Human resource recommendation algorithm based on ensemble learning and Spark

Zihan Cong[1], Xingming Zhang[1], Haoxiang Wang[1] and Hongjie Xu[1]

[1] School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510006, China.
Email: hxwang@scut.edu.cn

Abstract. Aiming at the problem of "information overload" in the human resources industry, this paper proposes a human resource recommendation algorithm based on Ensemble Learning. The algorithm considers the characteristics and behaviours of both job seeker and job features in the real business circumstance. Firstly, the algorithm uses two ensemble learning methods-Bagging and Boosting. The outputs from both learning methods are then merged to form user interest model. Based on user interest model, job recommendation can be extracted for users. The algorithm is implemented as a parallelized recommendation system on Spark. A set of experiments have been done and analysed. The proposed algorithm achieves significant improvement in accuracy, recall rate and coverage, compared with recommendation algorithms such as UserCF and ItemCF.

1. Introduction

With the rapid growth of data, human resources information management becomes complex and confusing. Job seekers are also easy to fall into the "information trail". The problem is mainly reflected as: for job seekers, the real interest in the job information only a small part of the information on the job seekers. Searching and screening process usually take a lot of time and effort. It is very challenging to accurately and quickly find their own interest in the recruitment of information.

In order to solve this problem, people has proposed a catalog and search engine two representative solutions [1]. These programs in the human resources industry has a more mature application. The current online recruitment service model can be divided into categories: Category recruitment model and Vertical recruitment model. Category recruitment model is mainly used in splitting catalogs. The recruitment of information is based on different topics for different categories. But the disadvantage is that it can only cover the limited popular information. Vertical recruitment mode does not produce recruitment data, but mainly by building a strong search engine, providing users with such as keyword search services, capturing the recruitment site to enrich the information to provide services for users. But the disadvantage is that in the real world, job seekers are often uncertain about their needs or difficult to describe their needs, hence the effect of search engines will greatly reduce [2].

The recommendation system is a kind of technology that can dig out the potential demand of the user. The recommendation system is considered to be the most effective way to solve the problem of "information overload"[3], compared with the split catalog and search engine. The system in the human resources research and application effect is unsatisfactory. Most human resource recommendation systems take into account only job seekers' application for jobs, and are not fully utilized for job seekers and job attributes (such as payroll, geography, industry) [4]; thus, based on Ensemble learning human resources recommendation algorithm, to help in-depth mining job seekers and business needs between the existing relationship between the realization of the precise recommendation of human resources.

2. Human Resource Recommendation Algorithm Based on Ensemble Learning

As the content, form and quantity of the original data are different. The recommendation algorithm is generally based on the specific business scenario. For the human resources circumstance, some common recommendation algorithm recommended effect is not obvious. In order to solve this problem, this paper proposes a human resource recommendation algorithm based on ensemble learning. It can be used to personalize the job seekers by combining the characteristics of human resources, including the following four aspects: data pre-processing, data sampling, ensemble learning, recommendation generation.

2.1. Data Pre-Processing

The application background of this algorithm is based on social security business data. In the system business scenarios, the data sources include user data, job data and behaviour data. These data are collectively referred to as raw data. There are problems of inconsistencies, incompleteness and incorrectness in the data format on the actual analysis, so the raw data must be processed which is known as the step of Extract-Transform-Load(ETL). Otherwise, it is futile and meaningless in the analysis and calculation of ambiguous, inaccurate numerical data [7]. So the need for pre-processing of the original data, the entire data pre-processing, includes the following sections. The flow chart is shown in figure 1:

1) Data extraction: The raw data is extracted from the original data in a streaming fashion.
2) Data cleansing: Clean dirty data from raw data.
3) Data transform: through the development of conversion rules, the original text information is transformed.
4) Data loading: the processed data is loaded into a well-designed data warehouse.

