An improved PSO algorithm coupling with prior information for classification of large scale dataset

Juanjuan Tu[1,a,*], Wenlan Zhou[2,b], HongmeiLi[1,c]

[1]School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu 212003, China

[2]School of electronic information, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu212003, China

[a]ecsitu@126.com,[b]wenlan_zhou@163.com,[c]liredmei@yeah.net

Keywords: Particle Swarm Optimization,Prior Information, Classification

Abstract. An improved particle swarm optimization(PSO) algorithm coupling with prior information for classification of large scale dataset is proposed in this paper. The prior information derived from the data set is used to determine the initialposition of the particles. In the new algorithm, neural network is first trained by improved PSO and then by back-propagation (BP). The prior information narrows the search space and guides the movement direction of the particles, so the convergence rate and the generalization performance are improved. Experimental results demonstrate that the new algorithm is more effective than traditional methods.

1 Introduction

The early classic classification methods are statistical classification. Its forms of expression are based on some solid formulas.In contrast with the statistical model, the neural network is a "model free" machine. Itmakes the output value to converge to the correct target valuesby adjustingweight coefficient and shows good classification performance in unsupervised learning conditions.

The generalization ability of neural network directly affect the performance of classifier. The PSO algorithm with global optimization ability can be used to train the neural network[1-4]. It can overcome some shortcomings of traditional algorithm.Many researchers have proposed some improved PSOalgorithm to train neural network[5-10]. But the existing literature seldom research how to select the effective sample set and obtain the prior information to guide the design of neural network. But the prior information is necessary for improving the search efficiency of PSO and guiding the search direction[11].

In addition, the classic parameter statistical methods generally require known mathematical model structure.The training samples are used to estimate parameters and its distribution is necessary to be known, so a large number of training samples is inevitable.However, the conventional algorithms cannot solve such problems in pattern classificationwhen the number of samples is largebecause of its high time complexity.Therefore, aiming to the classification of the large sample dataset, the urgent problem is how to enhance the training speed based on ensuring the accuracy of learning.

This paper proposes a neural network algorithm optimized by PSO couplingwith prior information. It is used in the classification of large sample data set.

2 The neural network algorithm optimized by PSO coupling with prior information

2.1 The extract of prior information from large sample data set

Biashas become the focus in many data mining methods.The Bias method can integrate the prior information and the data information of observed sample, and then generate the posterior distribution of the unknown variables. Several researchers combined the posterior distribution with artificial neural network to construct a new model that is suitable for large sample data analysis[12-13].

Inspired by this, we combined the Bias method with neural network algorithm optimized by PSO. The Bias method is used to extract the prior information that is coupled into the PSO, and then the neural network model for classification of large sample data set can be constructed.

The initialposition of swarm in the standard PSO algorithm is randomly generated, resulting in the uncertainty of the initial search space range[14]. The prior information implied in the sample data can reduce the search space of particle swarm.Aiming at the large sample data set, first the posterior probability of the various weights namely the position of particlesis calculated under the given conditions. And thentheposition of particles is initialized according to the calculated probability distribution to reduce the modification magnitude of each particle in the process of learning and improve the learning speed.

Next the detailed calculation process of prior information is introduced.

Assume that the weights of neural network namely the position of particles is $W = (w_1, w_2, ..., w_n)$ and the sample data is $D = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$,where n and m respectively express the number of weight and sample data. The conditional probability of W in the sample data is:

$$P(W / D) = \frac{P(D / W)P(W)}{P(D)} , \qquad (1)$$

Where, $P(W)$ represents the prior distribution of W, $P(D/W)$ represents the likelihood function and $P(D) = \sum_i P(D / w_i)P(w_i) = 1$ is the normalization factor.

