrix}$ by Equation (6).

Step 6. Returning to Step 3, iterations $k = M$ have been made, up to the end.

Step 7. From p_g the global optimal value $x = [a, C]$ is obtained; corresponding to the fitness value of the best fitness.
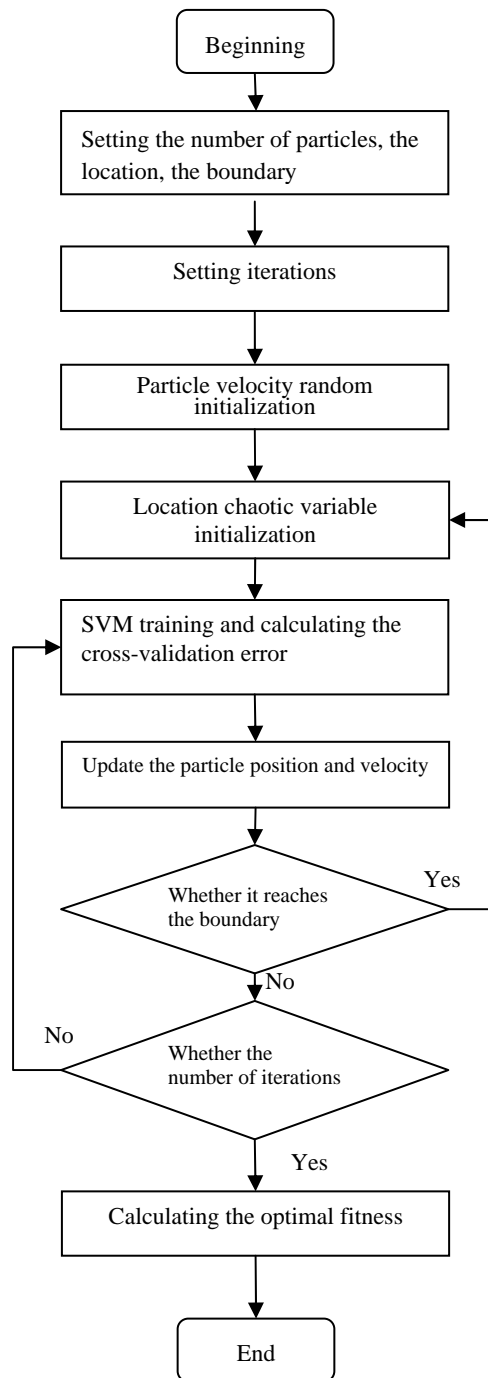
```
                    ┌─────────────┐
                    │  Beginning  │
                    └──────┬──────┘
                           │
                           ▼
         ┌──────────────────────────────────────┐
         │ Setting the number of particles, the │
         │ location, the boundary               │
         └──────────────────┬───────────────────┘
                            │
                            ▼
         ┌──────────────────────────────────────┐
         │          Setting iterations          │
         └──────────────────┬───────────────────┘
                            │
                            ▼
         ┌──────────────────────────────────────┐
         │     Particle velocity random         │
         │          initialization             │
         └──────────────────┬───────────────────┘
                            │
                            ▼
         ┌──────────────────────────────────────┐
         │     Location chaotic variable        │◄────┐
         │          initialization             │     │
         └──────────────────┬───────────────────┘     │
                            │                          │
                            ▼                          │
         ┌──────────────────────────────────────┐     │
         │  SVM training and calculating the    │     │
         │  cross-validation error              │     │
         └──────────────────┬───────────────────┘     │
                            │                          │
                            ▼                          │
         ┌──────────────────────────────────────┐     │
         │ Update the particle position and velocity │ │
         └──────────────────┬───────────────────┘     │
                            │                          │
                            ▼                          │
                    Whether it reaches   Yes           │
                    the boundary ───────────────────────┘
                            │ No
                            ▼
         No          Whether the
         ◄──────     number of iterations
                            │ Yes
                            ▼
         ┌──────────────────────────────────────┐
         │      Calculating the optimal fitness  │
         └──────────────────┬───────────────────┘
                            │
                            ▼
                    ┌─────────────┐
                    │     End     │
                    └─────────────┘
```

Fig. 1. The flow chart of the parameters selection of SVM with CPSO algorithm

146

5. Simulations and analysis

To verify the performance of the proposed algorithm, the following function is sampled to generate one-dimensional sample data [17]:

$$(8) \qquad f(x) = \begin{cases} -2.186x - 12.864, & -10 \le x \le -2, \\ 4.246x, & -2 \le x \le 0, \\ 10e^{-0.05x-0.5} \cdot \sin\left((0.03x + 0.7)x\right), & 0 \le x \le 10, \end{cases}$$

where x is as a starting point to -10, 0.005 gradually increases until the value is obtained as 10. So far, 400 pairs of samples will be obtained (with the one boundary point removed). The odds are training samples, the even numbered ones are test samples.