Figure 1. Data Pre-processing Flow Chart

2.2. Data Sampling

First, for the data in the data warehouse for statistical analysis, the recommended problem is a classification problem to produce recommended / not recommended two classifications. So 'not be interested behaviour' should be marked as a negative sample, and the behaviours of application, collection and browsing are marked as a positive sample. The total amount of behaviour data is 170,884, in which behaviour of 'not interested' is 23,516, and the behaviours of apply, collect, browse is 147,328. Hence, the sparse degree of the entire data set is:

$$\mathrm{p} = \frac{17084}{15000 \times 5000} \cong 0.00227 \tag{1}$$

Due to the sparseness of the data, if these non-user behaviour data is not dealt with, they will have a great impact on the recommended quality. Taking into account the reality of human resources circumstance, if the user does not produce behaviour on the post, it may mean the user is not interested in the post. However, if all the data without user behaviour is marked as negative, the positive and negative sample ratio r for the entire data set is:

$$r = \frac{147328}{15000 \times 5000 - 147328} \cong 0.00196 \tag{2}$$

In order to solve the problem of data sparseness and imbalance between positive and negative samples in the sampling process, the samples will be solved by sampling means. On one hand, sampling needs to ensure that the number of negative samples and the number of positive samples are equal; on the other hand, the implicit data and emphasis should be on those who are very popular. Based on this, the following sampling strategy is used to sample:

1) The positive samples were sampled using the SMOTE [8] algorithm, and the original 147328 positive feedback behaviour data was over-sampled for five times the original, i.e., 736640 positive samples.
2) The user is marked as "not interested" behaviour data using SMOTE algorithm, the original 23516 negative feedback behaviour data oversampling to 235160.
3) For each user, we take the highest view of the top 100 non-user behaviour of the job data as negative samples. Hence we sampled a total of 5000 * 100 = 500000 data.

2.3. Ensemble Learning

In the training phase, there are two ways to calculate: Bagging and Boosting [5]: The user interest model for each algorithm is output respectively, and then these two interest models are merged to get the fusion user interest model. The algorithm uses the decision tree as a basic classifier for integration [6]. The flow chart of generating user interest model is shown in figure 2.

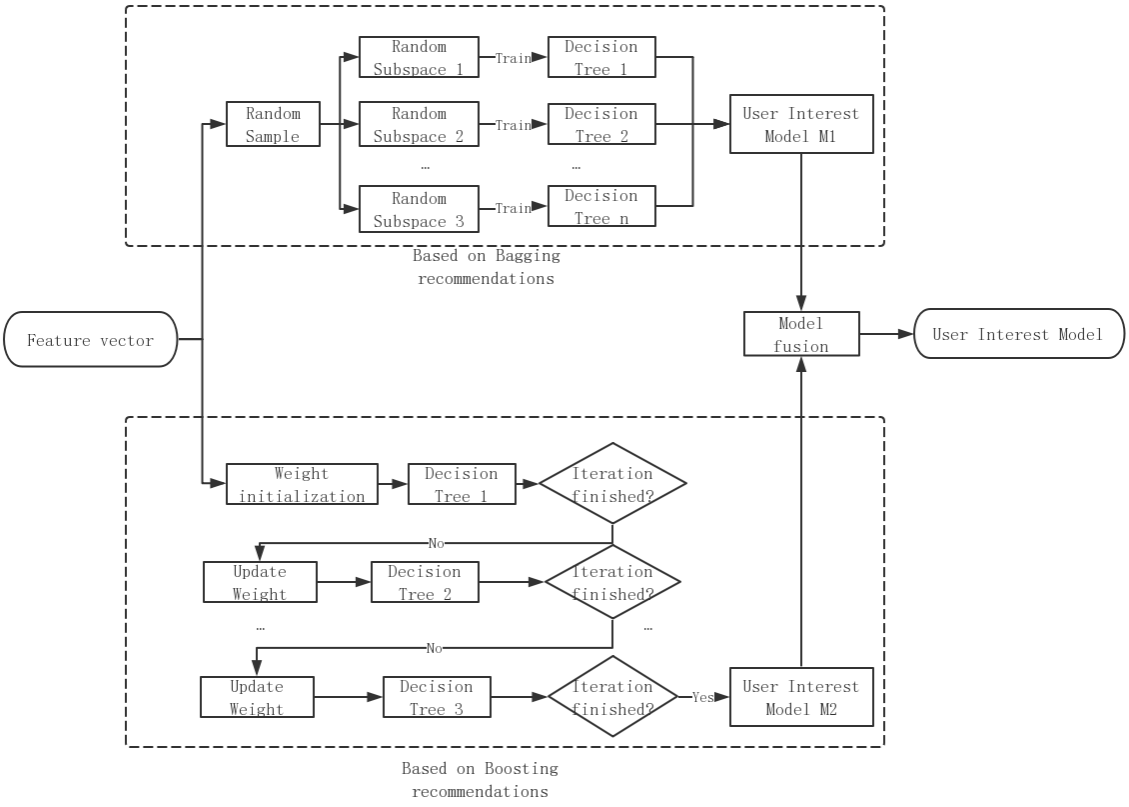

Figure 2. Generating a flow chart for the user interest model

Among them, the use of Bagging integration, random sampling to get n random sub-space and training for the n decision tree model, these decision tree model are combined, which let n models to each user U to predict job I of the classification, select the most frequent class p, and at the same time calculate the confidence. The Confidence formula is as follows:

$$Confidence(U,I,p) = \frac{\sum_{i=1}^{K} label(p)}{K} \quad (3)$$

Then the user's classification of the job results and classification confidence are added to M1, until the traversal of the entire user and post, the output interest model M1, it is a <user id, post id, predictive results, confidence> tuple.

Based on Boosting's user model interest generation, we first need to define a cost function, and the whole algorithm goal is to get the cost function to be optimized. In this process, the model is established

for each tree, and the cost function of each tree is calculated. Then, the cost function is used to calculate the optimal solution in the descending direction of the gradient. After constant iterations, the output model predicts the effect gradually ideal.

The model of the user interest models M1 and M2 can be obtained. For each target user i and target post j, the following formula is used to define the user interest level of i in j: $l_{m1,i,j}$ represents the user's forecast category for the post. If the forecast is 1, the prediction is negative; $q_{m1,i,j}$ represents the user's confidence in the job prediction, the confidence interval range is [0, 1].

$$C_{m1,i,j} = l_{m1,i,j} * q_{m1,i,j} \tag{4}$$

The following formula defines the user interest degree of i in j, where $l_{m2,i,j}$ represents the user's prediction category for the job:

$$C_{m2,i,j} = l_{m2,i,j} \tag{5}$$

M2 has a degree of interest with value 1 or -1, and if the result is a direct result fusion, the contribution generated by M2will be greater than M1. In order to avoid this problem, we introduce the fusion factor in the fusion process. The following formula defines the degree of interest that occurs after fusion:

$$C_{i,j} = C_{m1,i,j} + r * C_{m2,i,j} \tag{6}$$

The calculated value of $C_{i,j}$ is positive if user i is interested in post j, and the higher the value, the higher the degree of interest. Conversely, the negative representative is not interested, the greater the absolute value stands for less interested. Calculate the corresponding position of each user and add to M, get the final integration of the user interest model, the model is a <user id, job id, interest degree> of the triple set, as the basis for the recommendation.

2.4. Recommended to Produce

The user interest model obtained in the previous section represents the user's classification results and the degree of interest for each post. In order to generate the final recommendation, a recommendation based on Top-N is used to generate the recommended results. First, the user interest model is grouped according to user id, the interest set for each user is selected to sort out the position of interest of the user, and the sorting operation is according to the degree of interest is obtained. The result set is from high to low, K posts. For online experiments, recommendations will be made for all users. For offline experiments, the behavioural records of the test set will be compared with the predicted behaviour and the corresponding recommended indicators will be calculated for the analysis and improvement of the algorithm. The flow chart is as follows:

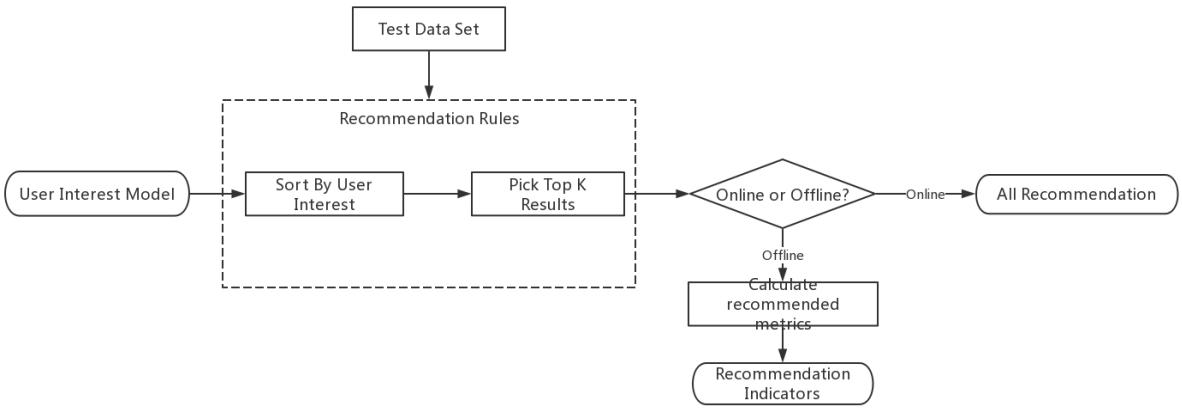

Figure 3. Recommends generating a flow chart

2.5. Algorithm Parallelization

In order to improve the efficiency of the implementation of the algorithm, here will be based on the Spark computational framework for the original algorithm parallelization. Spark first needs to interact with the data warehouse on the cloud platform to read and write raw data. Spark SQL is the interface through which Spark reads and writes to the data warehouse. It takes the records of the raw data from the data warehouse and converts it to RDD, providing a data foundation for the recommended calculation [9]. At the same time, the program main function builds Driver role through the Spark Context. Driver is responsible for generating computing tasks and directed acyclic graph of the scheduling. Tasks will be assigned to the Executor to perform. After executor completes parallel calculation of the training task, the data will be in the form of RDD as the output of the calculation results. Spark on the overall recommendation algorithm calculation process is shown as follow:

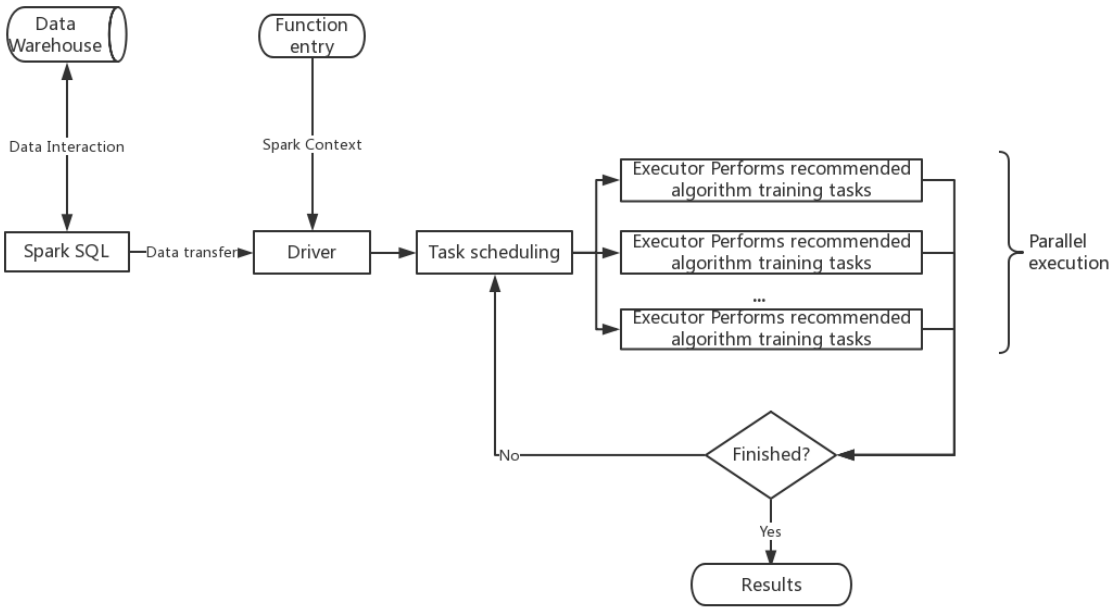

Figure 4. Algorithm parallelization flow chart

3. Experimental Results and Discussion

In order to verify the overall effect of the recommended system, the experiments are done in the three aspects of accuracy rate, recall rate and coverage [10]. For the algorithm, the following table describes the key parameter descriptions:

Table 1. Meaning of the meaning of the list

Parameter	Meaning	Source
D1	Maximum depth of Decision tree	Bagging
N	Number of decision trees integrated	Bagging
D2	Maximum depth of Decision tree	Boosting
T	Maximum number of iterations	Boosting
r	Fusion factor	Interest model fusion

In this paper, we use the parameter group of D1=12, N=25, D0=10, T=10, r=0.5 to generate the interest model. Meanwhile, in order to carry out the comparison experiment, this paper uses two kinds of most common algorithms based on user co-filtering (UserCF) and content-based collaborative filtering (ItemCF) on the same data set. The experimental results are as follows:

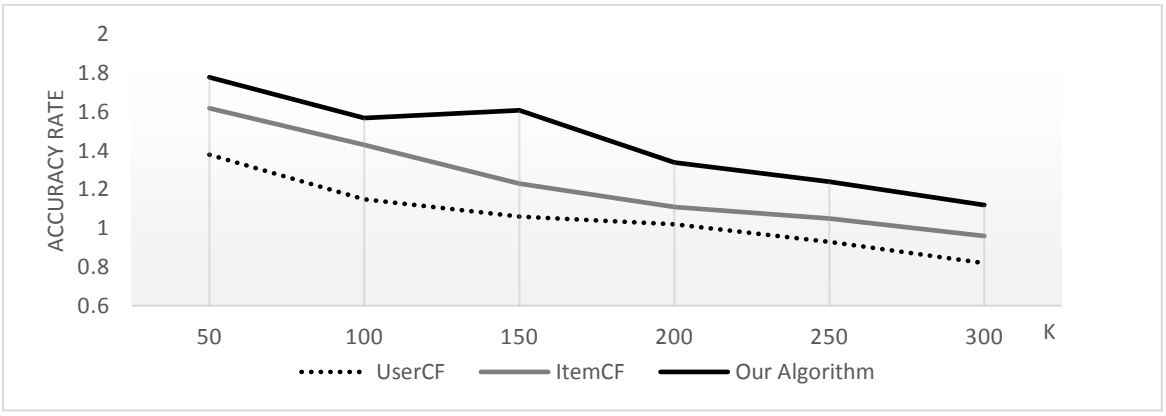

Figure 5. Accuracy curves for different recommended algorithms

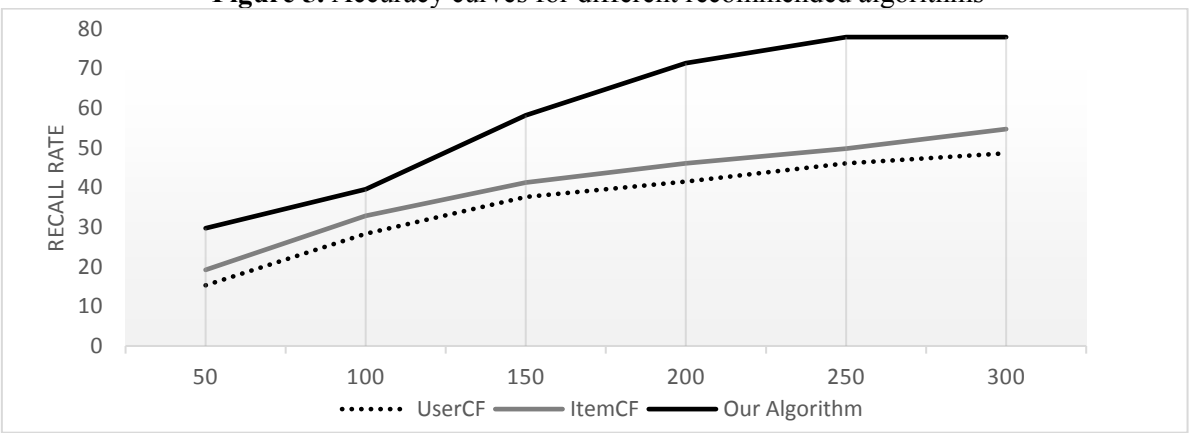

Figure 6. Recall curves **for** different recommended algorithms

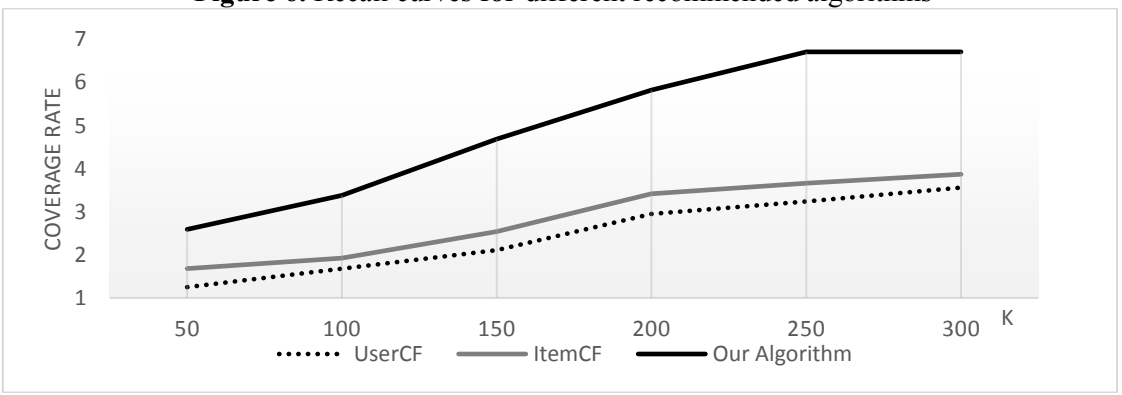

Figure 7. Coverage curves for different recommended algorithms

From the above graphs, it can be seen that the proposed algorithm has been more accurate than the traditional cooperative filtering algorithm with the increasing of recommended length. In the aspect of recall rate, since the recall rate calculation formula is based on the entire test data, so the recall rate curve always keeps increasing. At the particular points for K=250 and k=300, the recall rate of the algorithm is fixed, which shows that the proposed algorithm for the users predicted as "interested" entries has reached the bottleneck value. The recall rate of our algorithm can reach 77.9, which is higher than that of UserCF (48.6) and ItemCF (54.7). It shows that our algorithm has obvious advantages in recall rate. Similarly, for coverage, with the increase of K, the coverage of this algorithm has been significantly higher than the traditional co-filter algorithm coverage. Our algorithm's highest coverage rate can reach 6.71, which is about twice of the coverage of UserCF. Through the analysis of above experiments results,

it can be approved that the proposed algorithm conducts a significant improvement compared with the traditional collaborative filtering algorithms.

The significant improvement is due to the reason that the traditional collaborative filtering recommendation algorithms only take into account the user's historical behaviour data, regardless of the user and project attribute information. In some recommendation scenarios, which have a large number of user interactions, traditional algorithms may have a good performance. Such as movie recommendation, user and film attributes are difficult to be facilitated, and the available data is mainly about user behaviours. For human resources recommendation, besides of user behaviour information, user and job themselves are abundantly informational, which reflects the characteristics of the user and the characteristics of the post. This article aims to use this information through the ensemble learning. In addition, in order to improve the recommended coverage, our algorithm is based on the post of the popularity of the post, focusing on the jobs which are popular but not interested by some users. This approach can improve the extraction of users' interests, solving the "long tail distribution" problem for recommendation.

