Collaborative Filtering Algorithm Based on Rating Difference and User Interest

Haihui Huang
Software Engineering
Chongqing University of Posts and Telecommunications
Chongqing, China
e-mail: huanghh@cqupt.edu.cn

Guo Yu
Software Engineering
Chongqing University of Posts and Telecommunications
Chongqing, China
e-mail: yuguo.hugo@qq.com

Xin Wang
Communication and Information Engineering
Chongqing University of Posts and Telecommunications
Chongqing, China
e-mail: wxinedu@foxmail.com

***Abstract*—Collaborative filtering algorithm is one of widely used approaches in daily life, so how to improve the quality and efficiency of collaborative filtering algorithm is an essential problem. Usually, some traditional algorithm focuses on the user rating, while they don't take the user rating differences and user interest into account. However, users who have little rating difference or have a similar interest may be highly similar. In this paper, a collaborative filtering algorithm based on scoring difference and user interest is proposed. Firstly, a rating difference factor is added to the traditional collaborative filtering algorithm, where the most appropriate factor can be obtained by experiments. Secondly, calculate the user's interest by combining the attributes of the items, then further calculate the similarity of personal interest between users. Finally, the user rating differences and interest similarity are weighted to get final item recommendation and score forecast. The experimental results on data set shows that the proposed algorithm decreases both Mean Absolute Error and Root Mean Squared Error, and improves the accuracy of the proposed algorithm.**

Keywords-collaborative filtering; user similarity; rating differences; user interest

I. INTRODUCTION

With the rapid development of Internet technology, cloud computing and smart phone, information overload has become a major issue in life, which make users hardly handle a wide variety of important information effectively [1]. However, the recommendation system has been shown to help users filter out unwanted information and make reasonable choices. Therefore, many large shopping websites use various recommendation algorithms to provide convenience so that they can attract customers. The examples of recommendation algorithms are content-based recommendation, recommendation of association rules, Collaborative filtering recommendations, etc. [2] Among the above example, the collaborative filtering algorithm is one of the most widely used and the most successful algorithms in the recommended field by far [3]. It mainly includes item-based, user-based and model-based collaborative filtering algorithms [4]. The idea of user-based collaborative filtering algorithms is to find out the users with similar ratings to the target users according to historical data and its rating data, and to take them as neighbors, then organize them into a sorted directory according to their favorite objects. The core of collaborative filtering algorithm is to calculate the similarity between users [5], [6]. So Increasing accuracy for the similarity calculation leads to a more effective and efficient recommendation system.

Recently, many scholars have proposed to improve the algorithm based on different perspectives: Kaleli [7] introduced a novel entropy-based neighbor selection approach, which assigned a degree of uncertainty for each user, and the approach solves the optimization problem of gathering the most similar entities with a minimum entropy difference within a neighborhood. Wu et al. [8] proposed the trapezoidal fuzzy scoring model to calculate the similarity between users in the model, which optimized the data sparse problem and reduced the running time. P Pirasteh et al. [9] proposed a collaborative filtering algorithm that combines user activities and content to effectively optimize data sparsity. G Pitsilis used user ratings to calculate a hypothetical trust between users and improves the efficiency of traditional collaborative filtering algorithms [10]. X Tang et al. [11] considered several factors of trust, proposed a trust model of trust recommendation, and optimized the recommended cold start problem. Pera M S et al. [12] considered the impact of project popularity on recommendation, and integrated popularity into the calculation of traditional similarity to improve the accuracy of recommendation. To a certain extent, these researches alleviate the problem of sparsity and cold start of data to the algorithm, improve the performance of the recommended system.

However, when calculating the user similarity, the above algorithms don’t take user interest into account, and neglect the influence of different rating standards between users on the similarity calculation. Therefore, a new method of

978-1-5386-4301-3/18/$31.00 ©2018 IEEE

similarity calculation is proposed based on the rating difference and user interest will be introduced in this paper.

