

Research on the Recognition of the Nature of Pulmonary Nodules based on PSO

Yan-mei Wang¹, Jing Dai¹, Gengxin Sun² and Sheng Bin²

¹College of Information Science and Engineering, Shenyang Ligong University, No. 6, Nan Ping Zhong Road, Hun Nan Xin District, Shenyang, 110159, China

²Software Engineering College of Qingdao University, Qingdao, 266071, China
wangyysit@hotmail.com

Abstract

In order to increase the recognition rate of the CT image of benign or malignant pulmonary nodules, Support vector machine (SVM) was adopted to classify them. Meanwhile, Particle swarm optimization (PSO) algorithm is used to optimize parameters of the kernel function of SVM. Various optimization results were acquired through multiple methods such as consistent inertia weight, linear decreasing inertia weight, first increasing and then decreasing inertia weight and non-linear decreasing inertia weight. As was proved by experiments, recognition rates of these methods in training set were the same. The method with consistent inertia weight had a short optimizing time, but recognition rate in test set was low. The remaining methods had a long optimizing time while recognition effects were better in test set.

Keywords: CT image, PSO, pulmonary nodules SVM, inertia weight

1. Introduction

Pulmonary nodules are single or multiple nodules which have clear boundary and are opaque in iconography. Their diameters are less than 3 centimeters and are totally surrounded by gas lung tissues. They are not accompanied with pulmonary atelectasis, hila of pulmonary swell and pleural effusion. Pulmonary nodule is one of the most common lesions of lung which can be either benign or malignant [1]. Among nodules which are excised because of indefinite diagnosis, approximately half are benign [2]. Even malignant nodules are difficult to be found positive by sputum cytology and fiberoptic bronchoscopy on account of the lack of accumulation in bronchus [3]. Therefore, the significance of early diagnosis between benign or malignant lesions of pulmonary nodules is shown, which is vital to patients' recovery. Researching with classification and recognition system on lung CT image of benign and malignant nodules based on SVM, the problem of how to raise recognition rate is urgent to be solved.

2. Method of Parameter Optimization

Penalty factor c and kernel function of SVM are most obvious elements in classification and recognition system on lung CT image of benign and malignant nodules. Penalty factor c is used to control the degree of punishment of wrong sample, to control the balance of the sample deviation and generalization. Radial basis kernel function (RBF) is suitable for various conditions such as low-dimensional, high-dimensional, small sample, large sample and has a relatively wide convergence domain, which makes it an ideal reference function for classification [4]. RBF kernel function can be described as formula (1).

$$K(x, z) = \exp\left(-\frac{\|x - z\|^2}{2g^2}\right), g > 0 \quad (1)$$

So the key of improving the recognition rate is to select the penalty factor c , parameters g and optimize. There are currently three common methods for SVM parameter optimizing problems: grid search method (GSM), genetic algorithm (GA) and PSO. GSM is to divide feasible intervals of each parameter into a series of grids and then find an optimal parameter by traversing all the points in grids. After sequentially calculating respective parameters of each grid point, a corresponding target value is gained from calculation and then compared and picked over for a best one. As a result, the optimal target value and its corresponding parameters in a certain interval are acquired. This method has the advantage of simple algorithm and easy realization. In an interval which is large enough when pace is small enough, it is guaranteed that the search result is basically the globally optimal solution. Due to the fact that most parameters in grids have fairly low classification accuracy which is relatively high only in a small interval, it will waste plenty of time trying to traversing all the parameter sets in the grids.

GA and PSO are both heuristic algorithms. Heuristic algorithms have plenty of advantages over non heuristic GSM that there is no need to traverse all to parameters to find a globally optimal solution as well as quick convergence when data size is huge, they can also search for optimal parameters in a bigger numeral space. GA is an effective way to solve optimization problems based on the principle that individuals which is not adaptable to the environment will be obsoleted while individuals with abilities of adaption can have the next generation which makes the whole species more and more suitable for the environment during the process of surviving. However, GA is not effective in solving specific issues, triggering the problems of precocity and poor convergence ability.