The computer simulation platform was configured for Intel Core i7 quad-core processor, with 4 GB of memory, 3.2 GHz frequency, on Windows7 operating system and development environment for Matlab12.0.

The SVM expression (2) in the kernel function is taken as a Gaussian kernel, and it uses a SMO algorithm [18] to question (1) dual problem solving and accuracy; calculated by the following formula:

$$(9) \qquad \delta = \left(\sum_{i=1}^{l} (y_i - f_i)^2 \right)^{\frac{1}{2}} \Big/ \left(\sum_{i=1}^{l} (y_i - \bar{y})^2 \right)^{\frac{1}{2}}, \quad \bar{y} = \frac{1}{l} \sum_{i=1}^{l} y_i.$$

If the insensitive loss coefficient is set to $\varepsilon = 0.05$, the population size is taken as $m = 20$, the width of nucleus a in [0.001, 1], the regularization coefficient C in [1, 15], the maximum speed limit type $v_d^{\max} = \bar{k}\, x_d^{\max}$, $\bar{k} = 0.2$, and the maximum number of iterations is taken as $M = 30$, according to the above parameters of PSO set by the CPSO algorithm section, this gives SVM parameters of iterative optimization.

As shown in Fig. 2, PSO is optimization of the individual particles' convergence process. Here it was used to improve the chaotic PSO. PSO optimization is repeated 10 times (10 times in average), as the iterations convergence is shown in Fig. 3. As it can be seen, the convergence of iterations will determine both the offline optimization method and the convergence time. When there are just a few iterations, the convergence time will be reduced accordingly.

For particles reaching the boundary, the CPSO algorithm uses the chaotic variable part to re-initialize the particle positions, in order to obtain good results. As shown in Fig. 3, compared with the PSO algorithm, this method reduces the number of iterations and makes it easy to select the global optimum. The diversity of the population and particle search properties have also been improved. The CPSO algorithm, using the SVM parameters selection experiment was repeated 3 times to obtain the best set of parameters and has the following adaption: $[\sigma, C] = [0.075, 9.75]$. This group, used as an optimization problem (1), shows the dual problem of the parameters; in defining the selection of parameters other than worth, the generalization error is smaller. It can be seen that the improved chaotic

particle swarm optimization algorithm can search for a group of optimal parameters of a SVM.
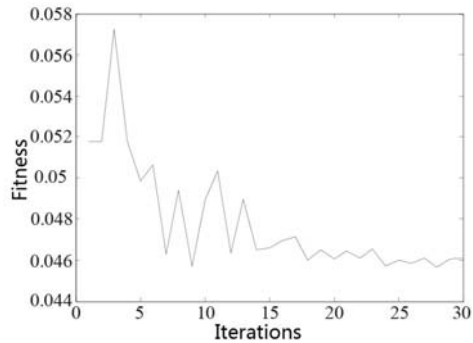

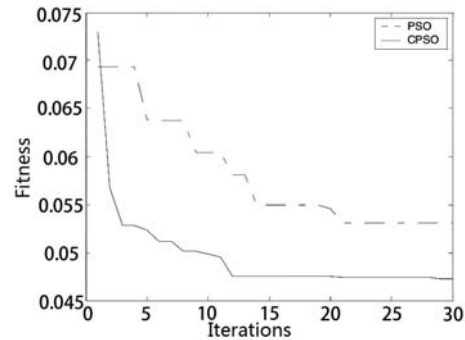Fig. 2. The convergence process of the single particle optimization

Fig. 3. Convergence of two PSO algorithms

6. Conclusion

SVMs are new and developing methods, based on statistical learning theory. An SVM training algorithm is a quadratic programming technique to solve a problem. In SVM solving process, firstly, the nuclear width parameter and the regularization coefficient of the optimization problem and other parameters need to be determined. The selection of the appropriate parameters will directly affect the performance of the SVM. In the course of SVM parameters selection, a CPSO algorithm is proposed to improve the traditional PSO algorithm with the sensitivity and characteristics of the initial value of the chaotic motion. The experimental verification shows that a set of optimal parameters of SVM can be quickly found with the improved CPSO optimization algorithm.

