# A Research of Job Recommendation System Based on Collaborative Filtering

Yingya Zhang, Cheng Yang, Zhixiang Niu
Information Engineering School
Communication University of China
Beijing, China
zyy90@aliyun.com, cafeeyang@163.com, niuzhixiang4321@126.com

***Abstract*—Dealing with the enormous amount of recruiting information on the Internet, a job seeker always spends hours to find useful ones. To reduce this laborious work, we design and implement a recommendation system for online job-hunting. In this paper, we contrast user-based and item-based collaborative filtering algorithm to choose a better performed one. We also take background information including students' resumes and details of recruiting information into consideration, bring weights of co-apply users (the users who had applied the candidate jobs) and weights of student used-liked jobs into the recommendation algorithm. At last, the model we proposed is verified through experiments study which is using actual data. The recommended results can achieve higher score of precision and recall, and they are more relevant with users' preferences before.**

***Keywords-recommendation system; item-based collaborative filtering; content-based filtering; Vector Space Model(VSM); Mahout***

## I. INTRODUCTION

The increasing usage of Internet has heightened the need for online job hunting. In the year of 2013, the amount of people who searched jobs on *www.ganji.com* is almost a billion. According to *Jobvite's* report 2014, 68% of online jobseekers are college graduates or post graduates. The key problem is that most of job-hunting websites just display recruitment information to website viewers. Students have to retrieve among all the information to find jobs they want to apply. The whole procedure is tedious and inefficient. In addition, many E-commerce websites, the most general application of recommendation algorithms, uses collaborative filtering algorithm without considering user's resume and item's properties----in this case, that means students' resume and details of recruiting information. So we proposed an improved algorithm based on item-based collaborative filtering. The aim of the present paper is to give an effective method of recommendation for online job hunting. We hope to offering students a personalized service that can help them find ideal jobs quickly and conveniently. Even so, the sparsity of user profile can be obstructive, further studies on filling users' preference matrix with implicit behaviors will be summarized in our next study.

The remainder of this paper is divided into four sections. Section 2 introduces the related work about recommendation systems. Section 3 describes the design and algorithm of our recommendation system for college students job hunting. Section 4 shows results and evaluation of experiments and tests. Section 5 concludes all aspects of the implementation.

## II. RELATED WORK

### A. Recommendation Algorithms

#### 1) Content-based filtering (CBF):

In Content-based methods[1], features of items are abstract and compared with a profile of the user's preference. In other words, this algorithm tries to recommend items that are similar to those that a user liked in the past. It is widely applied in information retrieval(IR). However it performs badly in multimedia field such as music or movie recommendation because it is hard to extract items attributes and obtain user's preference sometimes.

#### 2) Collaborative Filtering (CF):

CF is a popular recommendation algorithm that bases its predictions and recommendations on the ratings or behavior of other users in the system[2]. There are two basic type: User-based CF and Item-based CF.

- User-based CF: find other users whose past rating behavior is similar to that of the current user and use their ratings on other items to predict what the current user will like.
- Item-based CF: Rather than using similarities between users' rating behavior to predict preferences, item–based CF uses similarities between the rating patterns of items[3]. Since finding similar items is easier than finding similar users, and attributes of items are more stable than users' preference, item-based methods are suitable for off-line computing[4].

Collaborative Filtering approaches often suffer from three problems: cold start, scalability and sparsity[3].

### B. Methods of Similarity Calculation

#### 1) Cosine Similarity

Cosine similarity uses two N-dimensional vector's cosine value to indicate the degree of similarity between them. It is widely used in information retrieval(IR).

$$sim(\vec{a} \cdot \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|} \quad (1)$$

#### 2) Tanimoto Coefficient

978-1-4799-7005-6/14 $31.00 © 2014 IEEE
DOI 10.1109/ISCID.2014.228



IEEE computer society

Authorized licensed use limited to: Malta College of ArtsScience&Tech. Downloaded on May 22,2023 at 15:24:19 UTC from IEEE Xplore. Restrictions apply.