II. RELATED WORK

A. Collaborative Filtering Algorithm Steps

Fig.1 is the basic process of traditional collaborative filtering algorithm. Collaborative filtering algorithm can be split into the following four steps [13]:

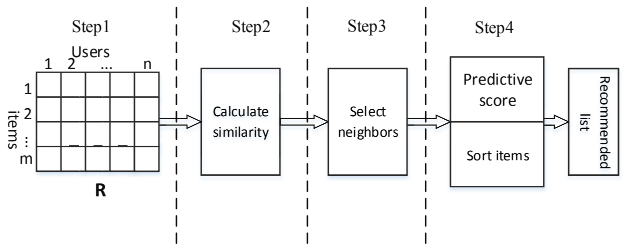

Figure 1. Collaborative filtering recommendation algorithm basic steps.

Step1: Establish the user-item rating matrix R (m users and n items) in the data set;

Step2: Calculate similarity of user-item rating matrix R using the user similarity algorithm to get a user similarity matrix;

Step3: Sort the user similarity matrix from big to small, and select some users as the nearest neighbors of the target user [14]

Step4: Use the score prediction algorithm to calculate the result of Step3, predict the target user rating of the unknown item and recommend it.

B. Traditional Algorithms of Calculating Similarity

When calculating similarity between users, there are mainly five kinds of Similarity algorithms, including Euclidean Distance, Jaccard Similarity, Cosine Similarity (COS), Adjusted Cosine(ACOS), Person Similarity (Pearson Correlation Coefficient, PCC) [15].

- Euclidean Distance. Euclidean distance refers to the distance between two points in space. The rating between two users can also be expressed as the distance between users, and the closer the distance between users is, the higher the similarity between users is. Similarity between user a and user b is expressed by the Euclidean distance $SimE(\mathrm{a,b})$:

$$SimE(\mathrm{a,b}) = \frac{1}{1+\sqrt{(\sum_{i\in I_{a,b}}(\mathrm{R}_{a,i}+\mathrm{R}_{b,i})^2)}} \quad (1)$$

where, $R_{a,i}$ and $R_{\mathrm{b},i}$ respectively are the rating of user a and b on items i respectively, and $I_{a,\mathrm{b}}$ denotes the set that user a and b co-evaluated.

- Jaccard Similarity. In the user rating matrix, there are many potential similarities between users who do not have a common rating item [16] [17]. The traditional method is to calculate the similarity of co-rating between users, but there is no way to find out the potential similarities between users. As a result, some users might not provide accurate predictions though they are high similarity. Therefore, Jaccard similarity is introduced to solve the problem of the traditional algorithms in the recommended system.

$$SimJ(\mathrm{a,b}) = \frac{|I_a \cap I_b|}{|I_a \cup I_b|} \quad (2)$$

where, a and b are the count of rating item sets of user a and b, respectively.

- Cosine Similarity. Cosine Similarity calculates the cosine of two vectors in space to evaluate its similarity. The closer the cosine value is to 1, the smaller the angle is, the closer the two vectors are. The user rating can be expressed as a space vector, the larger the cosine value is, the higher the user similarity is.

$$SimC(\mathrm{a,b}) = \frac{\sum_{i\in I_{a,b}}\mathrm{R}_{a,i}\mathrm{R}_{b,i}}{\sqrt{\sum_{i\in I_a}R_{a,i}^2}\sqrt{\sum_{i\in I_b}R_{b,i}^2}} \quad (3)$$

where, $R_{a,i}$ and $R_{\mathrm{b},i}$ are rating of user a and b on items i respectively, and $I_{a,\mathrm{b}}$ denotes the set that user a and b co-evaluated.

- Adjusted Cosine. There are different criteria in the user rating process, some users tend to rate high scores, while others prefer low scores. The cosine similarity does not consider the factor of the user rating scale, so the Adjusted Cosine similarity (a, b) is modified by subtracting the user average score from the Cosine similarity.

$$SimC'(\mathrm{a,b}) = \frac{\sum_{i\in I_{a,b}}(\mathrm{R}_{a,i}-\overline{R_a})(\mathrm{R}_{b,i}-\overline{R_b})}{\sqrt{\sum_{i\in I_a}(R_{\mathrm{a},i}-\overline{R_a})^2}\sqrt{\sum_{i\in I_b}(R_{b,i}-\overline{R_b})^2}} \quad (4)$$

where, $\overline{R_a}$ represents the average of the user ratings on item I_a , $\overline{R_\mathrm{b}}$ represents the average of the user ratings on item I_b .

