Wu *et al. EURASIP Journal on Wireless Communications and Networking* (2019) 2019:80
https://doi.org/10.1186/s13638-019-1385-5

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Recommendation algorithm based on user score probability and project type

Chunxue Wu[1], Jing Wu[1], Chong Luo[1], Qunhui Wu[2], Cong Liu[1], Yan Wu[3] and Fan Yang[4*]

Abstract

The interaction and sharing of data based on network users make network information overexpanded, and "information overload" has become a difficult problem for everyone. The information filtering technology based on recommendation could dig out the needs and hobbies of users from the historical behavior, historical data, and social network and filter out useful resource for users in accordance with the needs and hobbies from the accumulation of information resource. Collaborative filtering is one of the core technologies in the recommendation system and is also the most widely used and most effective recommendation algorithm. In this paper, we study the accuracy and the data sparsity problems of recommendation algorithm. On the basis of the conventional algorithm, we combine the user score probability and take the commodity type into consideration when calculating similarity. The algorithm based on user score probability and project type (UPCF) is proposed, and the experimental data set from the recommendation system is used to validate and analyze data. The experimental results show that the UPCF algorithm alleviates the sparsity of data to a certain extent and has better performance than the conventional algorithms.

Keywords: Collaborative filtering, Score probability, Project type, Similarity calculation

1 Introduction

Recommendation algorithm is a very important tool to help users deal with information overload in the era of big data [1]. In the scoring matrix, the scoring behavior and scoring value of the user are the basis for the recommendation algorithm to recommend the product. In the era of information explosion, because the number of commodities is too large, the user can only score a few projects of their preferences. This results in the sparsity and incomplete of scoring data in the user-product scoring matrix, which makes it impossible to find similar neighbors of the target user. If there are no similar neighbors, the recommendation algorithm cannot recommend the product to the user, or the recommendation product to the user is inappropriate.

The primary cause of the sparse data in the scoring matrix of the recommendation system is that the user does not take the initiative to score the commodities. Therefore, the number of scores in the scoring matrix is not random, but depends on the user' subjective choice. Traditional recommendation algorithms think that users randomly choose and score the commodity. They also believe that users score high on commodities, which indicates that users like the product, and low scores on commodities, which indicates that users do not like the product. In [2], it is proved that the hypothesis of the traditional recommendation algorithm is inaccurate and does not accord with the reality of the massive information era, because the conventional algorithms ignore the performance of the user's subjective behavior.

In the big data era of information explosion, the number of commodities is very large. Users can only access to a small number of commodities, and then choose the type of interest preference from the small number of products to score. This results in the sparsity of scoring data in user commodity scoring matrix, which affects the accuracy of recommendation. Users choose products and score them, which is an invisible embodiment of user interest preference.

On the basis of the conventional algorithm, this paper integrates the subjective behavior of users to score the commodities and puts forward the algorithm of

* Correspondence: sallyf@zuel.edu.cn
[4]School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China
Full list of author information is available at the end of the article

integrating score preference and project type. Compared with the traditional algorithm that users randomly choose a product to score, the paper improves the recommendation algorithm (UPCF) with integrating score preference and project type, which makes full use of the subjective behavior of the user to choose and score commodity. The two-step predictive recommendation algorithm proposed in [3] and the probabilistic latent semantic recommendation algorithm based on an autonomous prediction all present similar points to this chapter. There are two kinds of difference between the improved algorithm and them in this paper. First, the method to calculate the probability of the user to score product is different. Second, the UPCF algorithm takes the product type into consideration in the similar calculation.

The main contents of this paper are as follows:

1. For the accuracy problem, the paper in the calculation of similarity integrates the project type. Combining the similarity calculated from the scoring matrix and the similarity obtained from the commodity type, the calculation of the similarity will be more accurate.
2. For the problem of data sparsity, the fundamental reason for data sparsity is that users do not take the initiative to score the project. This paper calculates user score probability by analyzing the user's historical scoring behavior and the type of the commodity. According to the score probability and commodity type, the similarity S2 of two users is calculated. The similarity S1 is calculated by a score matrix. The combination of the two similarities overcomes the problem that data sparsity cannot calculate user similarity.

2 The recommendation algorithm model of score preference and project type

2.1 User behavior information

The core idea of the recommendation algorithm is to obtain the information implied in the user's behavior, identify the user's behavior, use the collective wisdom [4] to match the user, and recommend the product to the user.

