Employment Service System Based on Hybrid Recommendation Algorithm

Zhenqi Dong, Chunxia Leng, and Hong Zheng
Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai, China
cxleng@ecust.edu.cn

Abstract. The recommendation system can calculate the similarity between users and items in the system, which is based on different computational methods, analyzing of calculation results, and then calculates the items that may interest the users and recommend those items to users. Collaborative filtering algorithm is a popular algorithm in academia and industry, but it does have certain shortcomings such as cold start and sparse data. In this paper, a hybrid recommendation algorithm is proposed and applied to an employment service system by using job postings from websites obtained by web crawler as the dataset. The experimental result shows that the hybrid recommendation algorithm is able to the accuracy of employment information recommendation to some extent and meet the personalized needs of job-seeking users.

Keywords: Web crawler · Collaborative filtering recommendation · Content-based recommendation · Hybrid recommendation algorithm · Employment recommendation system

1 Introduction

In spring 2020, campus job fairs have been postponed due to the COVID-19 epidemic, making it imperative to promote college graduates’ job search by enhancing online employment services.

Traditional employment websites provide a huge amount of job information to users for retrieving, but they are not able to provide differentiated services by differentiating the user groups. Therefore, they cannot help users to find suitable jobs more easily and thus cannot increase employment rates.

Anderson points out in The Long Tail Theory that “we are about to leave retrieval and enter the age of recommendation” [1]. The recommendation system is a new type of information filtering system. After analyzing a large number of historical user behavioral records, they can filter out useful information from a huge amount of information by building a specific user’s interest model [4], and then actively push the recommendation results to the personalized service system of each user. If we only use collaborative filtering or content-based algorithm in the system, we cannot combine comprehensive information, such as personal data and browsing history, which will lead to the consequence that the recommendation results will be too simple [13], thus causing limit of the range of search results and reducing the chance of success. At the
same time, employment websites may recommend the same job to many users, which will lead to the fact that there will be too many uses compete for the same job, thus reducing the number of recommendation times of other jobs and harming the company’s interests.

Therefore, a hybrid recommendation algorithm is used to solve the “cold start” problem in the collaborative filtering algorithm. It effectively ensures the accuracy of the employment recommendation, which is of great value and significance for improving the level of satisfaction of both sides and the quality of employment.

2 Architecture of the Employment Recommendation System

The architecture of the employment system based on the hybrid recommendation algorithm is shown in Fig. 1 and consists of 3 main layers.

(1) Data layer: In the recommendation system, the personal information of job seekers, the results of web crawlers, and the data of user behavior are saved in the database and used for subsequent recommendation algorithms.

(2) Recommendation Business Layer: A hybrid recommendation algorithm is used to recommend jobs to users in different scenes by using collaborative filtering recommendation algorithm and content-based recommendation algorithm. The recommendation service layer generates a list of recommended jobs through the recommendation algorithm, which will be transmitted to the user interface layer for display.

(3) User Interface Layer: The user interface layer interacts with the user directly. The recommendation service will transmit the job list calculated by the

![Fig. 1. The architecture of the employment recommendation system](image-url)
recommendation algorithm to the user interface layer, and then this layer will query the detailed information of all the jobs in the list and recommend the information to the user for easy viewing.

3 Hybrid Recommendation Algorithm Model

The key to the research of the recommendation system lies in the recommendation algorithm. Currently, the most widely used recommendation algorithms are collaborative filtering recommendation algorithm and content-based recommendation algorithm. The recommendation algorithm based on collaborative filtering is a successful personalized recommendation technique. The algorithm extracts the user’s previous behavioral records, then builds user similarity model [4], and finally calculates the user’s interest prediction for the unbrowsed jobs, thus achieving the purpose of recommendation. The content-based recommendation builds a job similarity model by calculating the similarity between different jobs [6] such as text similarities and recommends the job which has the highest similarity with the job user is browsing. In this paper, based on the above two algorithms, a switching hybrid algorithm [10] is used to improve the “cold start” problem, and the “sparsity” problem of user similarity matrix [7] in the collaborative filtering algorithm is solved by pre-screening users with interest in intersection. In addition, it solves the problem of limited recommendation results when just uses one algorithm.

3.1 Advanced Collaborative Filtering Algorithm

3.1.1 Inadequacies of Traditional Collaborative Filtering Algorithm

The traditional collaborative filtering algorithm needs to calculate the similarity between any two users as the following expression [3].

\[
U_{m \times n} = \begin{pmatrix}
R_{1,1} & R_{1,2} & \cdots & R_{1,n} \\
R_{2,1} & R_{2,2} & \cdots & R_{2,n} \\
\vdots & \vdots & & \vdots \\
R_{m,1} & R_{m,2} & \cdots & R_{m,n}
\end{pmatrix}
\]

where \( R_{i,j} \) represents the similarity between user \( i \) and user \( j \).