- Pearson similarity. Person similarity is the same as the modified cosine similarity numerator, while the denominator is changed from the original, user personal item set to a, b common item set $I_{\mathrm{a,b}}$, $SimP(\mathrm{a,b})$ denotes Person similarity:

$$SimP(\mathrm{a,b}) = \frac{\sum_{i\in I_{a,b}} (\mathrm{R}_{a,i} - \overline{R_a})(\mathrm{R}_{b,i} - \overline{R_b})}{\sqrt{\sum_{i\in I_{a,b}} (R_{a,i} - \overline{R_a})^2}\sqrt{\sum_{i\in I_{a,b}} (R_{b,i} - \overline{R_b})^2}} \quad (5)$$

C. Predict Target User Ratings

After user similarity is calculated and N neighbor set of the target user is selected, it combines with the similarity between the users and the neighbor rating of the items to calculate and predict the target user rating on the items [18]. The calculation formula is defined as follows:

$$P_{a,c} = \overline{R_a} + \frac{\sum_{c\in N_a} Sim(a,c)\times(R_{c,i} - \overline{R_c})}{\sum_{c\in N_a} |Sim(a,c)|} \quad (6)$$

where, $\overline{R_a}$ and $\overline{R_\mathrm{b}}$ are the average rating of target users and neighbors, N_a is the N neighborhoods set of target user A.

III. The Proposed Algorithm

A. User Rating Similarity

The rating of the user is shown in Table 1, in which there are ratings of users and items. Where user is $U = \{\mathrm{u}_1, \mathrm{u}_2, \cdots, \mathrm{u}_m\}$, movie is $I = \{i_1, i_2, \cdots, i_n\}$, and a user-item score matrix is generated from the user rating table. Given that some of the rating matrices are empty, these missing values are replaced by the symbol?.

TABLE I. User Rating Matrix

	I1	I2	I3	I4	I5
U1	5	3	4	3	3
U2	4	?	?	?	1
U3	1	?	1	?	4

In the experiment, Euclidean distance, Jaccard similarity, Cosine similarity, Adjusted Cosine and Person similarity are respectively used to find the user similarity matrix. Usually, the algorithm with the lowest MAE is selected as the initial user similarity algorithm.

B. User Rating Difference

The traditional collaborative filtering algorithms ignore such a phenomenon, in which two users rated on multiple items differently, but there is high similarity between the users. One of the reasons is some users tend to rate high when they rate an item, while others tend to rate low. In fact, although the two users are similar, the results calculated by traditional algorithms can be quite different. We assume there are two users u_1, u_2, as shown in Fig. 2, and their rating vectors are $\vec{\mathrm{r}_1}$, $\vec{\mathrm{r}_2}$. The output is going to be different between u_1 and u_2 with traditional algorithms, however, it is obvious that $\vec{\mathrm{r}_1}$ and $\vec{\mathrm{r}_2}$ rating difference vectors are the same, that is, the users are also similar in some respects. If a large amount of data only is calculated the user's similarity by the cosine similarity, then there will be a large number of similarity errors, Therefore, It is better to add the factors of the user rating differences in the calculation of the similarity, which can improve user accuracy [19].

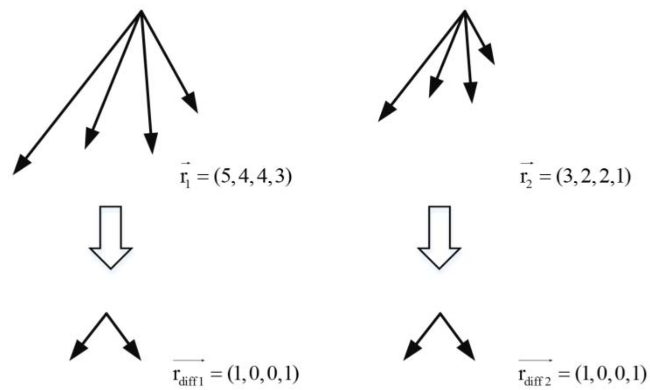

Figure 2. Similarity of difference rating vectors.

