

SVM Multi-Classification Optimization Research based on Multi-Chromosome Genetic Algorithm

Ren Qian, Yun Wu*, Xun Duan, Guangqian Kong, and Huiyun Long

College of Computer Science and Technology, Guizhou University, Guiyang, 550025, China

Abstract

Regarding SVM multi-classification problem, optimizing the parameters of SVM has become the key problem to improve the performance of the SVM multi-classification algorithm. In order to solve this problem, multi-chromosome genetic algorithm is proposed in this paper and used to optimize these parameters. In the SVM multi-classification decision tree, the algorithm constructs a chromosome for SVM parameter of each node and improves the corresponding rules of crossover and mutation in the genetic algorithm. The improved genetic algorithm optimizes the parameters of SVM in all nodes in the SVM multi-classification decision tree. The experimental results show that the SVM multi-classification decision tree algorithm using the multi-chromosome genetic algorithm has higher classification quality, compared with the traditional multi-SVM multi-classification algorithm.

Keywords: multi-classification; SVM; optimization; multi-chromosome; GA

(Submitted on January 3, 2018; Revised on February 14, 2018; Accepted on March 25, 2018)

© 2018 Totem Publisher, Inc. All rights reserved.

1. Introduction

Support Vector Machine (SVM) is a machine learning method based on statistics theory. Since birth, due to its outstanding performance on small sample and nonlinear classification problem, it is widely used in image recognition [13], text classification [16,17], and other fields [5].

Traditional SVM can only deal with dichotomous problems, but in many classification problems, the sample shall be divided into multiple categories. In order to apply the SVM to multi-classification problems, a one-time solution method and reconstruction method have been proposed, and the one-time solution method is not well applied due to the large number of variables and high computational complexity. However, the reconstruction method decomposes the multi-classification problem into a series of dichotomies, which is simple and easy to implement, so it is more widely used than the one-time solution method. Due to the fixed structure of the multi-classification algorithm, the adaptive adjustment cannot be made according to the specific problem, so the performance of the algorithm cannot be improved. Therefore, this paper proposes an improved GA to optimize the multi-classification algorithm DT-SVM.

2. Analysis of the Existing Algorithms

The Support Vector Machine was first proposed in the 1990s. After continuous development, there have been many new optimization schemes proposed [3,8], for example, Novakovic J et al. [9] put forward the use of the PCA algorithm to denoise data before using SVM classification, so that the SVM classification results are more accurate. For real-time updated data

* Corresponding author.

E-mail address: wuyun_v@126.com

systems, Xue Y et al. [15] proposed to set a weight parameter in SVM to determine the degree of dependence of the system on the non-updated data. The improved SVM is more suitable for dynamic update systems. However, in the actual classification problem, it often needs to be divided into multiple categories, and as a solution to the dichotomous problem, SVM needs to be improved accordingly.

In the current research situation, the multi-classification of SVM has been applied in many fields [1,4,10,18], and there are two kinds of SVM multi-classification ideas [2]. One is to compute all the classification decision functions and solve multiple classification problems at the same time; however, the optimization process of such method is very complex, the computation is huge, and the implementation is difficult, so it has not been widely applied. The other is to decompose the multi-classification problem into multiple dichotomous problems, which are also divided into three categories: OAO (one-against-one), OAA (one-against-all), and DTSVM. OAO uses voting strategies to conduct multiple classifications, and OAA trains SVM through one category to other categories to separate each category one-by-one. However, there will be unclassified regions in the final classification results of the two methods, so it cannot be separated into any category. DTSVM uses a two-forked tree structure and separates one of them on each node by using the OAA method until all categories are separated. The decision tree of DT-SVM is fast and avoids the unclassified region.

The structure of the DT-SVM decision tree is fixed, as shown in Figure 1. Structure of DT-SVM, and it cannot be adapted to the specific problem, which restricts the performance of DTSVM. In the aspect of optimizing SVM, there are a lot of researches at home and abroad [6,7,11], of which the use of GA to optimize the SVM is the most extensive [14]. In order to optimize the DTSVM algorithm and make the DT-SVM adaptively adjusted according to the problem, a multi-chromosome genetic algorithm is proposed, which is used to optimize the DT-SVM. Multi-chromosome genetic algorithm constructs multiple chromosomes for multiple SVM parameters and improves the pairing mutation of the algorithm in genetic manipulation. With the help of GA self-adaptation, the best solution is obtained, and the performance of DT-SVM algorithm is improved.