PSO is put forward by James Kennedy and Russell Eberhart corporately in 1995 whose basic idea is inspired by their former study on group behavior of birds using biological population models [5]. Concepts of ‘group’ and ‘evolution’ are adopted, through assistance and competition among individuals, each individual is regarded as a particle without weight or volume in a dimensional space which fly towards the location with the best fitness in a certain speed. Global searching ability and local searching ability are both possessed to realize the search of optimal solutions in a complicated space. PSO has a simple concept which is easy to realize and it has a wide range of application fields.

3. Standard Particle Swarm Optimization Algorithm

Each potential solution of optimization problems can be assumed as a particle without weight or volume in a k dimensional object searching space, every particle flies in the searching space at a certain speed. $X_i = (x_{i1}, x_{i2}, \dots, x_{ik})$ represents the current location of its particle in the k dimensional space. $V_i = (v_{i1}, v_{i2}, \dots, v_{ik})$ represents the flying speed of its particle. Then the optimal solution is gained through iteration. During each time of iteration, particle is updated by tracking two extreme values p_i and p_g . $P_i = (p_{i1}, p_{i2}, \dots, p_{ik})$ represents previous optimal solutions searched by its particle, $P_g = (p_{g1}, p_{g2}, \dots, p_{gk})$ represents optimal solutions searched by all particles. The evolutionary process of PSO can be described as formula (2) and (3).

$$v_{ik}(t+1) = v_{ik}(t) + c_1 r_{1k}(t)(p_{ik}(t) - x_{ik}(t)) + c_2 r_{2k}(t)(p_{gk}(t) - x_{gk}(t)) \quad (2)$$

$$x_{ik}(t+1) = x_{ik}(t) + v_{ik}(t+1) \quad (3)$$

Subscript “ k ” represents the k th dimension of particles, “ i ” represents particle i , t represents the t th generation, c_1 and c_2 are acceleration constants, c_1 is the weight coefficient of particles when tracking their own previous optimal solutions which represents particles’ recognition of themselves, c_2 is the weight coefficient of particles

when tracking optimal solutions of the swarm, which represents particles' recognition of the whole swarm. Without c_1 , particles only have social experience but no self-practice which accelerates convergence speed but may well fall to a local optimum. Without c_2 , particles only have self-abilities but lack communication among particles which makes it hard to find optimal solutions. When c_1 is chosen to be bigger, it is hard for particles to converge in searching area. When c_2 is chosen to be bigger, the search will end too early.

4. PSO with Inertia Weight

In order to improve convergence performance of basic PSO algorithm, inertia weight is introduced into speed evolution equation as formula (4).

$$v_{ik}(t+1) = wv_{ik}(t) + c_1r_{1k}(t)(p_{ik}(t) - x_{ik}(t)) + c_2r_{2k}(t)(p_{gk}(t) - x_{gk}(t)) \quad (4)$$

Where w is called inertia weight, as a result, basic PSO algorithm is a special circumstance when inertia weight $w = 1$. Inertia weight keeps movement inertia of particles which makes them have the tendency to extend searching space as well as the ability to explore new areas. The quantity of inertia weight demonstrates the extent to which the previous speed of particles can be reserved. With regard to inertia weight w , as is proved big inertia weight helps the algorithm to avoid falling into local optimum at the beginning of the algorithm, little inertia weight accelerates convergence speed which makes the convergence steadier and avoid oscillation in the late stage of the algorithm. Therefore, dynamically decrease inertia weight w can make the algorithm stabler.

5. PSO Algorithms with Changing Inertia Weight

5.1. Inertia Weight Linear Decrease

Inertia weight is similar to annealing temperature. Bigger w has a better ability of global convergence which helps the algorithm to avoid falling into local optimum while smaller w has a better ability of local convergence which makes the convergence steadier and avoid oscillation. Therefore, as iterations increase, inertia weight is supposed to decrease constantly in order to make PSO algorithm has a better ability of global convergence at the beginning as well as a better ability of local convergence later[6-7]. Inertia weight w should meet formula (5).

$$w = w_{\max} - \frac{i}{I_m}(w_{\max} - w_{\min}) \quad (5)$$

w_{\max} is the maximum value of w , w_{\min} is the minimum value of w , linearly decreasing from 1.4 to 0.4 [8]. i is the current number of iterations, I_m is maximum number of iterations, so , the inertia weight w as a function of the number of iterations, linear decrease from w_{\max} to w_{\min} , as in shown in Figure 1.