When calculating the user rating difference, a user rating difference factor is added to the cosine similarity calculation. Firstly, the rating difference is calculated between user A and B, as follows:

$$sumDiffer(u_a, u_b) = \sqrt{\sum_{i\in I_{a,b}} (\mathrm{R}_{a,i} - \mathrm{R}_{b,i})^2 / M} \quad (7)$$

The user rating difference factor as follows:

$$\omega(\mathrm{u}_a, \mathrm{u}_b) = \lambda^{sumDiffer(\mathrm{u}_a, \mathrm{u}_b)}, 0 < \lambda < 1 \quad (8)$$

where, $sumDiffer(u_a, u_b)$ denotes the rating difference between user u_a and u_b, $I_{a,b}$ is the common rating of user u_a and u_b, and M is the count of $I_{a,b}$. $\omega(\mathrm{u}_a, \mathrm{u}_b)$ is the rating difference factor between user u_a and u_a, and λ($0 < \lambda < 1$) is the weight of the score difference factor. Next, the rating difference factor is added to the traditional algorithm to get the improved algorithm, so that to get the Collaborative filtering algorithm base on rating difference.

$$SimD(\mathrm{u}_a, \mathrm{u}_b) = Sim(\mathrm{u}_a, \mathrm{u}_b) * \omega(\mathrm{u}_a, \mathrm{u}_b) \quad (9)$$

where, $Sim(\mathrm{u}_a, \mathrm{u}_b)$ is the traditional similarity algorithm, $SimD(\mathrm{u}_a, \mathrm{u}_b)$ is improved algorithm, which includes rating difference factor.

C. User Interest Similarity

In reality, the user rating of the items is not only correlated to user similarity, but also to user interest, because the user interest also occupies a certain proportion. Assumed that user u_a loves movie i_p, user u_b loves movie i_q. Obviously, the similarity between the two is 0 according to COS. However, movie i_p and movie i_q belong to the same category of [adventure, comedy], so that user u_a and users u_b like [adventure, comedy] genre, which means they have a high level of similarity. Therefore, the similarity of the user interest to the item genre needs to be taken into account when calculating the user similarity, so that to further improve the accuracy of user similarity [20].

With this, the two users similarity of interest is high when they like the same genre of a movie. The user interest in movie genres can be calculated as follows:

$$H_{u,\mathrm{t}} = N_{u,\mathrm{t}} / N_u \tag{10}$$

where, $N_{u,\mathrm{t}}$ is the count of the ratings of user u for movie genre t, N_u is the count of ratings of user u for all movie genres, and $H_{u,\mathrm{t}}$ is the user interest in movie genre t.

The user interest vector is $H_u = \{\mathrm{H}_{u,1}, \mathrm{H}_{u,2}, \cdots, \mathrm{H}_{u,\mathrm{n}}\}$, and the user interest similarity is calculated as follows:

$$SimI(\mathrm{u}_a, \mathrm{u}_b) = \frac{\sum_{t \in n} H_{u_a,\mathrm{t}} H_{u_b,\mathrm{t}}}{\sqrt{\sum_{t \in n} H_{u_a,\mathrm{t}}} \sqrt{\sum_{t \in n} H_{u_b,\mathrm{t}}}} \tag{11}$$

where, n is the count of all movie genres.

Based on the above formula, combined with equation (6) ~ (10), we get the formula for calculating the user similarity based on the difference between rating and user interest as follows:

$$\begin{aligned} Sim\_DI(\mathrm{u}_a, \mathrm{u}_b) = {} & SimD(\mathrm{u}_a, \mathrm{u}_b) * \alpha \\ & + Sim\_\mathrm{I}(\mathrm{u}_a, \mathrm{u}_b) * (1 - \alpha) \end{aligned} \tag{12}$$

where, $\alpha \in (0,1)$, which is the weight of similarity of users interest in the whole users similarity. Through experiments, we can get the most appropriate α, so that to get the more accurate and more convincing user similarity.

