Hybrid employment recommendation algorithm based on Spark

Zuoquan Li[1], Yubei Lin[2]*and Xingming Zhang[1]

[1] School of Computer Science and Engineering, South China University of Technology, Higher Education Mega Centre, Panyu District, Guangzhou, China
[2] School of Software Engineering, South China University of Technology, Higher Education Mega Centre, Panyu District, Guangzhou, China
Email: yupilin@scut.edu.cn

Abstract. Aiming at the real-time application of collaborative filtering employment recommendation algorithm (CF), a clustering collaborative filtering recommendation algorithm (CCF) is developed, which applies hierarchical clustering to CF and narrows the query range of neighbour items. In addition, to solve the cold-start problem of content-based recommendation algorithm (CB), a content-based algorithm with users' information (CBUI) is introduced for job recommendation. Furthermore, a hybrid recommendation algorithm (HRA) which combines CCF and CBUI algorithms is proposed, and implemented on Spark platform. The experimental results show that HRA can overcome the problems of cold start and data sparsity, and achieve good recommendation accuracy and scalability for employment recommendation.

1. Introduction

With the improvement of communication infrastructure and the development of Internet-related technologies, the amount of network information is exploding, which greatly satisfies the growing information needs of people. However, it makes the user cannot find the desire information in time. Therefore, people put forward recommendation algorithms, which can quickly find the desire information by analysing people's historical behaviour data, and provide an optimal recommendation result. At the same time, the continuous development of the network has also greatly broadened the channels of employment. Both job seekers and companies have begun to use the network for employment. However, there is increasing difficulty to deal with massive job-related information. Therefore, a good employment recommendation algorithm is so important that it is worth doing further research.

Recommendation algorithm, which is the core of recommendation system, has become a hot topic since the research of recommendation system was begun at the end of the last century. In 1994, user-based collaborative filtering algorithm was applied for news recommendation [1]. In 2003, item-based collaborative filtering algorithm (IBCF) was put forward [2]. In the same year, Slope one collaborative filtering algorithm was proposed. It is simple and scalable. However, it cannot solve the problems of cold start and data sparsity [3]. To overcome the cold-start problem, content-based recommendation algorithm (CB) was proposed [4]. In recent years, machine learning algorithms, such as clustering, association rules and matrix decomposition, have been implemented for recommendation [5]. In [6] a collaborative filtering algorithm based on user interest change was presented. A graph recommendation algorithm based on attention relation and multi-user behaviours was introduced in [7]. A factor decomposition machine algorithm based on high order deviation was developed in [8].

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

Some scholars introduced recommendation algorithms into the field of human resources. The implicit semantic model was used to conduct two-way recommendation of talent and position in [9]. The technology of machine learning was implemented to predict the future career change of applicants in [10]. In [11] the ontology-based network recruitment recommendation was proposed. In [12] PageRank algorithm based on random walk model for job recommendation was introduced. Models of user interest and job preference were built to improve the collaborative filtering recommendation algorithm in [13].

For employment recommendation, not only requirements of companies, but also the job seekers' conditions and preferences should be considered. Moreover, with the continuous expansion of network information, a good recommendation system must be powerful enough to deal with the massive data. However, the algorithms mentioned above were almost designed as stand-alone calculation models. Therefore, in this paper a hybrid recommendation algorithm (HRA) base on Spark, which is a fast and versatile computing engine designed for large-scale data processing, is proposed. First a clustering collaborative filtering algorithm (CCF) is developed. The predictive scores of users can be obtained by CCF. The highest five predictive scores of each user are filled into the original matrix of user-item rating. Then this new rating matrix serves as the input data of a content-based algorithm with users' information (CBUI), which takes job seekers' information into account to improve the users' preference vectors. Finally, better recommendation results can be achieved.

2. HRA Based on Spark

2.1. Data Preprocessing

The experimental data of this paper are collected from a research project, including the applicants' personal information, jobs' basic information and the historical behaviour of applicants. The quality of the experimental data will greatly affect the final recommendation result. Since the collected data cannot be used directly in the recommendation algorithm, it is necessary to conduct data pre-processing as follows.

1) Some original data have duplicate values. For example, some companies repeatedly publish the same job. Only the latest information of this job will be retained after data pre-processing.

2) Some original data have missing or illegal values. In data pre-processing, default values will be set for these data. For example, the average wage of all jobs will be set as the default value of salary.