5.2. Inertia Weight Linear Differential Decrease

The shortcoming of inertia weight linear decreasing is that at the beginning of the algorithm, if optimal area is not found, as the decrease of w , the ability of local searching is enhanced, and the algorithm may fall into local optimum. In order to improve the limitation of inertia weight linear decreasing, the method of inertia weight linear differential decreasing is adopted [9], as shown in Figure 2. At the beginning of the algorithm, the tendency of w to decrease is slow, global searching ability is strong which is beneficial for finding good optimization of seed, and avoiding falling into local optimum. Later in the algorithm, the speed of w to decrease is accelerated, once suitable seeds are found at the beginning, the convergence speed is accelerated to find optimal

solutions. The inertia weight w can be expressed by formula (6) and (7).

$$\frac{dw}{di} = \frac{2(w_{\max} - w_{\min})}{I_m} \times i \quad (6)$$

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{I_m^2} \times i^2 \quad (7)$$

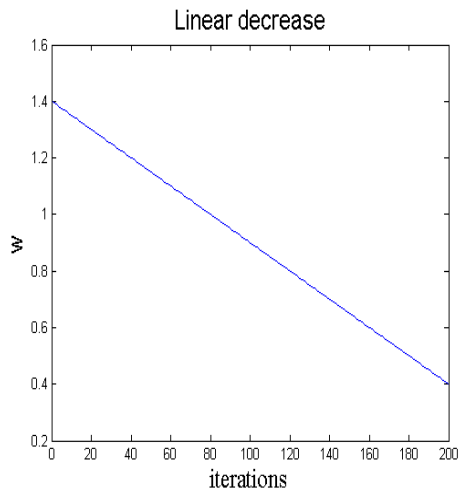


Figure1. The w is a Linear Decrease

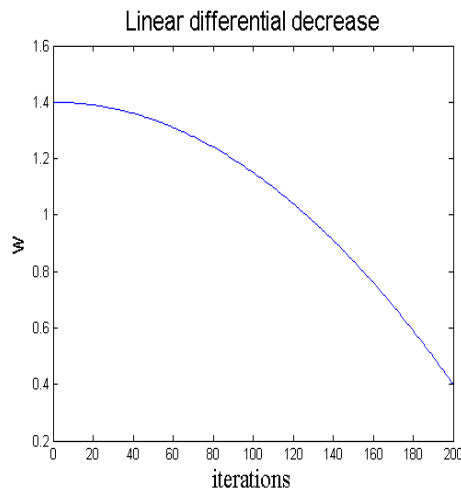


Figure2. The w is a Linear Differential Decrease

5.3. Inertia Weight First Increase then Decrease

Inertia weight first increase then decrease can get a bigger w at the beginning of optimizing and have a bigger range to search for optimal values which avoids local optimum. Bigger convergence speed is also obtained. During later time of optimizing, the ability of local searching is better [10]. The curve of inertia weight first increase then decrease is shown as figure 3. The inertia weight w can be expressed by formula (8).

$$w = \begin{cases} 1 \times \frac{i}{I_m} + 0.4 & 0 \leq \frac{i}{I_m} \leq 0.5 \\ -1 \times \frac{i}{I_m} + 1.4 & 0.5 < \frac{i}{I_m} \leq 1 \end{cases} \quad (8)$$

5.4. Inertia Weight Nonlinear Decrease

Inertia weight linear decrease can hardly jump out once fall into local extremum. In order to change the limitation of linear methods, nonlinear decrease strategies to which decreasing index λ and iteration threshold I are introduced can be adopted [11]. The inertia weight w can be expressed by formula (9).

$$w = \begin{cases} w = w_{\max} - \left(\frac{i-1}{I_m-1} \right)^\lambda (w_{\max} - w_{\min}) & i \leq I_0 \\ w = w_{\max} - \left(\frac{I_0-1}{I_m-1} \right)^\lambda (w_{\max} - w_{\min}) & i > I_0 \end{cases} \quad (9)$$