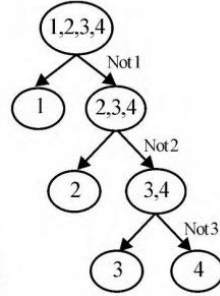


Figure 1. Structure of DT-SVM

3. Model Definition

3.1. SVM and Its Multi-Classification Algorithm

SVM is developed from the linearly separable classification problem, and its basic idea is to find a classification line between the two categories. The line can not only distinguish the two categories, but it can also make the classification interval between the line and the two categories largest, as shown in the figure. The general sample sets can be described by Equation (1):

$$Z = \{(x_i, y_i) | x_i \in R^n, y_i \in \{+1, -1\}, i = 1, 2, \dots, m\} \quad (1)$$

Where i represents the spatial dimension, x_i represents the vector, y_i represents the category that the vector x_i belongs to. Classification hyper plane can be described by Equation (2):

$$(w \cdot x) + b = 0 \quad (2)$$

Classification interval is $2/\|w\|$. In order to maximize the classification interval, the problem is converted to the following convex quadratic programming problem, and it can be described by Equation (3):

$$\min \|w\|^2/2, s.t. y_i(w \cdot x_i + b) - 1 \geq 0, i = 1, 2, \dots, l. \quad (3)$$

When the classification sample does not support hard partitioning, it is necessary to introduce a slack variable to the sample. The slack variable is $\varepsilon_i \geq 0 (i = 1, 2, \dots, l)$. Including penalty parameter C, the above formula can be described below by Equation (4):

$$\min \|w\|^2/2 + c \sum_{i=1}^n \varepsilon_i, s.t. y_i(w \cdot x_i + b) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0, i = 1, 2, \dots, l. \quad (4)$$

Finally, we get the classification function in Equation (5):

$$f(x) = \text{sgn} \left[\sum_{i=1}^l y_i a_i (x \cdot x_i) + b \right] \quad (5)$$

Where a_i is the Lagrangian coefficient, and the value of the non-support vector a_i is 0. Since the non-support vector represents the majority of the entire sample, it is only necessary to calculate a small number of support vectors to complete the classification.

In solving nonlinear classification, SVM solves the problem of "dimensionality disaster" by introducing a kernel function. The kernel function allows the inner product operation to be carried out in the original space instead of the high dimensional space after the mapping, which simplifies the inner product calculation in the mapping space. After using a kernel function $K(x_i \cdot x_j)$, the classification function can be described by Equation (6):

$$f(x) = \text{sgn} \left[\sum_{i=1}^l y_i a_i K(x \cdot x_i) + b \right] \quad (6)$$

Most of the kernel functions currently used are the polynomial kernel in Equation (7):

$$K(x, x_i) = [(x, x_i) + 1]^d \quad (7)$$

and Gaussian kernel function in Equation (8):

$$K(x_1 \cdot x_2) = \exp \left(-\frac{\|x_1 - x_2\|^2}{g} \right) \quad (8)$$

The Gaussian kernel has a very high flexibility in adjusting the parameter g . Therefore, this paper chooses the Gaussian kernel as the kernel function of the SVM. In the Gaussian kernel function, if the parameter g is very small, all vectors will become the support vector. However, this will cause the overfitting problem; if the parameter g is made great, then the weighting of function in high dimensional feature will be very small, resulting in SVM inability of classification in high-dimensional feature space.

With regards to the multi-classification problem of SVM, the most widely used solution is to transform the SVM multi-classification problem into multiple dichotomous problems. In this paper, the DT-SVM is adopted, by which it can not only avoid the problem of undetermined data categories, but its stable tree structure is also conducive to the optimization of SVM parameters.

3.2. Genetic Algorithm

Genetic algorithms mimic the process of biological evolution in nature, which mainly includes four steps:

1) Construction of Chromosomes

Genetic algorithm uses specific data structure to simulate creature's chromosomes in the nature, and the algorithm defines different data structures according to the problem. Currently, the most used are the digital coding structure and binary coding structure. The digital coding structure uses different numbers to represent different variables directly. Binary encoding structure uses a specific length of binary bit representing each variable. Because this coding structure is very similar to the biological gene structure, it is more suitable for the operation of mutation, evolution, etc., so this paper chooses the binary coding structure as the data structure of the chromosome.