3) The salaries in the original data can be paid monthly, weekly or daily, which makes the wage-related attribute very confusing. We need to convert these data to a union form.

4) Some of the attributes in the original data, such as applicant's telephone number, job title, contact telephone number, and so on, are not necessary for our recommendation algorithm. Therefore, we should delete these attributes in data preprocessing.

2.2. CCF Based on Spark

The traditional IBCF has to find the neighbours of the target item in all items, which results in a lot of time spent on searching. We can reduce the time for finding neighbours by introducing clustering to the algorithm. Specifically, the clustering algorithm is used to divide the item set into clusters. Then we can find the nearest neighbours of the target item in each cluster. Finally, IBCF is used in the neighbour set of the target item. At the same time, because a cluster set is much smaller than the entire set of all items, we can find most of the neighbours of target item in a smaller range rather than the entire item space, and effectively improve the speed of the recommendation algorithm. The flow chart of CCF is shown in figure 1.

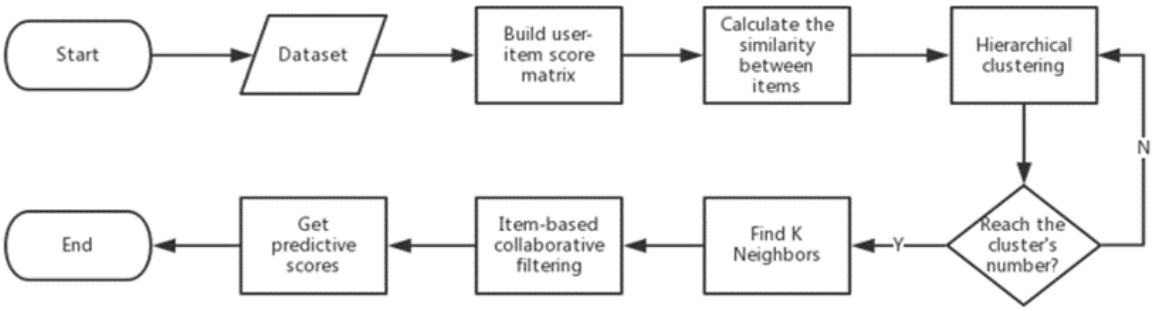


Figure 1. Flow chart of CCF.

As can be seen in figure 1, the algorithm mainly includes the following steps:

1) Construct the matrix of user-item score from the processed data set. In this paper, data cleaning, data conversion, attribute reduction, data aggregation and other methods are conducted in data pre-processing. After data pre-processing, the matrix of user-item score can be constructed by using the applicants' historical score data.
2) Calculate the similarity scores between items. Assume that vectors of item $\boldsymbol{i}$ and $\boldsymbol{j}$ are expressed by $\{r_{i1}, r_{i2}, \dots, r_{in}\}$ and $\{r_{j1}, r_{j2}, \dots, r_{jn}\}$ respectively, where r_{ix} and r_{jx} are the scores which user x awards to the items $\boldsymbol{i}$ and $\boldsymbol{j}$ respectively. The similarity score between items $\boldsymbol{i}$ and $\boldsymbol{j}$ defined as $\mathrm{sim}(\boldsymbol{i},\boldsymbol{j})$ can be calculated by the following equation:

$$\mathrm{sim}(\boldsymbol{i},\boldsymbol{j}) = \frac{\sum_{x=1}^{n}(r_{ix}\times r_{jx})}{\sqrt{\sum_{x=1}^{n} r_{ix}^2}\sqrt{\sum_{x=1}^{n} r_{jx}^2}} \tag{1}$$

3) Conduct hierarchical clustering on items. After calculating the similarity scores between items, the items can be clustered according to the similarity scores. Once the clustering is complete, we can define a cluster centre for each cluster. The new item needs to be compared with each cluster centre. Then the neighbour in the closest cluster can be found to improve the search speed.
4) Get the predictive score. After the K-nearest neighbours of the target item in the clustering result are obtained in the previous step, IBCF can be applied in the neighbour set to obtain the user's predictive score.

To improve the scalability and computing performance of CCF, the algorithm is implemented on Spark platform. In this platform, parallelism is achieved primarily through Resilient Distributed Datasets (RDD). The overall RDD conversion diagram of CCF is shown in figure 2.