The initial value of W should be smaller in order to avoid work in the saturated part of incentive function in the beginning. But, if W is too small, all the incentive functions are almost in a linear part lead to convergence speed descend. W generallyobey the exponential distribution:

$$P(W) = \frac{1}{Z_W(\alpha)} \exp(-\alpha E_W) , \qquad (2)$$

Where $Z_W(\alpha)$ represents a normalization factor to guarantee $\int P(W)dW = 1$ and $Z_W = \int \exp(-\alpha E_W)dW$. α isa parameter used to control the distribution of weight. E_W is a kind of error function and its general expression is:

$$E_W = \frac{1}{2}\|W\|^2 = \frac{1}{2}\sum_{i=1}^{n} w_i^2 。 \qquad (3)$$

The formula (4) can be derived by formula (2) and (3).

$$P(W) = \frac{1}{Z_W(\alpha)} \exp(-\frac{\alpha}{2}\sum_{i=1}^{n} w_i^2) 。 \qquad (4)$$

The likelihood function $P(D/W)$ can also be expressed asthe exponential form.

$$P(D / W) = \frac{1}{Z_D(\beta)} \exp(-\beta E_D) , \qquad (5)$$

Where, $Z_D(\beta) = \int \exp(-\beta E_D)dD$,and the error energy E_D is the error between the target output and the actual output of network. Normally, $E_D = \frac{1}{2}\sum_{i=1}^{m} e_i^2$,where e_i is the error between the target output and the actual output of network.

After the formula (2) and (5) is substituted into formula (1), the new expression form of is:

$$\begin{aligned} P(W / D) &= \frac{1}{Z_D(\beta)} \exp(-\beta E_D) \cdot \frac{1}{Z_W(\alpha)} \exp(-\alpha E_W) \\ &= \frac{1}{Z_M} \exp(-\beta E_D - \alpha E_W) \\ &= \frac{1}{Z_M} \exp(-M(W)) , \end{aligned} \qquad (6)$$

$$M(W) = \beta E_D + \alpha E_W , \qquad (7)$$

$$Z_M(\alpha + \beta) = \int \exp(-\beta E_D - \alpha E_W)dD , \qquad (8)$$

Where, $\alpha = \frac{m}{2E_W}$ and $\beta = \frac{n}{2E_D}$.

2.2 The PIPSO-BPNN algorithm

In this paper, we use PSO coupling with prior information and BP to train FNN (PIPSO-BPNN).

The detailed steps of are:

Step 1: Generate training sample dataset $D = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$;

Step 2: Define the number of neurons in each layer and target error of FNN;

Step 3: Define parameters of PSO: swarm size, particle dimension, inertia weight, acceleration coefficient and maximum number of iteration etc.;

Step 4: Initialize the velocity and position of each particle randomly;

Step 5: Adjust the position of each particle according to formula (6), namely use $P(W / D)$ to replace the position of particles;

Step 6: Compute the fitness of particles according to the training error;

Step 7: Compare the fitness and individual optimal extreme of each particle.If the former is better than the latter, it would be set as new individual optimal extreme;

Step 8: If the best individual optimal extreme is better than the current global optimal extreme, record the particle's number ,adjust its position according to formula (6) and then set it as new global optimal extreme;

Step 9: Update the velocity and position of each particle;

Step 10: Check that whether the stop condition of PSO is met. If it’s true, stop search and record global optimal extreme and then go to step 11; else return step6;

Step 11: Decode the global optimal extreme to the parameter of the BP neural network, including weights and thresholds;

Step 12:Continue to train neural network by BP algorithm until the target error is obtained.

3 Experimental results

In order to demonstrate the good performance of the new approach, a lot of experiments have been conducted in MATLAB 7.0. The simulations for classification of four algorithms which are BPNN, PSO-BPNN, APSO(adaptive particle swarm)-BPNN and PIPSO-BPNN have been carried out.

Two two-class datasets (pima&wdbc) and one three-class dataset (balance) were chose for experiments (shown in Table 1).

Table 1Datasets used in this study

Dataset	Samples	attributes	Categories
pima	768	8	2
wdbc	569	30	2
balance	625	4	3

The other parameters are shown in this section.The number of particle swarm $n = 50$, the inertia weight $\omega = 0.9$ and $c_1 = c_2 = 2$. The range of linear outputweight is [-1, 1], target error e=10-5 and the maximum iteration number of neural network is 3000.