2) The Generation of the Initial Population

The initial population is the initial solutions of genetic algorithm in search space. The number of general initial population to be selected is 10-200 individuals. The algorithm chooses different initial population sizes according to the scale of the problem. The larger the size of the initial population, the more difficult it is to fall into the locally optimal solution, but this will lead to the optimal solution not being easy to dominate the direction of evolution and allow for huge amounts of computation. Therefore, the convergence rate of the algorithm will be slow and even result in inefficient calculations. The smaller the initial population, the faster the convergence speeds of the algorithm, and easier to fall into the locally optimal solution.

3) Selection of Fitness Evaluation Function

The fitness directly affects the probability of the individual to be selected in genetic manipulation, and the higher the individual's fitness, the more likely for the individual to be selected for genetic manipulation. The fitness function continues to be used to evaluate the fitness of its offspring. The individual with ultimate fitness is the optimal solution. Different fitness evaluation functions are selected according to different problems.

4) The Formulation of Genetic Rules

Genetic operators consist of three main operators: selection operators, crossover operators and mutation operators.

i. Selection operator

The selection operator is a step that reflects the survival of the individual in the natural world: through the relevant rules, select the best fitness individuals from all individuals to reproduce.

ii. Crossover operator

In order to produce individuals with higher fitness, the algorithm will find two individuals to pair in the population. After pairing, the crossover operator will make gene exchange on the chromosome of individuals according to the corresponding rules. The current crossover rules mainly include single-point crossover, multi-point crossover, cycle crossover, and so on.

iii. Mutation operator

Gene mutation will occur occasionally on the individuals, and the mutation could result in the high fitness of the individual. The mutation operator simulates the mutation process: change operation to be done in the genes according to certain rules, and the change operation is different according to the data structure of different chromosome codes. In binary codes, the operation is an inverse operation.

3.3. Application of Improved GA to DTSVM Optimization

A decision tree is used to transform the original multi-classification problem into a series of dichotomous problems. In each decision tree, dichotomy is carried out to every non-leaf node, and the classification results are generated at the leaf nodes so as to achieve multi-classification. This paper divides the data into ten categories, so nine classification nodes are needed and nine SVMs are established. The kernel function of these SVMs uses the Gaussian kernel function.

In order to optimize the kernel parameter g and penalty factor C in the nine SVMs by genetic algorithm, the genetic algorithm is improved in this paper.

The first is the selection of the fitness function. In the decision tree structure of this paper, the error is located in the upper level, so it has a bigger influence on the performance of the algorithm: if the classification of a certain sample point is misclassified, the error classification of this sample point will always affect the classification of all subsequent layers. We use the error weighting function as the fitness function. For N classification problems, we first reset the first level of error weight to $n-1$, and then decrease it layer by layer. Therefore, the error weight at the bottom level is 1. The fitness function is described by Equation (9):

$$Fitness(x) = \sum_{i=1}^n (n-i)x_i / \left(total + \sum_{i=1}^n (n-i)x_i \right) \quad (9)$$

In each individual, a chromosome is created for each SVM, and nine chromosomes are created. The gene segments of chromosomes are divided into three parts: the first part is defined as the SVM number, which represents nine SVMs respectively and includes four binary codes. The second part is the kernel parameter g , which includes twenty bit binary codes. The third part is the penalty factor C , which also includes twenty bit binary codes.

Because the initial population is a prerequisite for the genetic algorithm, it is equivalent to a certain number of initial solutions in the search space, and its size can be changed according to the scale of the problem. The size of the initial population is generally 10-200 individuals. The initial population size of this paper is set to 20 individuals.

At the crossover stage, in order to generate crossovers among the nine chromosomes of two individuals, this paper adds a pairing procedure between chromosomes. The pairing rule of pairing steps is that two chromosomes are paired with the same number of chromosomes. The crossover rule of chromosomes is that one gene point is randomly selected as a crossover in the gene segment g and the gene segment c .

The mutation rules are as follows: in gene segment g and gene segment c , the mutation probability of each gene is 0.1. For SVM numbered gene segments, the probability of mutation is 0.009, because the mutation of the gene segment will lead to a larger mutation in the individual. In the algorithm, it is shown that the SVM exchange between nodes in SVM multi-classification decision tree model is caused. This large mutation has the probability of producing individuals with high fitness, which reduces the genetic generation and helps find the globally optimal solution. However, if the probability of this mutation is too large, it will lead to a slow convergence of the genetic algorithm and may even result in the inefficiency of the algorithm. In the mutation step, if the mutation result is converted into decimals and is greater than the number of DTSVM nodes, then the modulus of the number and the node number is adopted. In order to maintain an SVM in each node in the decision tree, the value corresponding to the chromosome marker after the mutation is modified to the value of the chromosome marker before the mutation.