The traditional recommendation algorithm only pays attention to the value of the product scored by the user [5, 6], ignoring the user's implicit information in the behavior of scoring the commodities. The traditional algorithms indicate that the user randomly selects some of the commodities and scores them according to the degree of preference for the commodities. A high score shows that the user likes this commodity, and the low score shows that the user does not like the commodities. In the era of online shopping information explosion, a user's shopping behavior is based on their own needs and preferences [7–10]. The user will score the goods according to the quality of commodities, customer service attitude, logistics speed, and other factors. If the user is not interested in a commodity, he will not buy it. That is to say, giving a commodity a low score can only indicate that the user is not satisfied with this product. In the traditional algorithm, this dissatisfaction is spread to the same type of commodity, making the system think that the user's preference for similar products is reduced and affecting the system's recommendation for similar products. Therefore, it is a subjective behavior of the user to select a product and score it and this behavior is an invisible embodiment of the user's interest preference [11–13].

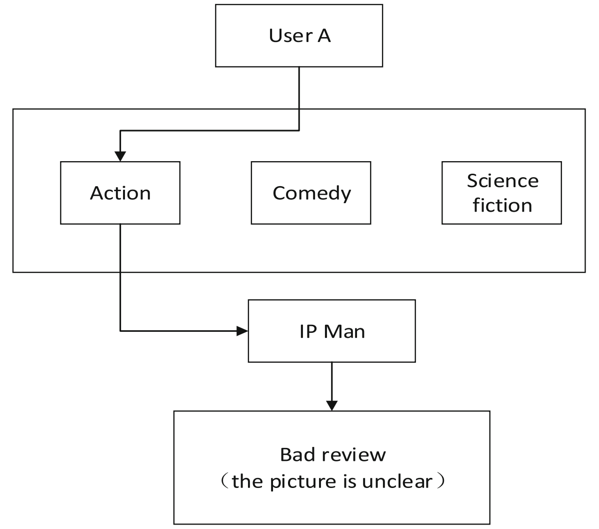

Fig. 1 Example of user behavior. User A loves to watch action movies, but does not love science fiction and comedy movies. User A finds a movie called "IP MAN" in the action movie category. After watching it, user A feels that the clarity of the picture is not good, so he gives the film a low score. If user A does not like the action movie, he will not see this category of movies. The traditional collaborative filter considers that as long as the user gives a low score that users do not like this type of film, but the fact is that if the user does not like a film, users will not pay attention to this type, let alone watch and score

About users' behavior, there are some different views between this paper and the traditional algorithms:

1. Different views on scoring behavior. The traditional algorithm considers that the user's scoring behavior is random. In this paper, it is considered that the scoring behavior is the implicit embodiment of the user's interest preference, and the user will only score the commodities that they are interested in.
2. Different reasons for the sparsity of scoring data [14–17]. The traditional algorithms think that the users' scoring behavior is random. This paper

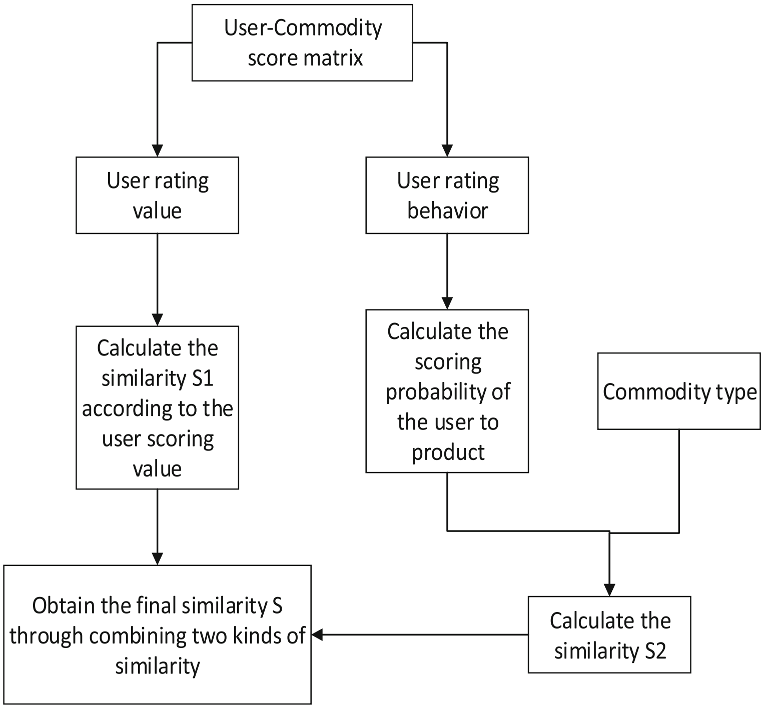
User-Commodity score matrix
User rating value
User rating behavior
Calculate the similarity S1 according to the user scoring value
Calculate the scoring probability of the user to product
Commodity type
Obtain the final similarity S through combining two kinds of similarity
Calculate the similarity S2