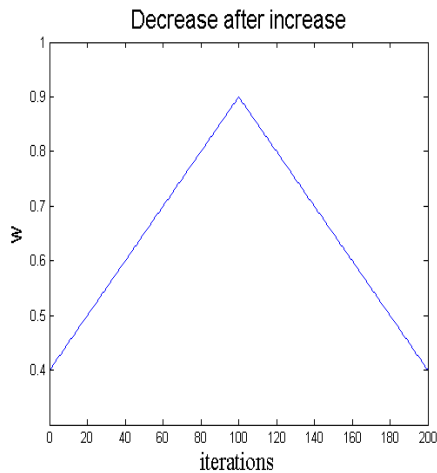


Figure 3. The w is a Decrease After Increase

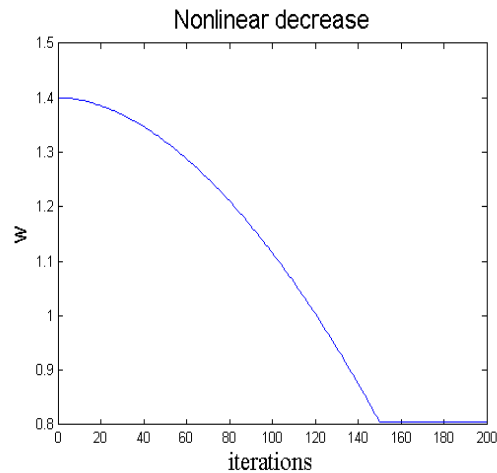


Figure 4. The w is a Nonlinear Decrease

At the beginning of iteration, w is bigger which is beneficial for searching in the whole space. Decreasing index λ is introduced during the process of iteration. The w nonlinear decreases as I increases, which is beneficial for jumping out of local optimum. When iterative number reaches I_0 , w equals w_0 and maintains the value until the end of iteration. The curve of inertia weight nonlinear decreasing is shown as Figure 4. As is proved by experiments, when I_0 equals 150, the effect is better when $\lambda=1.8$.

6. Experiment and Simulation

Based on matlab as an experimental platform, CT images after analysis and testing of pathology were selected. Each 70 samples of both benign and malignant nodules were separated from CT images and classified to be training set and test set. First 40 samples of benign nodules were selected to be training samples when last 30 samples to be test samples. The same selection is done to malignant nodules.

The group scale of PSO algorithm was 20 particles and evolutionary generation was 200, $c_1=1.5$, $c_2=1.7$, inertia weight was selected respectively to be a certain number 1, linear decrease, linear differential decrease, first increase then decrease and nonlinear decrease to do experiments [12].

When $w=1$, the result was $c=8.3272$, $g=13.5796$, recognition time was 7.2626s when best recognition rate of nodules in training set, the recognition rate in test set was 87.7% used the c and g above. The result of parameter optimization was shown as figure 5. A large inertia weight was not easy to fall into local optimum at the beginning of training, small inertia weight can made the convergence fast, made convergence more stable, and not oscillate in training later stage. Inertia weight remains unchanged, neither conducive to global optimization at the beginning nor conducive to local optimization in the later stage. The recognition rate close to 98% when w iterated to 20, but the particle missed the global optimum because of unchanging, then the recognition rate changed from 88% to 96%. Therefore, the recognition rate was low in the test set, generalization ability was poor.

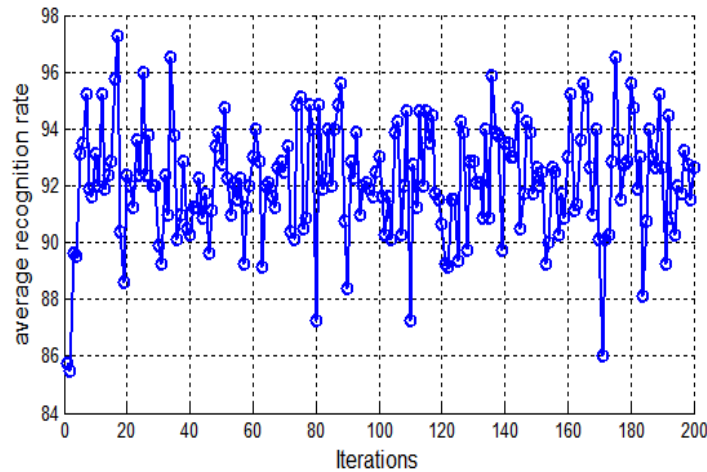


Figure 5. Recognition Rate of the Training Set when $w=1$

The result was $c=5.1183$, $g=6.7911$, recognition time was 11.2036s when w linear decreases, the result of parameter optimization was shown as figure 6. Because inertia weight linear decreases, w already below 1 when w iterated to 80 shown as figure 1, ability of global convergence was reduce, the global optimum was not found, so the recognition rate in training set had been below 94%. But the recognition rate in test set was 90%, the difference of recognition rate between training set and test set was reduce, generalization ability had improved.