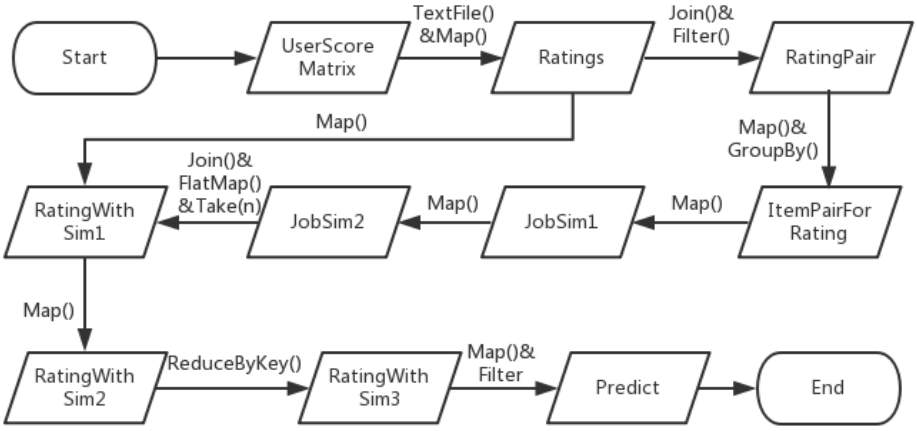


Figure 2. RDD conversion diagram of CCF.

It is noted that in figure 2, *UserScoreMatrix* is the input data of this algorithm, *Ratings* is the score matrix built from *UserScoreMatrix*, *RatingPair* is the collection of items rated by the same user, *ItemPairForRating* is a pair of items rated by the same user, *JobSim1 and JobSim2* represent the degree of similarity between each pair of items, *RatingWithSim1*, *RatingWithSim2* and *RatingWithSim3* are the records of *n* neighbouring items with the highest similarity to the target item, and the final predictive score is saved in the RDD named *Predict*.

2.3. CBUI Based on Spark

In Section 2.2, we study the clustering collaborative filtering algorithm, which is based on the user rating to cluster the items and improve the real-time performance of the employment recommendation algorithm. However, it only considers the users scores. Neither users' information nor items' information is taken in account. Moreover, collaborative filtering recommendation algorithm has the problem of cold start. To address these problems, an approach based on the traditional content-based recommendation algorithm is proposed, in which users' information, such as age, gender, educational level, and so on, is considered to find the users' preferences. The workflow of CBUI is as follows:

1) Construct the attribute vectors of items. The attributes of the items, including gender requirement, age requirement, academic requirements, job salary and job benefits, can be used to build the attribute vectors.
2) Mix the user's basic information and rating data, and construct the preference vectors of users. On one hand, we can obtain the user's favourite items from the user's rating information, and use the items' attributes obtained in the previous step to select these items' worktime, job benefits, job salary and other attributes. On the other hand, we can select some attributes from the user's basic attributes, such as gender, age, degree of education, working life, as part of the attributes of the user's preference vector. In this way, once the number of the user's rating is limit, the user's preference vector can be constructed according to the user's basic information.
3) Get Top-N recommendation results. After obtaining the attribute vectors of the item $\boldsymbol{i}$ and the user's preference vector $\boldsymbol{u}$, the similarity score between these two vectors defined as $\mathrm{sim}(\boldsymbol{u}, \boldsymbol{i})$ can be calculated by the following equation:

$$\mathrm{sim}(\boldsymbol{u}, \boldsymbol{i}) = \frac{|U \cap I|}{|U \cup I|} \tag{2}$$

where U and I are the attribute sets of $\boldsymbol{u}$ and $\boldsymbol{i}$ respectively. After obtaining the similarity scores, we can sort them and recommend a group of items with the highest similarity score.

Likewise, to improve the scalability and computational performance of CBUI, this algorithm is implemented on Spark. The parallel processing for CBUI mainly consist of constructing the item's attribute matrix, constructing the user's preference vector, calculating the similarity scores and obtaining the Top-N recommendation result. The RDD conversion diagram of CBUI is shown in figure 3.

It is noted that in figure 3, the RDD named *DataSet* is the input data of the algorithm, and we built the items' attribute matrix which is named *JobAttrs*, the users' rating matrix which is named "Ratings" and the users' attribute matrix which is named *UserAttrs* from *DataSet*. The other created RDDs are introduced as follows:

- *Ratings2*: the top *n* items rated by each user.
- *ItemWithAttrs2*: the set of jobs' attributes grouped by user.
- *Profile PartA*: the preference attribute of top *n* items with highest rating by each user.
- User Profile: the preference attribute of user.
- *Profile With Job Attrs*: the cartesian set of *UserProfile* and *JobAttrs*.
- *FilterItem*: rated items.
- *FilterResult*: result without rated items.
- *Recommend Result*:top *n* recommendation results.