Fig. 2 Similarity calculation of an improved algorithm model. The behavior that a user gives a score to a commodity is called the user rating behavior. The score value that the user gives to the commodity is called the user rating value. The frame calculates the score probability *P* based on the user rating behavior and calculates the similarity S2 according to the score probability *P* and the project type. Then, we calculate the similarity S1 according to the user rating value. Finally, we combine the two similarity values S1 and S2 to obtain the ultimate similarity S

believes that users will only choose the products that they are interested in and score them, which means that the user's subjective choice results in the lack of data.

3. Different views on the level of scoring value. The traditional algorithms consider that the user likes the commodity if he gives it a high score, and the user does not like it to give the commodity a low score. This paper believes that as long as the user gives a score, regardless of the level of the score value, the user has a preference for this kind of commodity.

For example, as shown in Fig. 1, user A loves to watch action movies, but does not love science fiction and comedy. As the number of movies on the video site is very large, so user A will choose his favorite movie to watch. First of all, user A will choose the action movie on the video site, then scores it after watching the movie. Because they do not like to see the comedy and science fiction film, user A cannot evaluate such films. User A finds a movie called "IP MAN" [18, 19] in the action movie category. After watching it, user A feels that the clarity of the picture is not good, so he gives the film a low score. If user A does not like the action movie, he will not see this category of movies. The traditional collaborative filter [20, 21] considers that as long as the user gives a low score that users do not like this type of film, but the fact is that if the user does not like a film, users will not pay attention to this type, let alone watch and to score.

2.2 Recommendation algorithm model of user score preference and project type

This paper believes that the score value cannot indicate whether the user likes this kind of commodity or not, but only indicate that the user is not satisfied with the current product. Users pay attention to and want to consume the products that they are interested in. Users will score a high score for products that they are interested in and satisfied with, and low scores for products that they are interested in but not satisfied with. Therefore, regardless of whether the user gives a product a high or low score, the behavior of the user to score the product fully indicates that the user is interested in this type of product.

Based on the above viewpoints and the implicit expression of the user score behavior [22, 23], this paper designs a recommendation algorithm model based on user score preference and project type. By analyzing the user score behavior, our algorithm obtains the user's preference for the commodity. According to the preference of the user, we can predict the probability of the user to score the

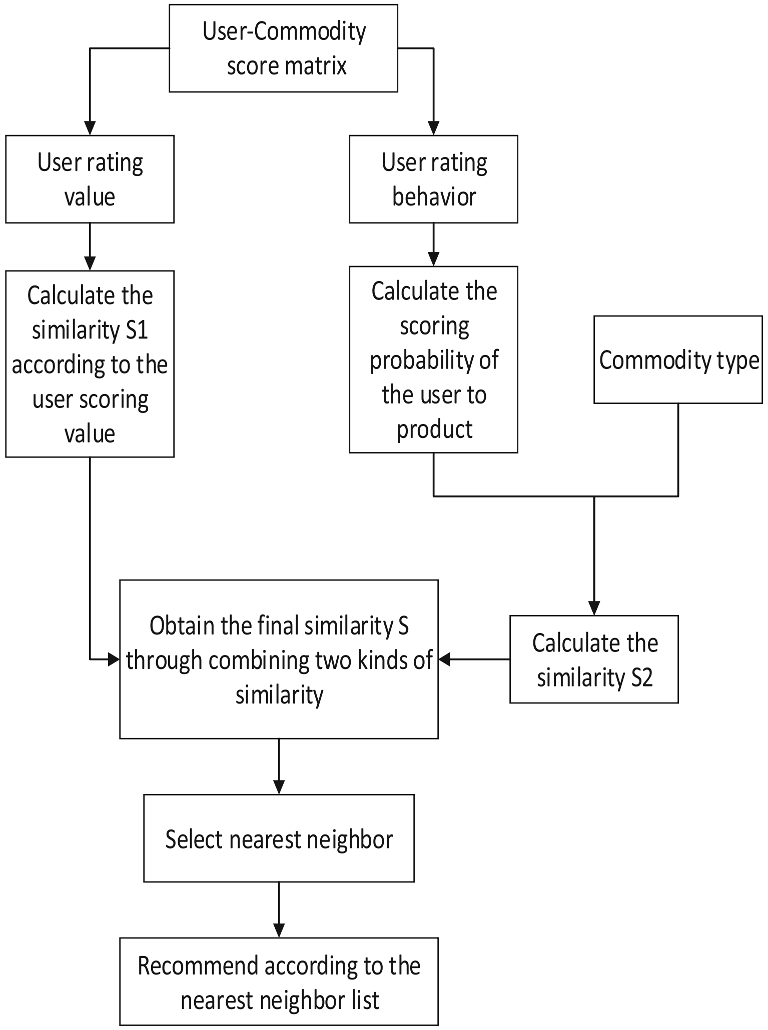
User-Commodity score matrix
User rating value
User rating behavior
Calculate the similarity S1 according to the user scoring value
Calculate the scoring probability of the user to product
Commodity type
Obtain the final similarity S through combining two kinds of similarity
Calculate the similarity S2
Select nearest neighbor
Recommend according to the nearest neighbor list