The selection of training sample and testing sample use three fold cross-validation. Before classification,the dimension of features in dataset wdbc was reduced by signal-noise ratio. The relationship between the dimension of features and the error rate of classification is shown in Fig. 1. From the figure we can find that the error rate of PIPSO-BPNN is the least in the vast majority of dimensions. In order to obtain the best classification results, the dimension of features was reduced to 8.

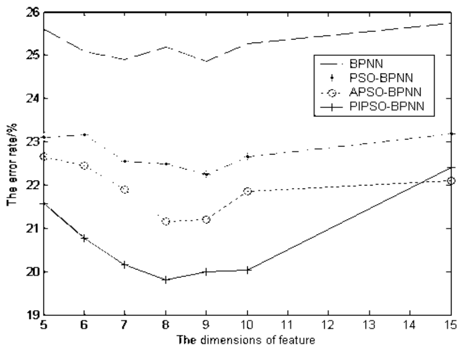


Fig. 1 The relationship between the dimension offeatures and the error rate of classification (wdbc)

The average accuracy of classification is shown in Table 2:

Table 2 The average accuracy of classification

Dataset	BPNN	PSO-BPNN	APSO-BPNN	PIPSO-BPNN
pima	95.67%	96.35%	97.13%	97.86%
wdbc	75.14%	77.52%	78.85%	80.19%
balance	85.62%	87.69%	88.24%	89.01%

The iteration number of four algorithms is shown inTable3- Table5.

Table 3 The iteration number of PSO and BP in four algorithms (pima)

Algorithms	Iteration number of PSO	Iteration umber of BP
BPNN	/	3000
PSO-BPNN	300	2500
APSO-BPNN	270	2500
PIPSO-BPNN	70	2000

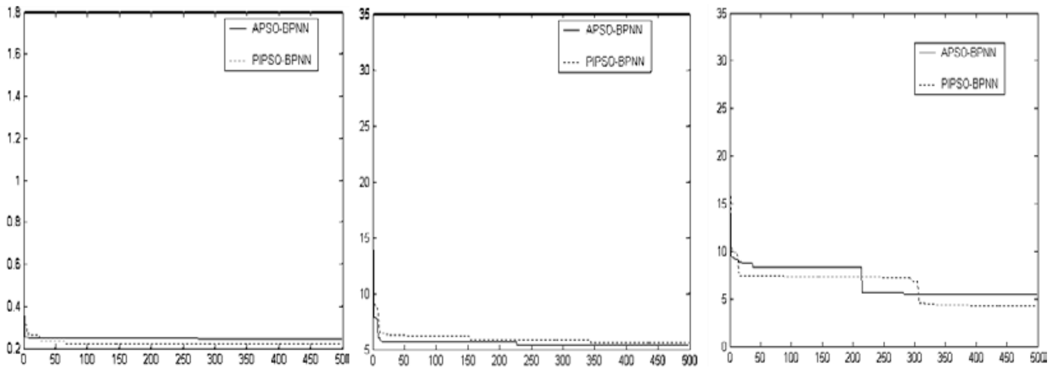
Table 4 The iteration number of PSO and BP in four algorithms (wdbc)

Algorithms	Iteration number of PSO	Iteration umber of BP
BPNN	/	3000
PSO-BPNN	350	2500
APSO-BPNN	350	2500
PIPSO-BPNN	230	2000

Table 5 The iteration number of PSO and BP in four algorithms (balance)

Algorithms	Iteration number of PSO	Iteration umber of BP
BPNN	/	3000
PSO-BPNN	350	3000
APSO-BPNN	300	2500
PIPSO-BPNN	270	2500

The updating curves of the global optimum along with the iteration of PSO are shown in Fig. 2.



(a) Pima (b) Wdbc (c) Balance

Fig. 2The updating curve of the global optimum along with the iteration of PSO

From the tables and figures described above, we observe:

1) From Table 2, we can find that the average accuracy of PIPSO-BPNN is the highest based on three datasets. It proves that the proposed method is more efficient than three other methods. The prior information helps PSO to acquire more efficient results in the training process of neural network.