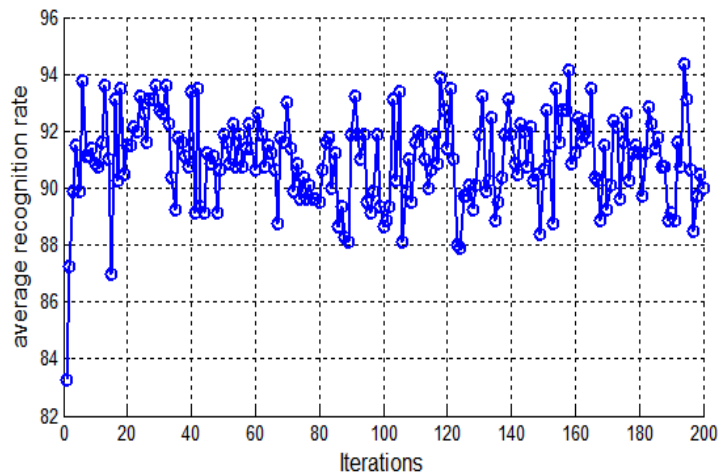


Figure 6. Recognition Rate of the Training Set when w Linear Decreases

The result was $c=1.9522$, $g=0.19256$, recognition time was 13.3885s when w linear differential decreases, the result of parameter optimization was shown as Figure 7. Because inertia weight linear differential decreases, w reduces to 1 when w iterated to 120 shown as Figure 2, maintain a certain global search capability, the recognition rate keep between 89% to 94%, the recognition rate in test set was 93.3%, the recognition rate of training set and test set was close, had a strong generalization ability.

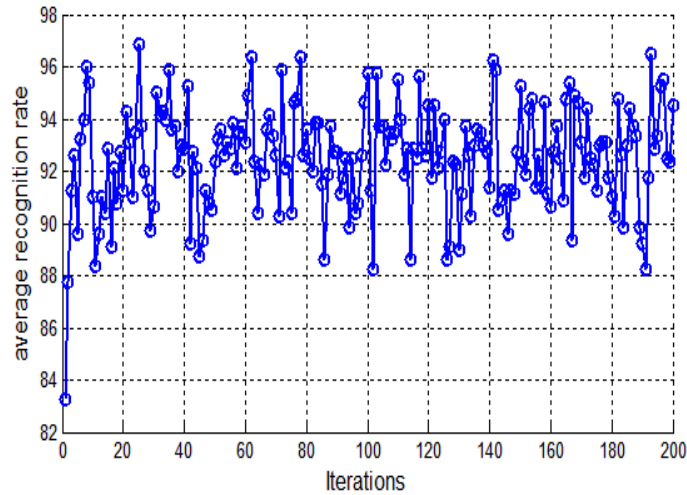


Figure 7. Recognition Rate of the Training Set when w Linear Differential Decreases

The result was $c=4.9674$, $g=0.037192$, recognition time was 12.812s when w linear differential decreases, the result of parameter optimization was shown as Figure 8. Inertia weight decreases after increases, w increased to 0.6 yet when iterated to 40 shown as Figure 3, global search capability was not strong, and recognition rate was low. The recognition rate was gradually increased, reaching up to 96% with increasing w . The local search capability enhancement with the decrease of w , the recognition rate remained at 93%. The recognition rate of test set used the best c and g was 90%, the generalization ability was not strong.