2.4. HRA

In this section, a HRA based on CCF and CBUI can be implemented as follows. First, users' predictive scores are obtained by CCF. Then the highest five scores of each user are filled into the original user-item rating matrix. Finally, the new rating matrix serves as the input data of CBUI, which can improve the user's preference vector and achieve better recommendation results. The overall process of HRA is shown in figure 4.

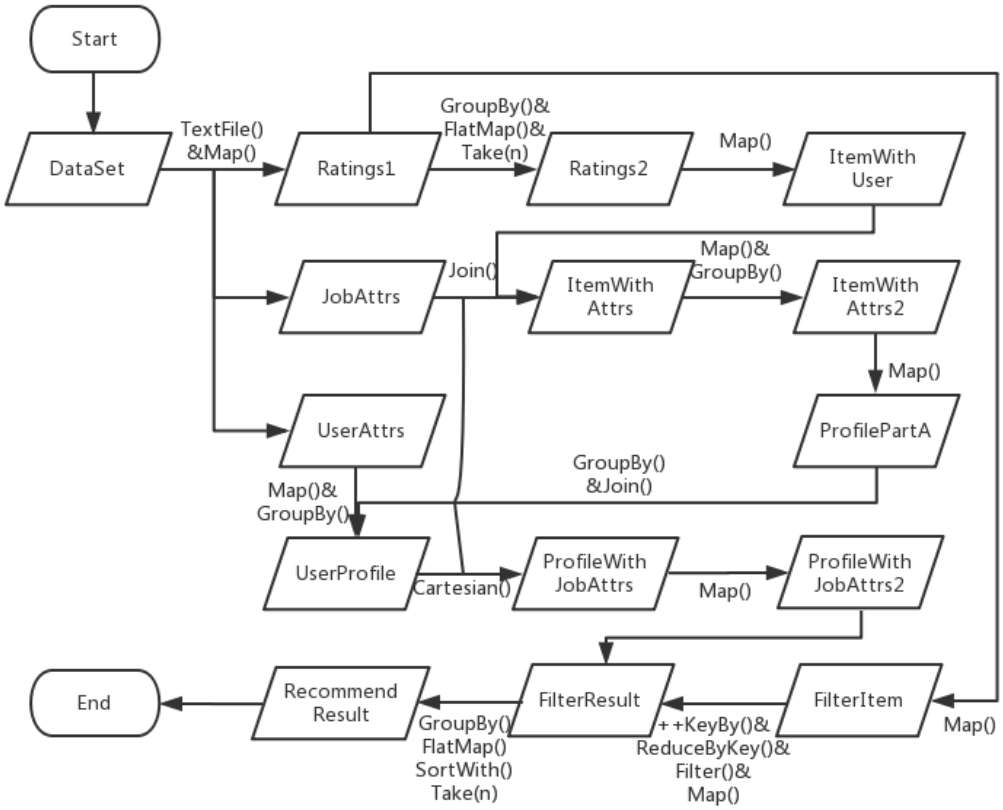


Figure 3. RDD conversion diagram of CBUI.

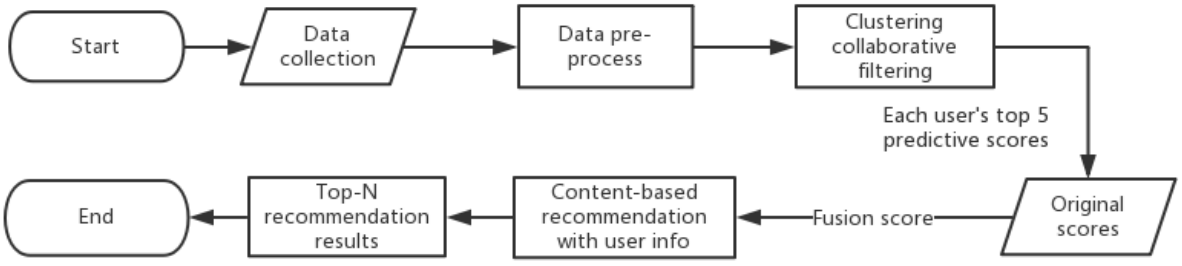


Figure 4. The overall process of the proposed HRA.