Fig. 3 Recommendation algorithm model of integrating score preference and project type. On the basis of the conventional algorithm based on neighborhood, the ideas of the user behavior in Fig. 1 and the improved similarity of Fig. 2 are integrated. The improved algorithm framework of this chapter is obtained, which is called the recommendation algorithm framework of scoring preference and project type. The core idea of the model of score preference and project types in Fig. 3: Firstly, we calculate the similarity S1 based on the matrix. Then, we calculate the possibility Pro of the user to score the product according to preference information implied in the user rating behavior. The second similarity S2 is calculated according to Pro and the type of the product. Due to the different weights of the two similarities, the final similarity of the user S is obtained by combining the two similarities S1 and S2

target commodities. The recommendation system combines the similarity calculated from the score value with the similarity calculated from the user score probability and the project type to make the similarity between the users more accurate. The similarity calculation of recommendation algorithm framework based on score preference and the project type is shown in Fig. 2.

As shown in Fig. 2, the behavior that a user gives a score to a commodity is called the user rating behavior. The score value that the user gives to the commodity is called the user rating value. The frame calculates the score probability *P* based on the user rating behavior and calculates the similarity S2 according to the score probability *P* and the project type. Then, we calculate the similarity S1 according to the user rating value. Finally, we combine the two similarity values S1 and S2 to obtain the ultimate similarity S [24, 25].

On the basis of the conventional algorithm based on neighborhood, the ideas of the user behavior in Fig. 1 and the improved similarity of Fig. 2 are integrated. The improved algorithm framework of this chapter is obtained, which is called the recommendation algorithm framework of scoring preference and project type. This framework makes full use of the user's preference information and calculates the user interest in a kind of commodity, that is, the probability of scoring. We will get the scoring probability and improved similarity by calculating [26]. The process of the model of the score preference and the project type is shown in Fig. 3.

The recommendation algorithm based on the user score preference and project type combines the probability of the user to score the commodity with the user's prediction value to the product. Some studies have shown that making full use of the user's behavior of scoring the product can effectively improve the recommendation accuracy of recommendation algorithm. The model takes full advantage of the user's scoring behavior, excavates the user's implied hobbies, and predicts the product type that the user may be interested in.

The core idea of the model based on the user score preference and project types is shown in Fig. 3: Firstly, we calculate the similarity S1 based on the matrix. Then, we calculate the possibility Pro of the user to score the product according to the preference information implied in the user rating behavior. The second similarity S2 is calculated according to Pro and the type of the product. Due to the different weights of the two similarities, the final similarity of the user S is obtained by combining the two similarities S1 and S2 [27, 28]. For all users, the similarity between any two forms an M×M similarity matrix (M is the number of users).

For example, in order to calculate the similarity of user A and user B, the algorithm reads the scoring information from the data set, gets the score matrix of 943×1682 (943 represents 943 users, and 1682 represents 1682 commodities), and calculates the Sim1(A,B) by score matrix and Pearson's formula. Then, a 943×18 score count matrix and a 943×18 score probability matrix are created (943 represents 943 users, and 18 represents of 18 types). Next, the algorithm traverses the score matrix, records the number of user score for each commodity into the 943×18 score matrix. The algorithm also traverses the score count matrix, calculates the probability of user score for each commodity, and records it into the scoring probability matrix. We obtain the second similarity S2(A,B) through probability matrix and Pearson's formula and get the final S(A,B) by combining S1(A, B) and S2(A, B). By calculating the similarity by the above way, we can obtain a 943×943 similarity matrix.