Above all, the steps of using the Multi-Chromosome Genetic Algorithm to optimize the multi-classification SVM parameters are as follows:

- 1) Establish the SVM multi-classification decision tree DT-SVM.
- 2) Construct the number of chromosomes according to the number of SVM in DT-SVM.
- 3) Generate 20 initial populations, with an upper limit of genetic generation of 200.
- 4) Calculate the fitness of each individual using formula (9), check whether the fitness reaches the threshold and whether the genetic generation reaches 200, and judge whether the termination condition is reached. If the termination condition is reached, jump to step (9).
- 5) Calculate the probability of each individual being selected. Individuals with high fitness of high probability selection will enter the crossover stage.
- 6) Apart from the numbered gene segment, select a cross point at random in the gene segment g and the gene segment c for crossover operation.
- 7) The number of numbered genes segment is mutated at a probability of 0.009, and the other gene segments are mutated at a probability of 0.1.
- 8) Genetic generation plus 1, jump to step (4).
- 9) Select the optimal solution and obtain the optimal classification by DT-SVM.

4. Experiment

4.1. Introduction of Experimental Environment and Data Set

The computer configuration of this experiment is I5 processor, 2GB memory, 500GB hard disk, and Linux operation system. The data set is artificial characters open data set on UCI, which contains 6000 samples and ten categories. The experimental programming language is python.

4.2. Experiment and discussion

Nine SVMs are established in the experiment to produce nine chromosomes. There are three gene segments on each chromosome. The first segment is the chromosomal marker gene segment, which indicates the sequential number of the chromosome, that is, the SVM node number in the representative multi-classification decision tree. The second segment is the gene segment of the Gaussian kernel function parameter g , and the third segment is the gene segment of the penalty factor c in SVM.

The initial population of this experiment consists of twenty initial individuals. The SVM number gene segments in individual chromosomes are generated in order from one to nine. In order to ensure the completeness of parameters optimization (all candidate problem solution in space can be expressed in GA in the search space), we divide gene segment g and gene segment c into five sections, and the initial twenty individuals are divided into five equal parts. One part of the initial individuals is randomly selected from each part of the five parts of gene segment g and gene segment c . Values of initial individuals selected on each gene of the gene segment are one, and other gene values are zero other than that of the SVM gene number.

In the experiment, 4000 records are selected as training samples, and the remaining 2000 records are taken as test samples. 400, 1000, 1600, 2200, 2800, 3400, and 4000 records are selected from training samples randomly for the experiment. First, the classification accuracy of DT-SVM optimized by the Multi-Chromosome Genetic Algorithm (MCGADT-SVM) and proposed in this paper is compared with that of the traditional DT-SVM multi-classification algorithm and OAO multi-classification algorithm. Ten experiments are done for each number of training samples, and the average value of the accuracy was taken as a result of the experiment. The detail results can be seen in Figure 2. Comparison of classification precision

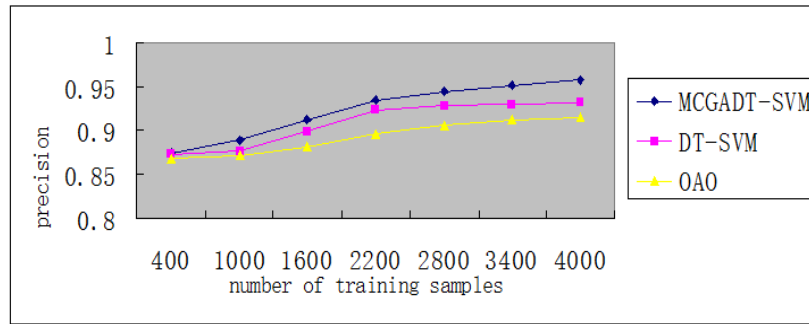


Figure 2. Comparison of classification precision

From the results, we can see that the classification precision of MCGADT-SVM has obvious advantages over traditional DT-SVM and OAO. This advantage increases with the increase of training samples, because with the increase of training samples, the advantage of GA in heuristic search is more obvious. In the process of genetic iteration, the algorithm adaptively selects the best parameter combination and eliminates the parameter combination with poor effect.

In the improved multi-chromosome genetic algorithm in this paper, the chromosome construction methods and pairing, crossover, and mutation rules are reformulated. The experimental results show that these rules make GA better optimize the SVM parameters of each node and achieve the desired result.