3. Experimental Setting and Results

3.1. Experimental Setting

Dataset & Metrics. In our experiment, 4692 users' information, 15000 jobs' information and 170844 user-job score data are involved. In addition, to verify the experimental results, accuracy rate P_u defined as the following equation (3) is used as the experimental indicator.

$$P_u = \frac{|R_u \cap I_u|}{R_u} \tag{3}$$

where R_u is the collection of the recommended items, and I_u is the set of items selected by the users in the test set.

Experimental Environment. In the experiment, we build a Spark cluster with four virtual machines. One of the virtual machines serves as the master node of the cluster, and the other three virtual machines serve as worker nodes. The configuration of each virtual machine is as follows: 8G memory, 4 cores 1.799GHZ CPU, 80G disk space, centos 6.7 operating system, 1.6.0 version of Spark, 2.6.0 version of Hadoop and 2.10.4 version of Scala.

3.2. Results and Analysis

Predictive Accuracy. CBUI is based on the traditional CB, so we firstly test these two algorithms on the same data set for the sake of comparison. The recall rate of these two algorithms are shown in figure 5.

As shown in figure 5, CBUI is superior to the traditional CB, which means that the users' basic information is used rationally so that when the ratings of users are less or even missing, users' preferences vectors can be built according to users' information. It can effectively alleviate the problem of cold start and improve the recommended results.

In addition, in the experiment, CCF, CBUI and HRA are tested on the same data set. When the number of clusters is 90 and the number of neighbour items is 40, the accuracy rate of these three algorithms are shown in figure 6.

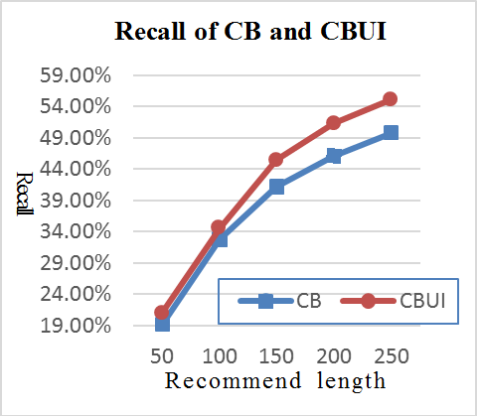


Figure 5. Recall of CB and CBUI.

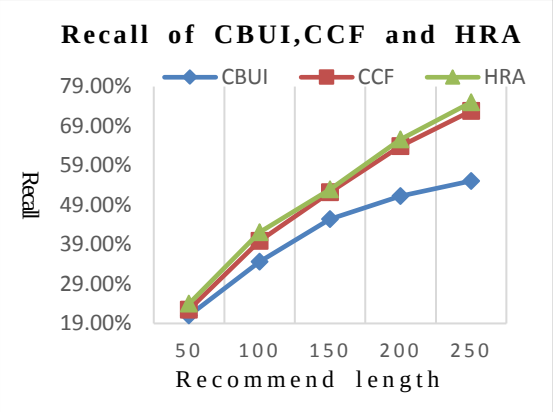


Figure 6. Recall of CBUI, CCF and HRA.

As shown in figure 6, with the increment of the recommendation list's length, the recall rates of these three algorithms are increased, while the rising speed is gradually slowed down. When the recommendation list's length is 250, the recall rates of these three algorithms are 75% (HRA), 72.8 % (CCF) and 55.1% (CBUI) respectively. Since the predictive score of CCF can alleviate the sparsity of the users' scores and improve the users' preferences vectors, the recall rate of HRA is much higher than that of CBUI, which shows that the proposed hybrid strategy is effective.

Parallelism. The parallelization of HRA is implemented on Spark. Speedup is defined as equation (4) to measure the scalability and performance of the parallel algorithms.

$$Speedup = \frac{T_1}{T_p} \tag{4}$$

where T_1 is the operation-time in the single-processor system and T_p is the operation-time in the p-node-processor system. In the ideal case, the parallel algorithm's Speedup should be linearly incremented as the number of nodes increases. In the Spark cluster, T_1 refers to the algorithm operation time in pseudo-distributed mode.

To validate the *Speedup* of the parallel HRA, the number of nodes in the cluster is gradually increased, and the actual operation time required for the task of the parallel HRA is recorded. The

Speedup of the parallel algorithm is shown in figure 7. As can be seen from it, with the increment of the number of data nodes, the *Speedup* curve of HRA basically conforms to the linear trend, which shows that the algorithm is scalable and can shorten the time by increasing the nodes. The *Speedup* curve of the algorithm is different from the ideal curve "Y = X", mainly because the data transmit between nodes takes time, and the problem of data tilt will also affect the completion time.