This SVM parameter selection method is a method of offline optimizing. Online SVM parameter optimization methods are chosen for further study of our work.

R e f e r e n c e s

1. V a p n i k, V. N. The Nature of Statistical Learning Theory. NY, Springer-Verlag, 1995.
2. V a p n i k, V. N. Estimation of Dependencies Based on Empirical Data. Berlin, Springer-Verlag, 1982.
3. G u o, Y a x i a n g, S h i f e i D i n g. Advances in Support Vector Machines. – Computer Science, Vol. **38**, 2011, No 2, pp. 14-17.
4. W u, Q., R. L a w, E. W u. A Hybrid-Forecasting Model Reducing Gaussian Noise Based on the Gaussian Support Vector Regression Machine and Chaotic Particle Swarm Optimization. – Information Sciences, Vol. **23**, 2013, No 8, pp. 96-110.
5. D i n g, G., L. W a n g, P. Y a n g et al. Diagnosis Model Based on Least Squares Support Vector Machine Optimized by Multi-Swarm Cooperative Chaos Particle Swarm Optimization and its Application. – Journal of Computers, Vol. **8**, 2013, No 4, pp. 975-982.

6. W a n g, H a i y a n, J i a n h u i L i, F e n g l e i Y a n g. Review of Theory and Support Vector Machine Algorithm. – Application of Computer, Vol. **31**, 2014, No 5, pp. 1281-1286.
7. C h e, J. X. Support Vector Regression Based on Optimal Training Subset and Adaptive Particle Swarm Optimization Algorithm. – Applied Soft Computing, Vol. **13**, 2013, No 8, pp. 3473-3481.
8. Z h a n g, L., J. W a n g. Optimizing Parameters of Support Vector Machines Using Team-Search-Based Particle Swarm Optimization. – Engineering Computations, Vol. **32**, 2015, No 5, pp. 119-127.
9. L i u, X i a n g d o n g, L u o b i n, Z h a o q u a n C h e n. Support Vector Machine Optimal Model Selection Research. – Computer Research and Development, Vol. **42**, 2015, No 4, pp. 576-581.
10. V a p n i k, V., E. L e v i n, Y. L e C u n. A Measuring the VC-Dimension of a Learning Machine. – Neural Computation, Vol. **6**, 1994, pp. 851-876.
11. K u a n g, F a n g j u n, W e i h o n g X u, S h i y a n g Z h a n g. A Optimization and Application of Improved Chaos Particle Swarm Mixed Kernel SVM Based on Parameters. – Application of Computer, Vol. **31**, 2014, No 3, pp. 671-674.
12. K e n n e d y, J., R. L. E b e r h a r t. Particle Swarm Optimization. – In: Proc. of IEEE International Conference on Neural Networks, 1995, pp. 1942-1948.
13. C h e n, H a i y i n g, M i n g h u i W u. Prediction of Chaotic Time Series Prediction Model Uniform Design Parameters. – Computer Science, Vol. **38**, 2011, No 2, pp. 14-17.
14. T a n g, q i, H o n g r u i W a n g, X i n y i X u. Hybrid Kernels SVM Hydrologic Time Series Model and Its Application. – Systems Engineering Theory and Practice, Vol. **34**, 2014, No 2, pp. 521-529.
15. Y o u, M, T. J i a n g. New Method for Target Identification in a Foliage Environment Using Selected Bispectra and Chaos Particle Swarm Optimisation-Based Support Vector Machine. – Signal Processing, Vol. **8**, 2014, No 1, pp. 76-84.
16. X i o n g, N a n, B a i f e n L i u. LSSVM Optimization Based on Adaptive Particle Swarm Network Traffic Online Prediction. – LSSVM Optimization Based on Adaptive Particle Swarm Network Traffic Online Prediction.
17. L u, L u, L i a n g l u n C h e n g. Improved Network Traffic Forecast Model Cuckoo Search Algorithm Optimization of SVM. – Computer Applications and Software, Vol. **32**, 2015, No 1, pp. 124-127.
18. C h e n, F., B. T a n g, T. S o n g et al. Multi-Fault Diagnosis Study on Roller Bearing Based on Multi-Kernel Support Vector Machine with Chaotic Particle Swarm Optimization. – Measurement, Vol. **47**, 2014, No 9, pp. 576-590.