

Construction and Application of PSO-SVM Model for Personal Credit Scoring

Ming-hui Jiang and Xu-chuan Yuan

School of Management, Harbin Institute of Technology, 150001 Harbin, China
{jiangmh@cast.cn, yuanxuchuan}@126.com

Abstract. The parameters of support vector machine (SVM) are crucial to the model's classification performance. Aiming at the randomness of selecting the parameters in SVM, this paper constructed a PSO-SVM model by using particle swarm optimization (PSO) to search the parameters of SVM. The model was used for personal credit scoring in commercial banks and particles' fitness function was used to control the type II error which costs huger loss to commercial banks. Compared with BP NN, the application results indicate that PSO-SVM gets higher classification accuracy with lower type II error rate and the model shows stronger robustness, which presents more applicable for commercial banks to control personal credit risks.

Keywords: support vector machine, particle swarm optimization, personal credit scoring.

1 Introduction

Support vector machine (SVM) [1] developed as a new machine learning method, compared with artificial neural networks, is more applicable for small samples classification based on statistic learning theory (SLT). SVM has widely used in text recognition, face recognition, image compression and so on for the model has no rigid requirements on variables' distribution and it performs well on generalization based on structural risk minimization (SRM). Through applying kernel functions to map the sample data from a low-dimensional feature space to a high-dimensional space, SVM constructs an optimal separating hyper-plane to classify the data. However, the parameters of SVM which are crucial to the model's classification performance are selected randomly or by cross validation which is time consuming. So searching optimal parameters is crucial to use SVM successfully.

The particle swarm optimization [2] (PSO) is a parallel evolutionary computation technique which has been widely used in function approximation, pattern recognition as well as neural network's training. In this paper, we make a combination of PSO and SVM to construct a PSO-SVM model by using PSO to search SVM's parameters, and then we use the model for personal credit scoring. In the end, we make a comparison between PSO-SVM and BP NN on classification results to examine the model's classification performance.

2 Construction and Application of PSO-SVM

2.1 Samples and Variables

In this paper, we use the personal credit data from one commercial bank in Shenzhen. The input variables figured with x_i and one output variable y listed in Table 1.

Table 1. Inputs and outputs

variable	index	value
x_1	Education	elementary 1; middling 2; advanced 3
x_2	Monthly Income	1: 1000~5000; 2: 5001~10000; 3: 10001~15000; 4: 15001~20000; 5: 20001~30000
x_3	Organization Character	national department 1; scientific education cultural and healthy department 2; trade and business 3; post and communication 4; financial and insurance 5; social service 6; supply of water, electricity and gas 7; industry 8; real estate 9; other 10
x_4	Career	manager 1; technique 2; officer 3; jobless 4; other 5
x_5	Spouse	yes 1; no 2
x_6	Loan Amount	1:6000~10000; 2:10001~50000; 3:50001~100000; 4:100001~300000; 5:30001~500000; 6: 500000~600000
x_7	Time limit	1: 24~36; 2: 37~49; 3: 50~60
x_8	Return mode	actual value 1; corpus 2
x_9	Surety	pledge 1; impawn 2; other 3
x_{10}	Age	1: 20~30; 2: 31~40; 3: 41~50; 4: 51~60
y	Default or not	yes -1; no 1

As the collectivity has many samples, which are complex and different, we use equal proportion allocation in stratified random sampling here. First, we divide the collectivity with the value y into two groups and then we extract about 500 samples respectively, thus to make the proportion of $y=-1$ and $y=1$ to be nearly 1:1. Finally, we get 1057 samples for the model, with 528 training samples composed of 257 default ones and 271 non-default ones and 529 testing samples composed of 248 default ones and 281 non-default ones.

2.2 Construction and Application of PSO-SVM

In practice of credit scoring, there exist two types of errors: type I error which mistakes the good applicants as bad ones and refuses to offer the loan and type II error which supplies loan to applicants with bad credit. The loss caused by type II error is huger, so we should concern more about type II error than type I error.

According to the principle of SVM [1], we know the parameters need to be selected include the kernel function's parameter and the penalty parameter C . In order to control the type II error, we set the penalty parameters of type I and type II errors as C_1 and C_2 respectively with the relation $C_2=K*C_1$, $K>=1$. Among four commonly used kernel functions, RBF kernel function has been used widely, for it has been proved with strong non-linear mapping and there is only one parameter in the kernel.

So we choose RBF function as the kernel. So the parameters in PSO-SVM need to be optimized are σ^2 , C_1 and K . We set the searching range as [1, 100].