However, in the real system, the similarity between a large number of users is 0. This means that the similarity matrix is a sparse matrix and a large number of values is 0. This situation will lead to the result that the computation cost is high, but most of the computed results are useless. Therefore, solving this problem can largely optimize the collaborative filtering recommendation algorithm.

3.1.2 Procedure of Advanced Collaborative Filtering Algorithm

The algorithm filters out the set of users who intersect with the current user and construct the user similarity matrix between these users, thus avoiding the appearance of the sparse matrix.
(1) Create table T with jobs as the key to record each job’s set of users who rate it.

(2) Create a user similarity matrix W to record the similarity between different users and set the default value as 0.

(3) Traverse through table T. For each post in table T, obtain the set of users who rate it. For every two of the different users \( j, k \) in the set, make the following updates in the user similarity matrix W: 

\[
W_{jk} = W_{jk} + 1, \quad W_{kj} = W_{kj} + 1,
\]

and apply a “heat penalty” through Eq. 2 to obtain the final user similarity matrix W.

\[
W_{uv} = \frac{\sum_{i \in N(u) \cap N(v)} \log(1 + |T(i)|)}{\sqrt{|N(v)| \cdot |N(u)|}}
\]  

Figure 2 shows the procedures of constructing the user similarity matrix W. The lowercase letters a, b, c, d, and e denote jobs, and the uppercase letters A, B, C, and D denote users. The algorithm firstly builds up user rating relationship based on matrix W, then calculates the user similarity matrix.

While constructing the similarity matrix W, the algorithm uses the “heat penalty” formula of Eq. 2 to impose the “heat penalty” on the popular jobs, thus reducing the phenomenon of recommending a small number of popular jobs to a large number of users.

(4) Calculate the current user’s interest in unrated jobs via Eq. 3 with the following formula.

\[
J(u, i) = \sum_{t \in P(u,k) \cap T(i)} W_{ut} r_{ti}
\]  

where \( J(u, i) \) stands for user i’s predicted interest of job u, \( P(u,k) \) represents the most top k similar users to the current user, \( T(i) \) represents the set of users who have a rating record of job i, \( W_{ut} \) represents the similarity between user u and user t in user similarity matrix W, \( r_{ti} \) represents the user t’s rating of job i.

(5) Sort the results of the operation in step (4) and select the five jobs with the highest interest prediction value to recommend to the current user.
3.2 Content-Based Algorithm

The TF-IDF algorithm is a common algorithm for extracting keywords from the text. The core idea of the TF-IDF algorithm is: words that appear very rarely in other texts, but have a high frequency in the current text, can be regarded as keywords of the current text. The TF-IDF algorithm needs to calculate the TF (Term Frequency) and the IDF (Inverse Document Frequency) respectively [5].

The formula for calculating TF (term frequency) is as follows.

$$\text{TF}_{t,D_i} = \frac{\text{count}(t)}{|D_i|}$$  \hspace{1cm} (4)

where \(\text{count}(t)\) represents the number of times that word \(t\) appears in the text, \(|D_i|\) represents the total number of words in text \(D_i\).

The formula for calculating IDF (inverse document frequency) is as follows.

$$\text{IDF}_t = \log \frac{N}{\sum_{i=1}^{N} I(t,D_i)}$$  \hspace{1cm} (5)

where \(N\) represents the total number of text, \(I(t,D_i)\) represents whether term \(t\) appears in text \(D_i\), it has a value in range \(\{0, 1\}\), if the text contains the term, the value of it is 1, otherwise, the value is 0.

3.2.1 Procedure of Content-Based Algorithm

The project uses the “jieba.analyse.extract_tags” method from the Python library “jieba”, which is good for word division, to implement the TF-IDF algorithm to extract keywords from texts.

The procedure of the algorithm is as follows.

1. Retrieve the jobs which have the same “keyword” field as the current job in the database.
2. Calculate the similarity of the columns whose value is a number or specific string value, such as “salary”, “education”.
3. Preprocess the “job_advantage” field of the job, remove the HTML tags and extract 2 keywords from it.
4. Preprocess the “job_detail” field of the job, remove the HTML tags and keywords extracted from step (3), and the words that are not relevant to the job. Finally, extract 10 keywords from it.
5. Calculate the text-similarity of the “job_advantage” and “job_detail” field.
6. Multiply the similarity of all the fields by their weights to calculate the similarity between two jobs.
7. Sort and return the 3 top jobs with the highest similarity as the result.

3.3 Analysis of Experimental Results

3.3.1 Evaluation Indicators of Recommendation System

Precision and recall are commonly used to evaluate recommendation systems [9]. Accuracy is the ratio of the number of correct job recommendations to the number of
all recommendations. The recall is the ratio of the number of correctly recommended jobs to the number of all jobs. Besides, $F$ [12] is also defined as the evaluation indicator of the recommendation system, and the formula for $F$ is as follows.

$$F = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$  \hspace{1cm} (6)

The computation time of the recommendation algorithm is also an evaluation indicator of the recommendation system.