D. The Algorithm Steps

Base on all above the formulas, the proposed algorithm for Collaborative filtering algorithm can be concluded as the following steps:

Step1: Input user-item-score training table, establish a user-item rating matrix;

Step2: Calculate the similarity between the users by using the traditional algorithm to get the user similarity matrix;

Step3: Calculate the difference between users and the difference factor, then add the factor to the result of the step2.

Step4: Input the item attributes table, establish a user-item-genre matrix. Then calculate the interest of each user, and further get user interest similarity.

Step5: Combining the user rating similarity matrix of the step3 and the user interest similarity matrix of the step4, the user comprehensive similarity matrix is obtained by weighted calculation;

Step6: Obtain the user top-N neighbor set according to the user's comprehensive similarity matrix;

Step7: Calculate the user's rating forecasting set according to the top-N neighbor set.

Fig. 3 represents the flow-chart of we proposed algorithm.

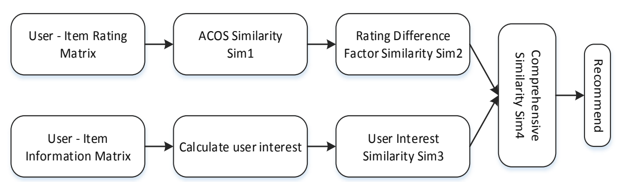

Figure 3. The proposed algorithm steps.

IV. EXPERIMENTAL RESULTS

This section will first introduce the experimental environment and data sets, and then introduce the experimental evaluation index, and finally, explain the results and compare traditional algorithm by the following experimental.

A. Experimental Environment and Datasets

The laptop configuration used in the experiment has a Core i5-2450M, dual-core, 2.5GHz, 4GB DDR3 memory, Window7 operating system, the programming language is Python, the version is Python3.6, and the development tool is Anaconda3.

The MovieLens dataset, collected by the GroupLens research project, includes 943 users with more than 100,000 ratings for 1682 movies. In this data set, the user rating in the range of 1–5, and the sparsity of the data is 93.7 %. Movie data information includes movie Id, name, issue date and movie genre, etc. Moreover, the movie genre includes 19 genres in total, and each movie could have multiple genres. In the experiment, 80% of the data was selected randomly as part of the training set and the remaining 20% as part of the test set.

B. Metrics

The proposed algorithm is mainly to improve the accuracy of the recommendation algorithm. In order to measure the performance of the algorithm, we use Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to evaluate the proposed algorithm [21]. The MAE and RMSE can be obtained by comparing the ratings between the user experimental prediction rating and the user real rating. The lower the MAE and RSME value, the better the performance of the recommended algorithm. The user predictive rating set by the recommended algorithm is $\{p_1, p_2, \cdots, p_n\}$, the user real rating set is $\{r_1, r_2, \cdots, r_n\}$, and n is the count of predictive items. Formally:

$$MAE = \frac{\sum_{i \in n}^{n} |p_i - r_i|}{n} \tag{13}$$

$$RMSE = \sqrt{\frac{\sum_{i \in n}^{n} (p_i - r_i)^2}{n}} \tag{14}$$

C. Comparison of Traditional Similarity Algorithm

Firstly, the most suitable algorithm was found as the initial algorithm to improve the algorithm by comparing the accuracy of several traditional similarity algorithms. The values of MAE and RSME were calculated from Euclidean distance, Jaccard similarity, Cosine Similarity, Adjust Cosine Similarity, Person Similarity respectively.

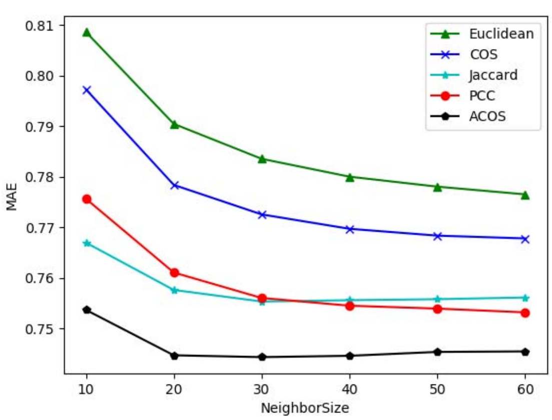

Figure 4. MAE comparison of traditional algorithms.