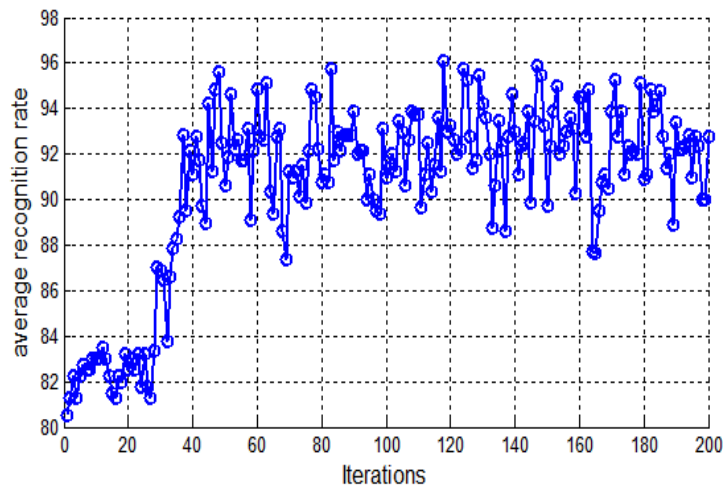


Figure 8. Recognition Rate of the Training Set when w Decreases after Increases

The result was $c=0.70165$, $g=0.44536$, recognition time was 13.2807s when w nonlinear decreases, the result of parameter optimization was shown as figure 9. Inertia weight nonlinear decreases, w decreased to 1.1 when iterated to 100 shown as figure4, had a strong global search capability, the recognition rate in training set was above 98%. w rapidly reduced to the minimum 0.8 when iterated to 150, the local search capability was enhanced, and the best average recognition rate in training set was still above 98%. At mean time, the recognition rate in test set reached 96.7%. The recognition rate in training set and test set were all improved in spite of a long recognition time, so the

generalization ability used the best parameter when w nonlinear decreases was very strong.

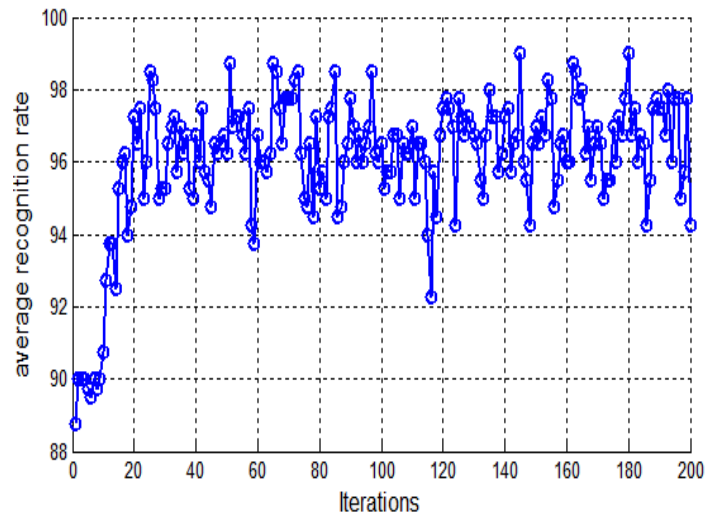


Figure 9. Recognition Rate of the Training Set when w Nonlinear Decreases

When inertia weight changes differently, the result of parameter optimization and optimization time and recognition rate in test set were as Table 1, shows.

Table 1. Parameter Optimization Results

w	1	linear decreases	linear differential decreases	decreases after increases	nonlinear decreases
c	8.3272	5.1183	1.9522	4.9674	0.70165
g	13.5796	6.7911	0.19256	0.037192	0.44536
$t(s)$	7.2626	11.2063	13.3885	12.812	13.2807
recognition rate	86.7%	90%	93.3%	90%	96.7%

As can be seen from Table 1, the different optimization parameters w obtained different c and g . the recognition rate in test set was higher when used smaller c ; the recognition rate was gradually increased with g increase, the recognition rate decreased when g increased a certain extent.

7. Conclusion

In the algorithm of PSO, inertia weight is the key parameter in balancing the abilities of global searching and local searching. According to experiments, in the recognitions of lung benign nodules and malignant nodules, the recognition rate in test set was higher when used smaller penalty factor c , the g of RBF too big and too small were not obtained higher recognition rate. The recognition rate reached the highest when g was certain middle value. The recognition rate in test set used the parameter obtained from fixed inertia weight parameter optimization was the lowest, the recognition rate in test set used the parameter obtained from inertia weight nonlinear decreases parameter optimization was the highest.