### 3.3.2 Loss Calculation of Recommendation System

Users’ predicted ratings of jobs can be calculated by recommendation algorithm. Since there is no direct record of user ratings in the system, the real user ratings of jobs are calculated by assigning different weights to the user’s browsing and favorite records [8]. Then calculate average MAE [2] as the final loss of recommendation algorithm. The formula for calculating the MAE for each user is as follows.

$$MAE = \frac{1}{N} \sum_{v \in P} |p_{u,v} - v_{u,v}|$$  \hspace{1cm} (7)

where $N$ represents the number of recommended jobs for each user, $P$ represents the recommended jobs list for the user, $p_{u,v}$ represents user $u$’s real rating to job $v$, $v_{u,v}$ represents user $u$’s predicted rating to job $v$.

### 3.3.3 Analysis of Experimental Results

There are 503 user information records, 23508 browsing records, 3122 collecting records, and 101251 job detail information records in the system for testing. Using each recommendation algorithm to recommend 10 jobs for each user and calculate the value of $F$, calculation time, and loss.

| Table 1. The test results of each algorithm |
|--------------------------------------------|
| Indicator | Advanced co-filtering | Co-filtering | Content-based |
| F         | 0.632                 | 0.410        | 0.298         |
| Time spend| 0.364                 | 0.738        | 0.775         |
| Loss      | 1.173                 | 2.850        | 2.359         |

As shown in Table 1, the $F$-values are calculated for each of the three different recommendation algorithms, and the hybrid recommendation algorithm performs much better than the other two algorithms in terms of $F$-value. This result shows that the hybrid recommendation algorithm optimizes the recommendation effect well.
In addition, the loss values of all three algorithms are controlled between 0 and 0.6, which shows that the recommendation algorithm has a low loss. Also, with the same test results as the F value, the hybrid recommendation algorithm has the best recommendation effect.

The collaborative filtering recommendation algorithm is the one with the highest average error when calculating the results of MAE, which is because the collaborative filtering algorithm recommends calculating the similarity between users. The content-based recommendation algorithm calculates the similarity between jobs, so the content-based will recommend jobs that are similar to the jobs the user is browsing, thus having the lowest loss.

Besides, the result also shows the comparison of computation time between the optimized collaborative filtering algorithm and the traditional algorithm. The optimized algorithm computes more than one time faster than the traditional algorithm, indicating that the algorithm optimization has a better effect.

4 Conclusions

This paper introduces the use of hybrid recommendation algorithms and the advanced collaborative filtering recommendation algorithm. The use of a hybrid recommendation algorithm can improve the recommendation effect, and the optimization of the collaborative filtering algorithm solves the problem of the sparsity of the user similarity matrix. However, the dataset of this experiment is relatively small, so how to expand the dataset to make the test results more reliable, and how to use more complex recommendation algorithm to make the recommendation effect more accurate will be the focus of subsequent research.

Acknowledgements. This work was supported by WJY2018002. (Online education & teaching research project of ECUST).

References

1. Anderson, C.: The long tail: why the future of business is selling less of more. J. Prod. Innov. Manag. 24(3), 274–276 (2005)
2. Alhaj, R., Rokne, J.: Encyclopedia of Social Network Analysis and Mining. Springer, New York (2014)
3. Linden, G., Smith, B., York, K.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. 7(1), 76–80 (2003)
4. Hong, W., Zheng, S., Wang, H., et al.: Dynamic user profile-based job recommender system. In: International Conference on Computer Science and Education, pp. 1499–1503 (2013)
5. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. 17(6), 734–749 (2005)
6. Balabanovic, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Commun. ACM 40(3), 66–72 (1997)
7. Bellogin, A., De Vries, A.P.: Understanding similarity metrics in neighbour-based recommender systems. In: International Conference on the Theory of Information Retrieval, pp. 48–54 (2013)

8. Bradley, K., Rafter, R., Smyth, B., et al.: Case-based user profiling for content personalisation. In: Adaptive Hypermedia and Adaptive Web Based Systems, pp. 62–72 (2000)

9. Buckley, C., Voorhees, E.M.: Evaluating evaluation measure stability. In: International ACM SIGIR Conference on Research and Development in Information Retrieval, vol. 51, no. 2, pp. 33–40 (2000)

10. Burke, R.: Hybrid recommender systems: survey and experiments. User Model. User-Adap. Inter. 12(4), 331–370 (2002)

11. Deshpande, M., Karypis, G.: Item-based top-N recommendation algorithms. ACM Trans. Inf. Syst. 22(1), 143–177 (2004)

12. Chen, C.: The recommendation system based on two-sides selection between college graduates and employers. Southwest University of Science and Technology, SiChuan Province, pp. 51–59 (2014). (in Chinese)

13. Zhang, Y.: Graduate employment recommendation system based on collaborative filtering. Nankai University, Tianjin, pp. 1–19 (2014). (in Chinese)