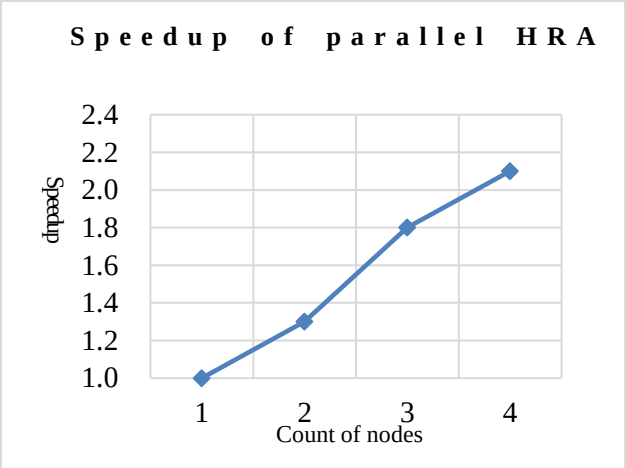


Figure 7. Speedup of parallel HRA.

4. Conclusion

The application of the employment recommendation algorithms, including CCF and CBUI, are studied in this paper. In addition, aiming at addressing the shortcomings of the single recommendation algorithm, a hybrid recommendation algorithm based on these two algorithms is proposed, which effectively solves the problem of cold start and alleviates the influence of data sparsity of the traditional CB. At the same time, the parallelization of the proposed algorithm is designed and implemented on Spark, which improves the parallelism and expansibility of the proposed algorithm. However, it is found that the parallelization efficiency of HRA has a certain gap with the ideal value, and it still has the space for improvement. Therefore, we will carry out an in-depth study on Spark to further improve the efficiency of the proposed algorithm.

5. Acknowledgement

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

6. References

[1] Goldberg D, Nichols D, Oki B M and Terry D 1992 Using collaborative filtering to weave an information tapestry Comm. ACM 35(12) 61-70
[2] Sarwar B, Karypis G, Konstan J and Riedl J 2001 Item-based collaborative filtering recommendation algorithms Int. Conf. on World Wide Web (Hong Kong) vol 4 (ACM) p 285
[3] Lemire D and Maclachlan A 2008 Slope one predictors for online rating-based collaborative filtering Comput. Sci. 21-3
[4] Longfei C and Xue Z 2013 The summary of personalized recommendation technology J. Hebei Normal Univ. Sci. Techno. 57-60
[5] Defu L, Cong Z, Xing X, Guangzhong S, Enhong C and Yong R 2014 GeoMF: joint geogra-phical modeling and matrix factorization for point-of-interest recommendation ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (New York City) (ACM) p 831
[6] Daoguo L, Dijiang H and Lianjie L 2017 Collaborative filtering recommendation algorithm based on user interest change Productivity Res. 1 19-21+55.
[7] Mengjuan L, Wei W, Yangxi L, Xucheng L and Zhiguang Qin 2017 AttentionRank ~ +: a graph recommended algorithm based on attention relation and multiuser behaviour J. Comput. 3 634-48

[8] Zihao Wang and Guiqiong Xu 2017 Recommended algorithm for factor decomposition machine based on high order deviation Application Res. Comput. 2 339-42
[9] Malinowski J, Keim T, Wendt O and Weitzel T 2006 Matching people and jobs: a bilateral recommendation approach Hawaii Int. Conf. on System Sciences (Hawaii) vol 6 (IEEE) p 137.3
[10] Paparrizos I, Cambazoglu B and Gionis A 2011 Machine learned job recommendation Proc. of the 5th ACM Conf. on Recommender Systems (Chicago) (ACM) p 325
[11] Qingqing S 2009 User modelling research of personalized network recruitment recommended system based on the ontology Nanjing Univ. Aeronautics and Astronautics 1 81-4
[12] Di W, Lijuan Z and Hongfei Lin 2011 Research and implementation of employment recommended system based on random walking 2011 J. Guangxi Normal Univ. (Natural Sci. Edition) 29(1) 179-85
[13] Hongjiao C 2014 The design and implementation of employment recommendation system for the graduate student based on context-aware technology Huazhong Normal Univ.