The UPCF algorithm takes full advantage of the user's behavior and the type of information of the product to score the product, which is the main difference between the UPCF algorithm and the traditional recommendation algorithm. The two-step prediction recommendation algorithm proposed in [3] and the probabilistic latent semantic recommendation algorithm based on autonomous prediction proposed in [13, 29] make full use of the user's behavior information to score the product. The difference between them is that the UPCF algorithm uses a different approach when calculating the score probability and considers the type of information of the project when calculating the similarity.

In order to verify the effectiveness of the framework, this paper combines IBCF with the framework in Fig. 2 to propose an algorithm of fusing score preferences and project types. The next section will detail the UPCF algorithm.

3 The algorithm of fusing score preference and project type

This section takes the user's subjective scoring behavior into consideration on the basis of the traditional recommendation algorithm based on neighborhood, proposing the algorithm of fusion score preference and item type. UPCF is short for collaborative filtering recommendation algorithm based on user score probability and project type.

3.1 Prediction of user score probability

The user's scoring preferences can also be used to calculate the user's score probability. The scoring value of all commodities in the score matrix can be regarded as an n-dimensional score vector, as follows:

$$P(U) = (I_1, I_2......I_n) \tag{1}$$

If the value is not 0, the user has scored the commodity; otherwise, there is no score on the commodities. Traverse the target user's n-dimensional score vector, count the type of commodities and the number of times each commodity is scored, and put the statistic results into the list. Each item in the list is an <i, n> binary relationship group, where i is the commodity type, and n is the number of times the commodities have been scored. We predict the users' interest in this type of product according to the number of users scoring a certain type in the list, that is, predicting the probability of users to score the commodity. If the target commodity type is j, $N(j)$ represents the scoring number of u on the j type and M is the total number of the user to score commodities. The user score probability is calculated as follows:

$$\Pr(u,j) = N(j)/M \tag{2}$$

The specific implementation of the score probability prediction is shown in the following pseudo code:

```
Prediction of scoring probability
Proba(int[][] grade)
    Create a new 943×18 matrix pro
    Traverse the grade
    If(grade[a][m]!=0)
        Get the type k of the movie m
        pro[a][k]+1
```

3.2 Improvements in similarity calculations

The cosine similar and Pearson et al. [30–32] are the most common way to calculate similarity in the conventional collaborative filtering algorithms. But regardless of the kind of calculation, the database of the calculation is the commodities' score commonly used by the users. This calculation ignores the commodity type [33]. This section incorporates the commodity type and the scoring probability of the user on the basis of the traditional similarity calculation method and adjusts the ratio of the two similarities. The improved similarity formula is as follows:

$$\mathrm{Sim}\left(U_i, U_j\right) = \beta S\left(U_i, U_j\right) + (1 – \beta) S_{\mathrm{sort}}\left(U_i, U_j\right) \quad (3)$$

where $\mathrm{Sim}(U_i, U_j)$ is the final similarity, $S(U_i, U_j)$ is the similarity calculated by using the user's score value, $S_{\mathrm{sort}}(U_i, U_j)$ is the similarity calculated by the commodity type and the scoring probability of product. The formula to calculate $S_{\mathrm{sort}}(U_i, U_j)$ is as follows:

$$S_{\mathrm{sort}}\left(U_i, U_j\right) = \frac{\sum\limits_{k \in L(U_i) \cap L\left(U_j\right)} \left(P_{U_i k} – \overline{P}_{U_i}\right)\left(P_{U_j k} – \overline{P}_{U\mathrm{j}}\right)}{\sqrt{\sum\limits_{k \in L(U_i) \cap L\left(U_j\right)} \left(P_{U_i k} – \overline{P}_{U_i}\right)^2 \sum\limits_{k \in L(U_i) \cap L\left(U_j\right)} \left(P_{U_j k} – \overline{P}_{U_j k}\right)^2}} \quad (4)$$

where $L(U_i)$ is a type collection of commodities that are scored by U_i. $L(U_j)$ is a type collection of commodities that are scored by U_j. $P_{U_i k}$ is the scoring probability of U_i for the k type, and $\overline{P}_{U_i}$ is the average of the scoring probability of U_i for all the types. The k type is one of the intersection types scored by U_i and U_j.

The selection of the nearest neighbor and the scoring prediction and scoring criteria have been described in detail in the previous section, and it is no longer described here.