In order to get a better performance, we use a modified PSO algorithm with a inertial weight w [3], that is

$$v_{ij}(t+1) = w \cdot v_{ij}(t) + c_1 \cdot r_1 \cdot (p_j(t) - x_{ij}(t)) + c_2 \cdot r_2 \cdot (g_j(t) - x_{ij}(t)) \tag{1}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{2}$$

The inertial weight w is linearly decreasing [4] according to Formula (3), that is

$$w_t = w_{\max} - \frac{w_{\max} - w_{\min}}{t_{\max}} * t \tag{3}$$

Here we set w_{\max} as 0.9 and w_{\min} as 0.4 and the maximum iteration as 100 to terminate the algorithm. We set the population size m as 20, the acceleration constants c_1 and c_2 as 2 [5], v_{\max} as 0.4. In order to control the type II error efficiently, we set PSO to maximize the particle's fitness function defined as Formula (4):

$$f = M \left(1 - \frac{n_1}{m_1} - k * \frac{n_2}{m_2} \right) \tag{4}$$

where, n_1 and n_2 represent the number of type I error and type II error respectively; m_1 and m_2 represent the number of non-default samples and default samples respectively. So n_1/m_1 and n_2/m_2 represent type I and type II error rates respectively; k is a variable and here we set $k > 1$. Through the experiments with different numbers, we choose $k = 2$ in the final program. M is set as 100 used to make the change of the fitness observable.

Using MATLAB to run the program of PSO-SVM and after 100 iterations we get the optimal parameters as $\sigma^2 = 85.351$, $C_1 = 10.685$, $K = 4.677$ and $C_2 = K * C_1 = 49.497$. The classification results of PSO-SVM are shown in Table 2.

In order to make a comparison, here we construct a forward three-layer BP NN model with 10 neurons in input layer, 1 neuron in output layer and 7 neurons in hidden layer which is commonly used in personal credit scoring for the same samples. Besides, the transfer functions in hidden and output layers are *tansig* and *logsig*. We set the epoch as 1000 to terminate the training and the performance function as *MSE* (Mean Square Error). We choose an improved BP algorithm with an adaptive learning rate and a momentum to train the network. The classification results of BP NN are shown in Table 2 with the critical value 0.5.

Table 2. Classification results

Model	Training samples			Testing samples		
	Type I error	Type II error	Accuracy	Type I error	Type II error	Accuracy
PSO-SVM	24(8.86%)	2(0.78%)	95.08%	26(9.25%)	5(2.02%)	94.14%
BP NN	6(2.21%)	2(0.78%)	98.86%	27(9.61%)	12(4.84%)	92.63%

3 Result Analysis

From Table 2, we can see on training samples, the classification accuracy of PSO-SVM is lower than that of BP NN, but on testing samples, PSO-SVM gets a higher accuracy than BP NN does. From the other aspect, we can see the difference on classification accuracy of BP NN is $98.86\%-92.63\%=6.23\%$, however, PSO-SVM is only $95.08\%-94.14\%=0.94\%$. This indicates PSO-SVM gets a better performance on robustness than BP NN. In a dynamic environment of personal credit, a model with strong robustness is more significant for commercial banks to keep away from credit risks and expand the personal credit market. From this aspect, we can say PSO-SVM is more applicable than BP NN. Besides, from the results of the two types of errors, we can see PSO-SVM classifies the default samples with lower type II error rate both on training and testing samples, especially on testing samples on which the result of PSO-SVM is much lower than that of BP NN. The results indicate that the fitness function of PSO is effective to control the type II error which is more significant for commercial banks to keep away from personal credit risks. The reason is that in order to increase the fitness of the particle in PSO, SVM enhances the punishment on type II error to achieve the goal.

4 Conclusions

This paper makes a combination of PSO and SVM to construct a PSO-SVM model and used the model for personal credit scoring. We get the conclusions from the empirical results: first, using PSO's global search to optimize the parameters of SVM, we can get the optimal parameters of SVM and get over the randomness; second, through particle's fitness function in PSO, we can control the type II error which brings huger loss, that is more significant for commercial banks to keep away from credit risks; third, compared with BP NN, the difference of accuracies on training samples and testing samples of PSO-SVM is smaller, that indicates the model gets a stronger robustness. In a word, we can use the PSO-SVM model for personal credit scoring to get a good result.

References

1. Vapnik V N.: The nature of statistical learning theory. Springer-Verlag, New York (1995)
2. Kennedy J, Eberhart R C.: Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks, Perth, Australia (1995) 1942-1948
3. Shi Y H, Eberhart R C.: A Modified Particle Swarm Optimizer. IEEE International Conference on Evolutionary Computation, Anchorage, Alaska (1998) 69-73
4. Shi Y H, Eberhart R C.: Empirical study of particle swarm optimization. Proceedings of the 1999 Congress on Evolutionary Computation (1999) 1945-1950
5. Shi Y H, Eberhart R C.: Parameter Selection in Particle Swarm Optimization. Proceedings of the Seventh Annual Conf. on Evolutionary Programming (1998) 591-601