Tanimoto coefficient[5], also known as the Jaccard index, measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets.

$$Jaccard(X,Y)=\frac{X\bigcap Y}{X\bigcup Y} \tag{2}$$

*3) Log Likelihood*

Similar to Tanimoto coefficient, the Log likelihood method calculate similarity based on the common preference two users shared. Given the total number of items and the number of each user rated items, the final result is the impossibility of that the two users have such common preference[6].

*4) The City Block Distance*

The city block[7] distance is the sum of the lengths of the projections of the line segment between the points onto the coordinate axes.

$$D(x,y)=\sum_{i=0}^{n}|x_i-y_i| \tag{3}$$

The similarity calculation using the city block distance:

$$sim(x,y)=\frac{1}{1+d(x,y)} \tag{4}$$

## III. Recommendation System Of Students Job Hunting(SJH)

### A. Procedure of SJH Recommendation

There are four steps in our system as *Fig. 1* shows:

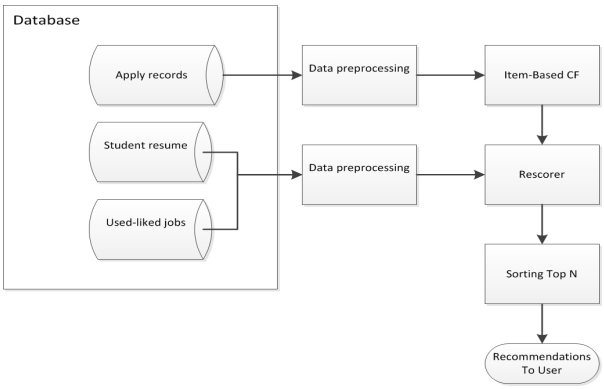


Figure. 1. Procedure of SJH recommendation

- Data Preprocessing: In this step, we clean the raw data to filter useless data including inactive users and expired recruiting information.
- Construct the item-based CF recommender: In this step, the recommender deals with the input data(apply records) and gives out original predict preference grades using item-based collaborative filtering algorithm.
- Rescorer: re-compute grades of candidate items.
- Sort all grades and push Top N jobs to users.

### B. Item-based CF Deals With Boolean Data

The traditional Item-based CF processes as follow: First, for each job which current user applied in the past(we regard user-applied jobs as user-liked jobs), find out other users who applied this job(we regard these users as co-applied users), and then find out other jobs these co-applied users also applied, except for the current job(user-liked jobs), uses these jobs as candidate set. The procedure are presented below:

*for each $job_i$ $user_i$ applied{*
*for each co-applied $user_j$ who applied $job_i${*
*find out jobs that $user_j$ applied;*
*add these jobs to candidate set;*
*}*
*delete $job_i$ from candidate set;*
*}*

Second, for every job $Item_j$ in the candidate set {Item1…$Item_p$}, compute the predict preference grade for it according to the formula (5). At last, sort all the grades and choose top N jobs as the result set. The procedure are presented below:

*for each $item_j$ in candidate set{*
*compute pref ($U_i$,$Item_j$);*
*}*
*sort these pref ($U_i$,Item);*

$$pref\ (U_i,Item_j)=\frac{\sum_{l=1}^{k} sim(Item_l,Item_j)*p(U_i,Item_l)}{\sum_{l}^{k} sim(Item_l,Item_j)} \tag{5}$$

*pref* ($U_i$,$Item_j$) stands for the predict degree that $U_i$ will rates $Item_j$, sim($Item_l$,$Item_j$) stands for similarity between $Item_l$ and $Item_j$, p($U_i$,$Item_l$) is user $U_i$'s preference on $Item_l$. $Item_l$ is from set {$Item_1$…$Item_k$} that $U_i$ has applied in the past.