4. Conclusion

This paper points out some shortcomings in solving the "information overload" problem for the current human resource recommendation. This paper proposes a recommendation algorithm based on ensemble learning, to generate user interest model. We use this algorithm to generate job recommendation for users. The implementation of the algorithm is parallelized on Spark platform. Finally, the performance of this algorithm is evaluated by a set of experiments. Compared with the traditional recommendation algorithms, such as UserCF and ItemCF, our algorithm shows significant advantages in the circumstance of human resource recommendation.

5. Acknowledgement

This work is financially supported by Guangdong Provincial Science and Technology Plan (No. 2016B030308002).

6. References

[1] Loeb S, Terry D. Information filtering [J]. Communications of the Acm, 1992, 35(12):26-28.
[2] Arayasirikul S, Chen Y H, Jin H, et al. A Web 2.0 and Epidemiology Mash-Up: Using Respondent-Driven Sampling in Combination with Social Network Site Recruitment to Reach Young Transwomen [J]. AIDS and Behavior, 2016, 20(6):1265-1274.
[3] Ricci F, Roach L, Shapira B, et al. Recommender Systems Handbook[M] Recommender systems handbook. Springer, 2011:1-35.
[4] White T J, Redner R, Skelly J M, et al. Examination of a Recommended Algorithm for Eliminating Nonsystematic Delay Discounting Response Sets[J]. Drug & Alcohol Dependence, 2015, 154:300.
[5] Freund Y. An improved boosting algorithm and its implications on learning complexity[C] ACM Conference on Computational Learning Theory, COLT 1992, Pittsburgh, Pa, Usa, July. DBLP, 1992:391-398.
[6] Podgorelec V, Zorman M. Decision Tree Learning [M] Machine Learning Models and Algorithms for Big Data Classification. 2016:1751 - 1754.
[7] Berro A, Megdiche I, Teste O. A Content-Driven ETL Processes for Open Data [M] New Trends in Database and Information Systems II. Springer International Publishing, 2015:29-40.
[8] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique [J]. Journal of Artificial Intelligence Research, 2011, 16(1):321-357.
[9] Shen Z, Johnsson M, Zhao Z, et al. Spark Plasma Sintering of Alumina [J]. Journal of the American Ceramic Society, 2010, 85(8):1921-1927.
[10] Okita S C N C. EVALUATION SYSTEM AND EVALUATION METHOD: US, WO/2006/129711[P]. 2006.