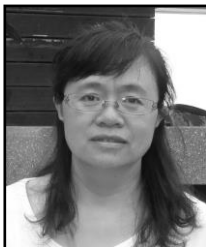
Acknowledgments

This work was supported by the scientific research fund of Liaoning provincial education department under project No. L2013084.

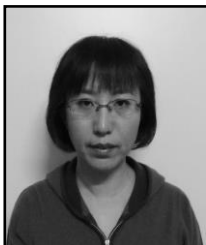
References

- [1] Y. Cui, D. Q. Ma, "The basic methods and application of computer-aided detection and diagnosis of lung nodule in CT", Chinese Journal of Medical Imaging Technology, vol. 23, no. 3, (2007), pp 469-472
- [2] L. Liu, W. Y. Liu and C. Y. Chu, "Fast classification of benign and malignant solitary pulmonary nodules in CT image", Optics and Precision Engineering, vol. 17, no. 8, (2009), pp. 2060-2067.
- [3] C. C. Zhang, C. D. Yang and M. Cai, "To Evaluate the Diagnostic Value of the CT Direct and Indirect Features of the Solitary Pulmonary Nodule in Distinguishing It Benign or Malignant", Medical Innovation of China, vol. 10, no. 25, (2013), pp. 90-92
- [4] Sh. L. Lin, Zh. Liu, "Parameter selection in SVM with RBF kernel function", Journal of Zhejiang University of Technology, vol. 35, no. 2, (2007), pp 163-167.
- [5] J. C. Zeng, J. Jie and Z. H. Cui, "Particle Swarm Optimization", Science Press, Beijing (2004)
- [6] Y. F. Chen, T. Yun and Zhou Yu, "Texture Image Segmentation Based on PSO Optimising SVM", Computer Applications and Software, vol. 31, no. 4, (2014), pp. 214-218.
- [7] Y. H. Shi and R. C. Eberhart, "A modified particle swarm optimizer", IEEE International Conference on Evolutionary Computation, Anchorage Alaska, America, (1998) May 4-9, pp. 69-73.
- [8] Y. H. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization", Proceeding of Congress on Evolutionary Computation, IEEE Service Center, Piscataway, New Jersey, America, (1999), pp. 1945-1949.
- [9] J. X. Hu and J. C. Zheng, "Selection on Inertia Weight of Particle Swarm Optimization", Computer Engineering, vol. 33, no. 11, (2007), pp. 193-195.
- [10] H. M. Cui and Q. B. Zhu, "Convergence Analysis and Parameter Selection in Particle Swarm Optimization", Computer Engineering and Application, vol. 43, no. 23, (2007), pp. 89-91
- [11] L. Wang and X. K. Wang, "Modified Particle Swarm Optimizer Using Non-Linear Inertia Weight", Computer Engineering and Application, vol. 43, no.4, (2007), pp. 47-48.
- [12] N. Zhao, S. K. Jia and J. Wang, "Survey on the Parameters of Particle Swarm Optimization", Machinery & Electronics, no.11, (2013), pp.1-6.

Authors



Yanmei Wang, received her B.E. in Industry Automation from Northeastern University in 1991, and the M.S. degree in Biomedical Engineering from the Northeastern University, Shenyang, China, in 2008. Currently she serves an Associate Professor in School of Automation and Electrical Engineering, Shenyang Ligong University. Her researches cover signal analysis and pattern recognition.



Jing Dai, received the B.E. in Electronic Engineering from Shenyang University of Technology in 1991, and the M.S. and the Ph.D. degrees in Detection Technology and Automatic Equipment from the Northeastern University, Shenyang, China, in 1999 and 2005, respectively. Currently she serves an Associate Professor in School of Automation and Electrical Engineering, Shenyang Ligong University, Shenyang, China. Her research interests include digital signal processing and wireless communication.



Gengxin Sun, received his Ph.D. degree in Computer Science from Qingdao University, China in 2013. He is currently an Associate Professor in the college of Software Engineering at Qingdao University. His main research interests include embedded system, operating system, complex networks, web information retrieval and data mining.



Sheng Bin, received her Ph.D. degree in Computer Science from Shandong University of Science and Technology, China in 2009. She is currently a lecturer in the college of Software Engineering at Qingdao University, China. Her main research interests include embedded system, operating system, complex networks, cloud computing and data mining.