In our case, users express their preference on items through apply behaviors. In this condition, p($U_i$,$Item_l$) is boolean type， values (0,1). So we use original grading formula as follow(6):

$$pref_o(U_i,Item_j)=\sum_{l=1}^{x} sim(Item_l,Item_j)*p(U_i,Item_l) \tag{6}$$

If $U_i$ had applied $Item_l$, p($U_i$,$Item_l$) equals 1. Otherwise, p($U_i$,$Item_l$) equals 0.

### C. Weight in Specific Fields

So far, our system recommends jobs to users simply according to apply records only. To improve quality of recommendation, we bring users' resumes and details of recruiting information into the algorithm.

*1) Weight of used-liked jobs*

$Item_m$ comes from the job set $U_i$ applied in the past {Item1… $Item_y$}, stands for user-used-like items. We compare the candidate job $Item_j$ to these jobs, so that we can



Authorized licensed use limited to: Malta College of ArtsScience&Tech. Downloaded on May 22,2023 at 15:24:19 UTC from IEEE Xplore. Restrictions apply.

aggravate the weights(7) of candidates that similar to user-used-liked ones.

$$w_h(Item_j) = \alpha + \frac{\sum_{m=1}^{y} sim(Item_m, Item_j)}{y} \quad (7)$$

$Item_m$ is a job that the user applied in the past. $\frac{\sum_{m=1}^{y} sim(Item_m, Item_j)}{y}$ stands for the average similarity between candidate job $Item_j$ and user-used-liked jobs. $\alpha$ equals 1.

*2) Weight of co-apply users*

We compare the users who had applied the candidate job $Item_j$ (namely co-apply users) to the current user $U_i$, then aggravate the weights(8) of candidates that has higher co-apply users' similarities.

$$w_c(U_i) = \beta + \frac{\sum_{n=1}^{z} sim(U_n, U_i)}{z} \quad (8)$$

$U_n$ is one of the co-apply users in $\{U_1 \ldots U_z\}$, $\frac{\sum_{n=1}^{z} sim(U_n, U_i)}{z}$ means the average co-apply user similarity of a candidate job.

Then, we have summarized the rescore preference grading formula as (9):

$$pref(U_i, Item_j) = pref_o(U_i, Item_j) * w_h(Item_j) * w_c(U_i) \quad (9)$$

### D. Similarity Method Dealing with Text

In student job hunting system, student resume information and job descriptions are stored in the form of text in the database. To compare the similarity between two pieces of information, we represent each piece of information as space vector and use cosine similarity distance calculation.

For example, job description is expressed as a vector like this: (job name, location, job type, field, category name). It is represent by $\vec{J} = (j_1, j_2, j_3, j_4, j_5)$; student resume is expressed as a vector like this: (college, major, degree, home place, gender). It is represent by $\vec{S} = (S_1, S_2, S_3, S_4, S_5, S_6)$.

The similarity between two jobs or two students can be calculated by the formula (10) and (11):

$$sim(J_1, J_2) = \cos(\theta_j) \quad (10)$$

$$sim(S_1, S_2) = \cos(\theta_u) \quad (11)$$

## IV. Experiments

In this section, user-based and item-based CF algorithms are tested on our data set respectively. Then item-based CF, the better performed one, to be applied on the Student Job Hunting recommendation system. At last, we evaluate the performance of improved recommender that using used-liked job and co-apply users weights based on item-based algorithm. The implementation of our experiments is based on Apache Mahout.

### A. Data Set

The data set we used is collected from a job hunting web-site during the years of 2012 and 2013, including 2,503 jobs, 7,610 students resumes, and 9,924 job apply records from 6,892 students. The structure of our data base is shown as fallow:

*1) Job apply records:*

| APPLY ID | STUDENT ID | JOB ID | APPLY DATE |
|---|---|---|---|

*2) Details of Jobs:*

| JOB ID | JOB NAME | LOCATION | TYPE | FIELD | CATEGORY NAME |
|---|---|---|---|---|---|

*3) Student resumes:*

| STUDENT ID | COLLEGE | MAJOR | DEGREE | HOME PLACE | GENDER |
|---|---|---|---|---|---|

### B. Indicators

*1) Precision*

Precision is the fraction of recommended items that are relevant to the user, as (12) shows[8]:

$$P = \frac{|hits|}{|recset|} \quad (12)$$

*2) Recall*

Recall is the fraction of the items that are relevant to the user that are successfully recommended, as (13) shows[8]:

$$R = \frac{|hits|}{|testset|} \quad (13)$$

*3) F1-Measure*

F1 score is the harmonic mean of precision and recall.

$$F1 = \frac{2 * P * R}{P + R} \quad (14)$$

### C. Contrast of User-based and Item-based Algorithm

We evaluate user-based CF and item-based CF on our data set respectively. We use Log likelihood, City Block and Tanimoto similarity methods to recommend three jobs for each user. In the user-based algorithm, the neighborhood number is ten. The precision and recall is evaluated.

TABLE I. Precision and Recall of Different Recommeders

| Recommender(r_num=3) | Similarity | Precision | Recall |
|---|---|---|---|
| **User-Based CF (n=10)** | Log likelihood | 62.82% | 53.85% |
| | City Block | 83.33% | 56.41% |
| | Tanimoto | 65.38% | 53.85% |
| **Item-Based CF** | Log likelihood | 58.33% | 58.33% |
| | City Block | 0.00% | 0.00% |
| | Tanimoto | 41.67% | 41.67% |



Authorized licensed use limited to: Malta College of ArtsScience&Tech. Downloaded on May 22,2023 at 15:24:19 UTC from IEEE Xplore. Restrictions apply.

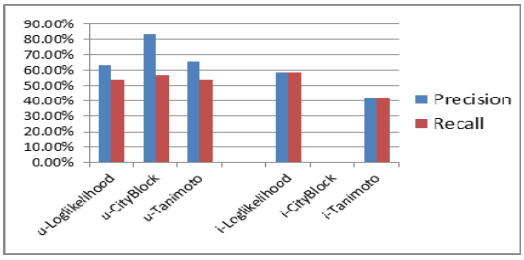


Figure. 2. Precision and recall of different recommeders

*Fig. 2* shows that all three user-based algorithm and item-based algorithm which using Log likelihood similarity reached higher precision and recall than other two algorithms. Under this circumstance, we continue to evaluate these four methods via some other variables, for example the number of neighborhood.

### *D. Evaluation Of User-based CF*

#### *1) Contrast of different similarities*

Since user's preference on jobs values (0, 1) in the job apply records, and Log likelihood, City Block and Tanimoto are three methods of similarity calculation that are suitable for boolean data, in this experiment we recommended three items to test precision and recall with these three methods. We chose different neighborhood numbers to reduce its influence.

TABLE II. PRECISION OF USER-BASED CF WITH DIFFERENT SIMILARITY METHODS

| Precision | Log likelihood | City Block | Tanimoto |
|---|---|---|---|
| **n=5** | 75.00% | 94.44% | 77.78% |
| **n=10** | 62.82% | 83.33% | 65.38% |
| **n=20** | 56.41% | 71.79% | 56.41% |
| **n=40** | 56.41% | 71.79% | 56.41% |

TABLE III. RECALL OF USER-BASED CF WITH DIFFERENT SIMILARITY METHODS

| Recall | Log likelihood | City Block | Tanimoto |
|---|---|---|---|
| **n=5** | 53.85% | 56.41% | 53.85% |
| **n=10** | 53.85% | 56.41% | 53.85% |
| **n=20** | 56.41% | 58.97% | 56.41% |
| **n=40** | 56.41% | 64.10% | 56.41% |