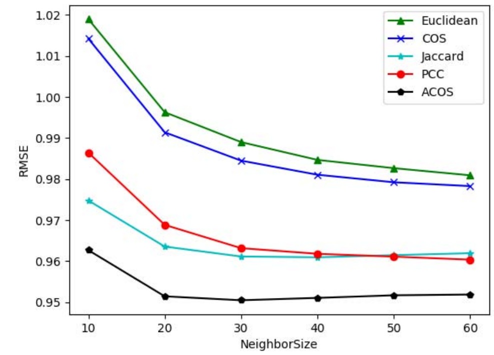

Figure 5. RMSE comparison of traditional algorithms.

The experimental results are shown in Figs. 5 to 6, in which show that the ACOS outperforms than others, so ACOS is used as a benchmark algorithm to improve the algorithm.

D. User Rating Difference Factor Weight λ

λ={0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9} , the number of neighbors N={10,20,30,40,50,60} , the result is obtained under different conditions are compared experimentally to determine the most appropriate λ .

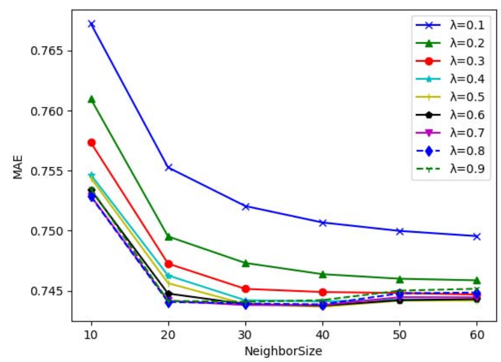

Figure 6. MAE under different λ .

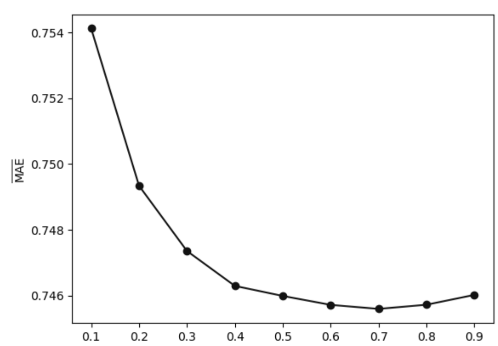

Figure 7. The mean of MAE under different λ .

It can be seen from Fig. 5, the value of λ is not the bigger the better, you need to choose a most appropriate λ.Fig. 6 shows the mean of different MAE. It can be seen that $\overline{MAE}$ is at the lowest point when λ =0.7, and the performance of the recommended algorithm is the best. Therefore, the improved algorithm based on user rating difference will generate a more accurate result when λ is 0.7.

E. The Weight of Interest Similarity α

User interest similarity occupies a certain proportion in the entire user similarity, that is, the weight α . So it is necessary to determine the most accurate to calculate the user similarity. α={0.1,0.2,0.3,0.4,0.5,0.6} , the result is shown as in Fig. 7.

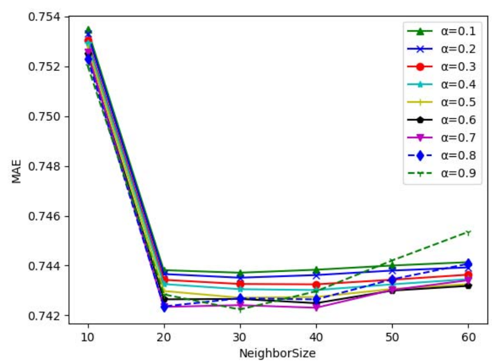

Figure 8. MAE under different α .

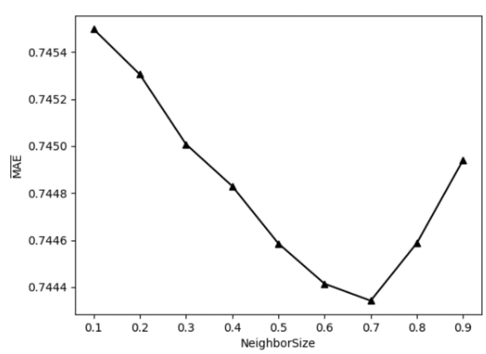

Figure 9. The mean of MAE under different α .