2) Table 3- Table 5 illustrate that the convergence speed of PIPSO-BBPNN is higher than that of two other algorithms. The prior information reduces the iteration number of RBF and PSO, because it helps the proposed algorithm to avoid unnecessary search.

3) Fig. 2 shows the updating speed of the global optimum along with the iteration of PSO. It explains that why the iteration number of PSO in PIPSO-BPNN is less than that of APSO-BPNN wherever on which dataset. They prove that the prior information help particle swarms to converge more quickly.

The above results prove that the prior information abstracted fromthe large scale sample data by using Bayesian method can improve the search efficiency of PSO effectively.Because PSO can search the global optimal value more accurately, the classification accuracy of the neural network model has been improved.

4 Conclusions

In this paper, an improved PIPSO-BPNN algorithm is proposed to solve the classification problem of large scale sample data. First, we analyze the feature of the large scale sample data, and then the prior information is derived from the sample data according to Bias theory which is coupled into PSO. Last, the improved PSO algorithm is used to train neural network.The experimental results show that PSO coupling with prior information has better convergence rate and can improve the accuracy of classification.

Acknowledgement

This research was supported by the Natural Science Foundationfor Colleges and Universities of Jiangsu Province (No. 14KJD520002) and the Dr. Scientific Research Foundation of Jiangsu University of Science and Technology.

References

[1] Fvanden Bergh F., Engelbrecht A. P. CooperativeLearning in Neural Networks Using Particle SwarmOptimizers [J]. South Africa Computer J , 2000, 26(11) : 84-90.

[2] GiesD., YahyaR.. Particle swarm optimization for reconfigurable phase differentiated array design [J]. Microwave and Optical Technology Letters (S0895-2477), 2003, 38(3): 168-175.

[3] LinY., ChangW., HsiehJ.. A particle swarm optimization approach to nonlinear rational filter modeling [J]. Expert Systems with Applications (S0957-4174), 2008, 34(2): 1194-1199.

[4] ChauK.W.. Application of a PSO-based neural network in analysis of outcomes of construction

claims [J]. Automation in Construction, (S0926-5805) 2007, 16(5): 642-646.
[5] HeS., WuQ. H., WenJ. Y., et al. A particle swarm optimizer with passivecongregation[J]. Biosystems, 2004, 78: 135-147.
[6] RatnaweeraA., HalgamugeS. K., WatsonH.C.. Self-organizing hierarchical particleswarm optimizer with time-varying acceleration coefficients[J]. IEEE Ttans. OnEvolut. Compute, 2004, 8: 240-255.
[7] MonsonC. K., SeppiK.D.. The Kalman swarm-A new approach to particle motionin swarm optimization[J]. Lecture Notes in Computer Science, 2004, 3102:140-150.
[8] TreleaI.C.. The particle swarm optimization algorithm: convergence analysis andparameter selection[J]. Information Processing Letters, 2003, 85: 317-325.
[9] VandenF. Bergh. An Analysis of Particle Swarm Optimizer[D]. PhD Dissertation. South Africa: Department of Computer Science, University of Pretoria, 2002.
[10] Wang Z. W. , Durst G. L. , Eberhart R. C.. Particle Swarm Optimization and Neural Network Application for QSAR [C].Proc. IEEE Int. Conf.on Parallel and Distributed Processing Symposium, New Mexico, 2004.
[11] HanF., LingQ..A new approach for function approximation incorporating adaptive particle swarm optimization and a priori information[J]. Applied Mathematics and Computation, 2008,205: 792-798.
[12] Mackay D. J.C.. A practical Bayesian framework for back-propagation networks [J] . Neural Computation, 1992, 4(3) : 415.
[13] Thodberg H.H.. A review of Bayesian neural networks with an application to near infrared spectroscopy [J] . IEEE Trans. NeuralNetworks, 1996, 7(1) : 56.
[14] KennedyJ., and EberhartR. C.. Particle swarm optimization: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, 1995[C]:1942-1948.