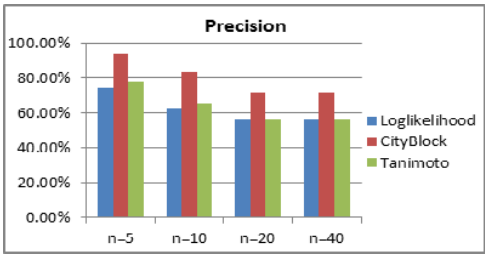


Figure. 3. Precision of user-based CF with different similarity methods

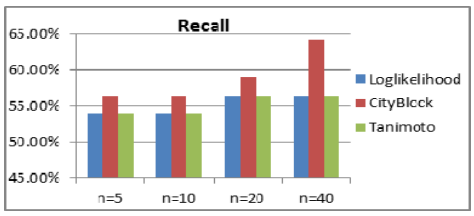


Figure. 4. Recall of user-based CF with different similarity methods

As *Fig.3* and *Fig.*4 described, the result indicates that City Block similarity can achieve higher precision and recall under different neighborhood number situations.

It should be noted that the *Generic Recommender IR Stats Evaluator* that Mahout offered just focuses on users who have recommended items, while automatically ignoring users that can not be recommended. So the result of evaluation seemed very good. In next section we will test the recommendation results artificially.

#### *2) Similarity between used-liked jobs and recommended results*

We chose ten students randomly and recommended two jobs for each one. The algorithm we used is user-based CF. The City Block similarity method is applied when computing similarities between users to divide neighborhood. N stands for the number of neighborhood. The result is displayed in Table IV, The former value before '/' is the average similarity between recommended jobs from original recommender and used-liked jobs. The latter value after '/' is the average similarity between recommended jobs from improved recommender and used-liked jobs. The improved recommender added weights of co-apply users and used-liked jobs.

TABLE IV. SIMILARITY BETWEEN USED-LIKED JOBS AND RECOMMENDED RESULTS FROM USER-BASED CF

| | Student ID | n=200 | n=100 | n=50 | n=30 | n=20 | n=5 |
|---|---|---|---|---|---|---|---|
| **1** | 4e585f8488ca47e9993891612a4cf45d | 0.51/0.51 | 0.48/0.48 | 0.39/0.39 | 0.39/0.39 | NaN/NaN | NaN/NaN |
| **2** | 8391a03fb0c8488ea5f6c7379d658e72 | 0.37/0.37 | 0.37/0.37 | 0.34/0.34 | 0.34/0.34 | NaN/NaN | NaN/NaN |
| **3** | 0104646004f2434b951ae3300d0f4396 | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN |
| **4** | c4dc0c6c3a2b4facb3358b463186f170 | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN |
| **5** | 8554271e58594c4aaa248e2495de24c9 | 0.65/0.65 | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN |
| **6** | d07a838d619f41f39a11c06a8311174f | 0.41/0.41 | 0.41/0.41 | 0.55/0.55 | 0.55/0.55 | 0.55/0.55 | NaN/NaN |
| **7** | 6ed1faf8eee9443c8b47b42ee092b4e0 | 0.42/0.42 | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN |
| **8** | 0c912ce344004f9da634d52eb2131d49 | 0.65/0.65 | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN |
| **9** | d3ad698574874d0ebb4ff448d3d82dd4 | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN |
| **10** | 7bf24dedbf0c422e9eb03f26cfe10123 | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN | NaN/NaN |



Authorized licensed use limited to: Malta College of ArtsScience&Tech. Downloaded on May 22,2023 at 15:24:19 UTC from IEEE Xplore. Restrictions apply.

*NaN*(Not a Number) means there is no valid recommendation result on current user. Due to the apply records are very sparse, the user-based recommender can not offer valid recommendation service for every user. Thus it can be seen in Student Job Hunting system, user-based algorithm is unsatisfactory.

### *E. Evaluation of Item-based CF*

According to the result of section IV.C, when using Log likelihood similarity method, item-based algorithm performed well. So we decided to use item-based CF and selected Log likelihood method to compute candidate items' similarities in the Student Job Hunting recommendation system.