Fig. 9 is a graph of average MAE with different in the neighborhood of 10 to 60.

Similarly, as can be seen from the figure, the $\overline{MAE}$ is at lowest point and the performance of the algorithm is the best when α =0.7. Therefore, the weight of the algorithm α is 0.7 in this paper.

F. Algorithm Comparative Analysis

In order to verify the performance of the proposed algorithm in this paper, we compared the value of MAE, RMSE from ACOS, DCF (rating Differences-CF), ICF (Interest-CF) and DI-CF (Differences and Interest similarity-CF). The experimental results are shown in the following figure:

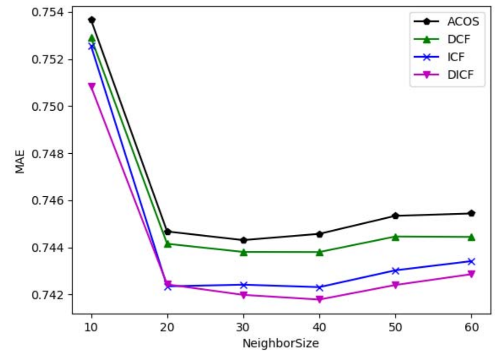

Figure 10. MAE of the algorithms comparison.

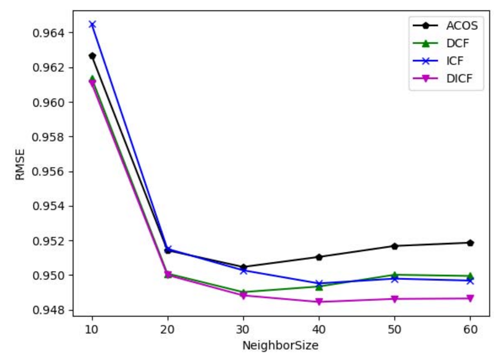

Figure 11. RMSE of the algorithms comparison.

From the figure above, the values of MAE and RMSE of DCF, ICF and DICF in this paper are smaller contrast with the traditional ACOS algorithm, which means that the proposed algorithm is more accurate than ACOS. The reason is that we add the rating difference factor and user interest similarity in the traditional algorithm, so as to get more accurate and reliable collaborative filtering algorithm. Meanwhile, when the number of neighbors is 10, the values of MAE and RMSE are large because of the data sparse problem due to insufficient neighbor size. However, the proposed algorithm still outperforms the traditional algorithm even when it faces data sparse problem.

V. CONCLUSION

Through several experimental schemes, this paper analyzes the performance of collaborative filtering recommendation algorithm based on user rating difference and user interest. Firstly, the design ideas and algorithm steps are introduced, and then rating difference factor and user interest are taken into the traditional similarity algorithm. Secondly, several parameters of the algorithm are determined through several experiments. Finally, the improved algorithm proposed in this paper is compared with the traditional one. The improved algorithm proposed not only improves the accuracy in general situations, but also generates a better result under the condition of sparse data. The disadvantage of the experiment is that the experimental parameters may not reach the best results under different datasets. So, how to make the recommended algorithm automatically adjust the parameters according to different data sets is the next direction of efforts.

ACKNOWLEDGMENT

The authors acknowledge support from the National Natural Science Foundation (Grant No. 61772099); the Program for Innovation Team Building at Institutions of Higher Education in Chongqing (Grant No.CXTDG201602010); the University Outstanding Achievements Transformation Funding Project of Chongqing (Grant No. KJZH17116); the Artificial Intelligence Technology Innovation Important Subject Projects of Chongqing (cstc2017rgzn-zdyf0140); The Innovation and Entrepreneurship Demonstration Team Cultivation Plan of Chongqing (cstc2017kjrc-cxcytd0063); the China Postdoctoral Fund (Grant No.2014M562282); the Project Postdoctoral Supported in Chongqing (Grant No. Xm2014039); the Wenfeng Leading Top Talent Project in CQUPT; the New Research Area Development Programme (Grant No. A2015-44); the Science and Technology Research Project of Chongqing Municipal Education Committee (Grant No. KJ1400422); the Common Key Technology Innovation of Important Industry by Chongqing Science and Technology Commission (Grant No. CSTC2015ZDCY-ZTZX40001); the Collaborative Innovation Center for Information Communication Technology (Grant No. 002).