Then, we test and observe the effect of the mutation probability of the sequence number gene segment on the performance of the algorithm. Because the mutation of the gene segment will lead to a larger mutation, this larger mutation will benefit the algorithm by not only allowing it to reduce the number of iterations, but also by allowing it to jump out of the locally optimal solution to find the globally optimal solution. However, if the probability of this mutation is too large, it is difficult for the algorithm to converge. The detail results can be seen in Figures 3 and 4.

From the experimental results, we can see that when the mutation probability of the gene segment is 0.009, the algorithm has the fewest number of iterations. After reaching 0.017, the algorithm reaches the largest genetic generation and cannot converge, and the accuracy of classification is also decreases abruptly.

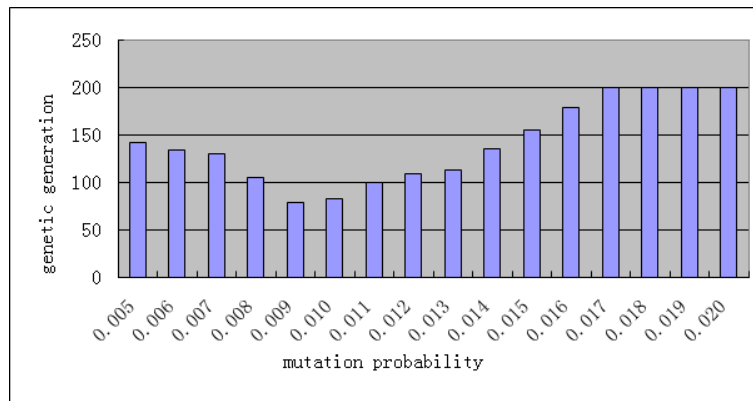


Figure 3. Comparison of genetic generation

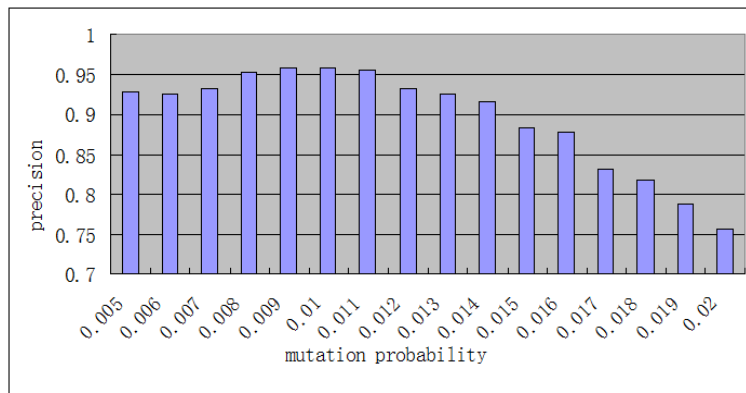


Figure 4. Comparison of classification precision

5. Conclusions

Aiming at the optimization problem of SVM multi-classification, this paper proposes an improved GA to generally optimize the parameters of all SVMs in the SVM multi-classification decision tree. First, the algorithm constructs multiple chromosomes for multiple SVM parameters. Each chromosome represents the SVM parameter of each node in the decision tree, and then re-designs the encoding strategy and crossover and mutation operators in genetic operation.

In addition to the parameter gene segment, the chromosome numbered gene segment is added during chromosome coding. At the crossover stage of chromosomes, the gene segment on each chromosome does not participate in crossover operation, but it will mutate at a lower probability. This mutation operation allows the GA to not only reduce the number of iterations when optimizing the parameters of multiple SVM, but also make the algorithm converge faster, and it can also jump out of the locally optimal solution and find the globally optimal solution.

The experimental results show that the algorithm improves the classification performance of the SVM multi-classification algorithm and improves the classification accuracy.

Acknowledgments

This work was supported by the National Natural Science Foundation of China ([2018]61741124) and Central Special Funds to guide the development of local science and technology projects ([2016] 4008). We also thank the anonymous reviewers sincerely for their significant and valuable feedback.