*1) The performance of improved recommender: We evaluate the capability of original item-based recommender and the improved recommender that takes co-apply users' weight and used-liked jobs' weight into account when recommending two or three jobs for each student. The r_num stands for number of recommended items. The results are recorded in Table V and Table VI.*

TABLE V. PERFORMANCE OF IMPROVED RECOMMENDER IN SJH SYSTEM(R_NUM=3)

| r_num=3 | Precision | Recall | Reach | F1-measure |
|---|---|---|---|---|
| **original recommender** | 50.62% | 47.13% | 93.10% | 48.81% |
| **improved recommender** | 51.85% | 48.28% | 93.10% | 50.00% |

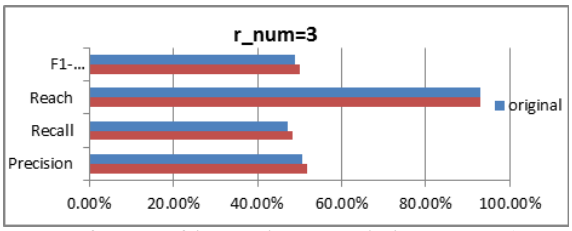

Figure. 5. Performance of improved recommender in SJH system(r_num=3)

TABLE VI. PERFORMANCE OF IMPROVED RECOMMENDER IN SJH SYSTEM(R_NUM=2)

| r_num=2 | Precision | Recall | Reach | F1-measure |
|---|---|---|---|---|
| **original recommender** | 40.91% | 37.50% | 91.67% | 39.13% |
| **improved recommender** | 59.09% | 54.17% | 91.67% | 56.52% |

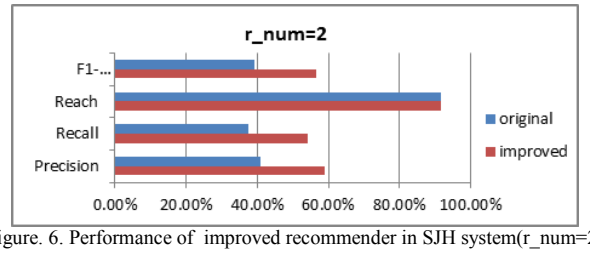

Figure. 6. Performance of improved recommender in SJH system(r_num=2)

As *Fig.5* showed, when recommending three jobs for each student, the improved recommender had a little promotion at precision, recall and F1 score. When the number of recommended item came to two, as *Fig.6* showed, all these three indicator score increased significantly. And the R*each* rate remains as before. Because the sparseness of our apply records dataset, recommender can only offer 3 or 4 recommended results for some students. To evaluate the overall recommender's performance, we considered the number of recommended items as three and two. If user U has three recommended jobs---(job 1, job 2, job 3), when recommending jobs for U, the improved recommender just re-ranking the three jobs recommended from the traditional recommender, so that it has no influence on the precision, recall and F1 score. However, when evaluating the Top 2 recommended jobs, the improved recommender changed the order of these three jobs, so that the top 2 jobs are different from former ones. The increased scores suggest that the improved recommender works well for the reason that jobs take precedence (1st and 2nd recommended jobs) are better than latter ones (3rd recommended job).

*2) Similarity between user's preference and recommendation results from improved recommender:* We chose ten students randomly and recommended five jobs for each one. The algorithm we used is improved item-based CF. The Log likelihood similarity method is applied when computing similarities between candidate items and current item. Then we compare the recommended jobs and jobs that user used to apply. The average similarities between them are displayed in Table VII as follow:

TABLE VII. SIMILARITY BETWEEN USER'S PREFERENCE AND RECOMMENDATION RESULTS IN SJH SYSTEM

| Item-Based (Log likelihood) | Student ID | Original Item-based CF | Improved Item-based CF |
|---|---|---|---|
| **1** | 9dd90881b2e040d69d5917e4a0a8acde | 0.36 | 0.57 |
| **2** | f6b073ecbdd84f18a98ed20bb33faf6f | 0.30 | 0.48 |
| **3** | f71ec48f99124875bf24dac64118d6d9 | 0.23 | 0.55 |
| **4** | dd7f431ba21545d8aab4123327416100 | 0.26 | 0.60 |
| **5** | aa7d690df4e841b79284c4f196ee2767 | 0.25 | 0.35 |
| **6** | f7a02b9efff7489bb1b57a2d01357a01 | 0.37 | 0.37 |
| **7** | 0036e442801349d68d3c49e15e71d016 | 0.40 | 0.80 |
| **8** | 001bab4efaa54988acb08fc56405f8db | 0.33 | 0.40 |
| **9** | 35d36ba8617244829d8ab9531503329b | 0.24 | 0.36 |
| **10** | 4f63a6ffe4e04258ba9e7a2de87c2401 | 0.36 | 0.66 |



Authorized licensed use limited to: Malta College of ArtsScience&Tech. Downloaded on May 22,2023 at 15:24:19 UTC from IEEE Xplore. Restrictions apply.

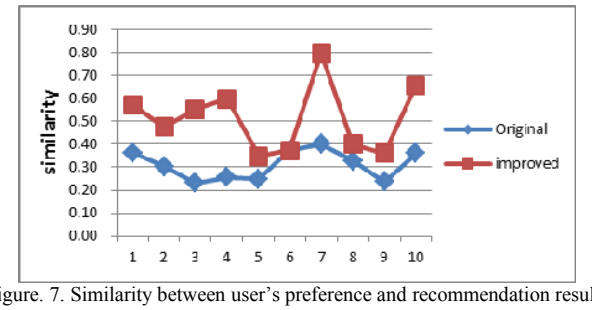

Figure. 7. Similarity between user's preference and recommendation results in SJH system

*Fig.7* shows that using improved item-based algorithm, the recommended results have higher similarity with user used applied jobs. That means they are more relevant with user's preference before.

## V. CONCLUSION

On the basis of chapter 4, we evaluated user-based CF and item-based CF with different similarity calculation methods. Finally, we selected item-based CF algorithm for its better performance considering various factors. In addition, we also take co-apply users and user apply records in the past into account. The test results indicate the improved recommender with a rescorer is better than a traditional item-based one. Our student job hunting recommender achieved higher precision, recall and F1 score. Furthermore, the recommended jobs are more relevant with students' preferences.

To further optimize the recommendation system and ameliorate the sparsity of user profile, some methods of filling users' preference matrix can be utilized, for example, take advantage of students' implicit behaviors in process of job hunting, which need further research.

## ACKNOWLEDGMENTS

The work on this paper was supported by National Key Technology R&D Program of China (2012BAH17F01-01, 2012AA011702-02, 2012BAH37F03, 2012BAH02B03), the National Culture S&T Promotion Program (WHB1002) and the SARFT Scientific and Technological Project (2012-20,2011-34,2012-27). Supported by Program for New Century Excellent Talents in University.

## REFERENCES

[1] Pazzani M J, Billsus D. Content-based recommendation systems[M]//The adaptive web. Springer Berlin Heidelberg, 2007: 325-341.

[2] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions[J]. Knowledge and Data Engineering, IEEE Transactions on, 2005, 17(6): 734-749.I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[3] Schafer J B, Frankowski D, Herlocker J, et al. Collaborative filtering recommender systems[M]//The adaptive web. Springer Berlin Heidelberg, 2007: 291-324.

[4] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th international conference on World Wide Web. ACM, 2001: 285-295.

[5] Jannach D, Zanker M, Felfernig A, et al. Recommender systems: an introduction[M]. Cambridge University Press, 2010.

[6] Anil R, Dunning T, Friedman E. Mahout in action[M]. Manning, 2011.

[7] Dunning T. Accurate methods for the statistics of surprise and coincidence[J]. Computational linguistics, 1993, 19(1): 61-74.

[8] Liang Xiang. Recommendation system in practice[J]. 2012.



Authorized licensed use limited to: Malta College of ArtsScience&Tech. Downloaded on May 22,2023 at 15:24:19 UTC from IEEE Xplore. Restrictions apply.