REFERENCES

[1] Choi K, Suh Y. A new similarity function for selecting neighbors for each target item in collaborative filtering[J]. Knowledge-Based Systems, 2013, 37(1):146-153.

[2] Zhang X, Li Y. Use of collaborative recommendations for web search: an exploratory user study[M]. Sage Publications, Inc. 2008.

[3] Park Y, Park S, Jung W, et al. Reversed CF: A fast collaborative filtering algorithm using a k -nearest neighbor graph[J]. Expert Systems with Applications, 2015, 42(8):4022-4028.

[4] Zhang J, Lin Y, Lin M, et al. An effective collaborative filtering algorithm based on user preference clustering[J]. Applied Intelligence, 2016, 45(2):230-240.

[5] Wen J H, Shu S. Improved Collaborative Filtering Recommendation Algorithm of Similarity Measure[J]. Computer Science, 2014, 41(5):68-71.

[6] Kaleli C. An entropy-based neighbor selection approach for collaborative filtering[J]. Knowledge-Based Systems, 2014, 56(C):273-280.

[7] Kaleli C. An entropy-based neighbor selection approach for collaborative filtering[J]. Knowledge-Based Systems, 2014, 56(C):273-280.

[8] Yi-Tao W U, Zhang X M, Wang X M, et al. User fuzzy similarity-based collaborative filtering recommendation algorithm[J]. Journal on Communications, 2016.

[9] Pirasteh P, Jung J J, Hwang D. Item-Based Collaborative Filtering with Attribute Correlation: A Case Study on Movie Recommendation[M]// Intelligent Information and Database Systems. Springer International Publishing, 2014:245-252.

[10] Pitsilis G, Knapskog S J. Social Trust as a solution to address sparsity-inherent problems of Recommender systems[J]. Computer Science, 2012.

[11] Tang X, Chen M. Reputation-based recommendation trust model in the interoperable environment[C]// International Conference on Electronics, Communications and Control. IEEE, 2011:2226-2228.

[12] Pera M S, Ng Y K. A group recommender for movies based on content similarity and popularity[J]. Information Processing & Management, 2013, 49(3):673-687.

[13] Wang Y, Deng J, Gao J, et al. A Hybrid User Similarity Model for Collaborative Filtering[J]. Information Sciences, 2017.

[14] Chen M C, Chen L S, Hsu F H, et al. HPRS: A profitability based recommender system[C]// IEEE International Conference on Industrial Engineering and Engineering Management. IEEE, 2008:219 - 223.

[15] Wu X, Cheng B, Chen J. Collaborative Filtering Service Recommendation Based on a Novel Similarity Computation Method[J]. IEEE Transactions on Services Computing, 2017, 10(3):352-365.

[16] Son J, Choi H, Dohn C Y. Skew-Tolerant Key Distribution for Load Balancing in MapReduce[J]. Ieice Trans.inf. & Syst, 2012, 95(2):677-680.

[17] Polatidis N, Georgiadis C K. A multi-level collaborative filtering method that improves recommendations[J]. Expert Systems with Applications, 2016, 48:100-110.

[18] Kaleli C. An entropy-based neighbor selection approach for collaborative filtering[J]. Knowledge-Based Systems, 2014, 56(C):273-280.

[19] Chen H, Li Z, Hu W. An improved collaborative recommendation algorithm based on optimized user similarity[J]. Journal of Supercomputing, 2016, 72(7):2565-2578.

[20] Gasmi I, Seridi-Bouchelaghem H, Hocine L, et al. Collaborative filtering recommendation based on dynamic changes of user interest[J]. Intelligent Decision Technologies, 2015, 9(3):271-281.

[21] Park Y, Park S, Jung W, et al. Reversed CF: A fast collaborative filtering algorithm using a k -nearest neighbor graph[J]. Expert Systems with Applications, 2015, 42(8):4022-4028.