References

1. S. Besbes, Z. Lachir, "Multi-class SVM for Stressed Speech Recognition," *International Conference on Advanced Technologies for Signal and Image Processing*, IEEE, pp. 782-787, 2016
2. G. S. Cho, N. Gantulga, Y. W. Choi, "A Comparative Study on Multi-class SVM & Kernel Function for Land Cover Classification in a KOMPSAT-2 Image," *Ksce Journal of Civil Engineering*, pp. 1-11, 2016
3. H. Guo, W. Wang, "An Active Learning-based SVM Multi-class Classification model," *Pattern Recognition*, vol. 48, no. 5, pp. 1577-1597, 2015
4. S. Kumar, S. Mishra, P. Khanna, "Precision Sugarcane Monitoring Using SVM Classifier," *Procedia Computer Science*, vol. 122, pp. 881-887, 2017
5. K. C. Lin, T. Y. Liu, P. H. Chen, "Use Support Vector Machine (SVM) to Estimate Gas Concentration in Mixture Condition," *International Conference on Applied System Innovation*, IEEE, pp. 944-946, 2017
6. P. Li, H. Meng, X. Wang, "A Feature Selection Method Based on the Sparse Multi-class SVM for Fingerprinting Localization," *Vehicular Technology Conference*, IEEE, pp. 1-5, 2014
7. Q. Li, L. Chen, "Study on Multi-class Text Classification Based on Improved SVM," *Practical Applications of Intelligent Systems*, Springer Berlin Heidelberg, pp. 519-516, 2014
8. A. S. Murugavel, S. Ramakrishnan, "Hierarchical Multi-class SVM with ELM Kernel for Epileptic EEG Signal Classification," *Medical & Biological Engineering & Computing*, vol. 54, no. 1, pp. 149-161, 2016
9. J. Novakovic, A. Veljovic, S. S. Ilic, "Improving the Accuracy of SVM Algorithm in Classification Problems with PCA Method," *Intelligent Information Technologies for Industry*, pp. 66-73, 2018
10. A. Reda, E. Fakharany, M. Hazman, "Early Prediction of Wheat Diseases Using SVM Multiclass," *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics*, pp. 257-269, 2018
11. B. Sidaoui, K. Sadouni, "Binary Tree Multi-class SVM Based on OVA Approach and Variable Neighbourhood Search Algorithm," *International Journal of Computer Applications in Technology*, vol. 55, no. 3, pp. 183-189, 2017
12. E. Saeedi, M. S. Hossain, Y. Kong, "Multi-class SVMs Analysis of Side-channel Information of Elliptic Curve Cryptosystem," *International Symposium on PERFORMANCE Evaluation of Computer and Telecommunication Systems*, IEEE, pp. 1-6, 2015
13. M. Shaik, "Improved Normalization Approach for Iris Image Classification Using SVM," *Advances in Electronics, Communication and Computing*, pp. 139-145, 2018
14. W. Wenjing, S. Xiaohua, "Classification of Images of Strip Surface Defects Based on the Improved SVM Algorithm," *Revista de la Facultad de Ingenieria*, vol. 32, no. 1, pp. 244-254, 2017
15. Y. Xue, P. Beausero, "Transfer Learning for One Class SVM Adaptation to Limited Data Distribution Change," *Pattern Recognition Letters*, pp. 100-108, 2017
16. C. Yin, L. Shi, J. Wang, "Short Text Classification Technology Based on KNN+Hierarchy SVM," *Advanced Multimedia and Ubiquitous Engineering*, pp. 633-639, 2017
17. J. Zhang, C. Zhao, F. Xu, "SVM-Based Sentiment Analysis Algorithm of Chinese Microblog Under Complex Sentence Pattern," *International Conference in Communications, Signal Processing, and Systems*, pp. 801-809, 2016
18. Z. Zhao, X. Zhang, M. Han M, "Research on Multi-Faults Classification of Hoister Based on Improved LMD and Multi-class SVM," *International Symposium on Computational Intelligence & Design*, IEEE, pp. 573-576, 2015

Ren Qian is a Master's student in the College of Computer Science and Technology, Guizhou University. His current research interests include Data Mining and Recommender System.

Yun Wu received his Ph.D. degree from Guizhou University, Guizhou, China, in 2009. He is currently an associate professor, graduate supervisor, and a member of the China Computer Society. His research interests include Distributed Computing, Game Theory, Recommender System, and Big Data and its Application.

Xun Duan received his Ph.D. degree from Guizhou University, Guizhou, China, in 2007. He is currently an associate professor and graduate supervisor. His research interests include Computer Network, Data Mining, and Big Data and its Application.

Guangqian Kong received his Ph.D. degree from Guizhou University, Guizhou, China, in 2009. He is currently an associate professor, graduate supervisor, and a member of the China Computer Society. His research interests include Computer Network, and Big Data and its Application.

Huiyun Long received his Ph.D. degree from Guizhou University, Guizhou, China, in 2009. She is currently an associate professor and graduate supervisor. Her research interests include Computer Software, and Big Data and its Application.