3.3 The selection of the nearest neighbor

There are two conditions for the target user to choose the nearest neighbor [34–37]. Firstly, the selected neighbor is highly similar to the target user. Secondly, the selected neighbor has been scored on the target commodity.

When selecting the best neighbor, there is a threshold needed to set in order to prevent the existence of less similar individuals that affects the final results in collaborative filtering. Only the neighbor who has given the target product score and the similarity is greater than the threshold value that can become the target neighbor. The selection of neighbors is as follows:

$$\mathrm{KN}(U_m) = \{U_n / \mathrm{Sim}(U_m, U_n) > \sigma \mathrm{Score}(U_n, I) ! = 0, m \neq n\} \quad (5)$$

where $KN(U_m)$ is the neighbor list of the user U_m and β is the threshold, which can be set to the average value of the similarity of all the users who are similar to the user U_m. The specific implementation of selecting neighbor is as follows:

```
The selection of the nearest neighbor
FindNeighbor(int i, int k, int[][] grade, int[][] similar)
  where i is the user, k is the commodity, grade is the score matrix,
  similar is the similarity matrix.
  if(grade[j][k]!=0&&similar[i][j]!=0)
    List_N[j]=similar[i][j], List_N is the neighbor list
    Sort(List_N), sort the list of neighbors
    choose the N neighbors we need
```

3.4 Calculation of prediction score

In the calculation of the predicted score, the traditional collaborative filtering algorithm only focuses on the similarity [38, 39] between the neighbor and the target user and the neighbor's score for the prediction item. Each user has different scoring criteria. For example, some users give three points to show that they like that product, and some users need to give five points to express the same meaning. In order to solve this problem, this algorithm takes the average value of the user's score into account to resolve the difference between users. The user's rating for item v is

$$\mathrm{Score}(U_i, v) = \left(\overline{r}_{ui} + \frac{\sum\limits_{U_j \in KN(U_i)} \mathrm{Sim}(U_i, U_i)\left(r_{U_j v} – \overline{r}_{uj}\right)}{\sum\limits_{\mathrm{U_j \hat{I} KN(U_i)}} \mathrm{Sim}\left(\mathrm{U_i}, \mathrm{U_j}\right)} \right) f \quad (6)$$

where $f = \exp\{-1 + \alpha(\overline{r}_{ui} – \overline{r}_{uj})\}$, exp represents the exponential function based on e, $\overline{r}_{ui}$ is the average of U_i, $r_{u_j v}$ is score of U_j to v, α is the attenuation factor, and $\mathrm{Score}(U_i, v)$ is the prediction score of U_i to v.

3.5 Evaluation indicators

The evaluation indicators of the recommendation system can be summarized as accuracy and the other indicators out of accuracy [40, 41]. The accuracy of this paper is mainly referring to the accuracy index of the prediction score. This kind of indicator is to judge the accuracy by comparing the difference between the prediction score and the real score. The most commonly used is the MAE (mean absolute error), $|\mathrm{test}|$ is the test set, r_{uv} is a prediction of U to V, and r_{uv}^{test} is the real score of U to V in the test set.

MAE is calculated as follows:

$$\mathrm{MAE} = \frac{\sum\limits_{(U,V) \in \mathrm{test}} \left| r_{uv} – r_{uv}^{\mathrm{test}} \right|}{|\mathrm{test}|} \quad (7)$$

MAE is easy to understand and to calculate, but it also has some shortcomings that the MAE makes a contribution to the inaccurate prediction of low-score products. The RMSE (root mean square error) is also an evaluation indicator related to MAE; RMSE is calculated as follows:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{(u,v)\in \mathrm{test}} \left| r_{uv} - r_{uv}^{\mathrm{test}} \right|^2}{|\mathrm{test}|}} \tag{8}$$

In RMSE, each absolute error is squared, which makes the larger absolute error becomes larger.

4 Experimental design and analysis

4.1 Data sources

In order to verify the validity of the collaborative filtering recommendation algorithm of score preference and project type, the experiment was validated on the MovieLens set provided by GroupLens. The MovieLens data set is collected by the GroupLens Study Group of the University of Minnesota [42–44], which contains three different versions. This chapter selects the ml-100K data set for experimentation. The data set has 943 users and 1682 movies and 943×1682 score records. The score is 0 or a positive integer between 1 and 5, the score is 0 that the user did not score the product, the higher the score indicates that the higher the degree of the user's preferences for commodities. Ninety percent of the data set was randomly selected as the training set, and the rest was used for the experiment.

4.2 Experimental design

The acquisition of the experimental data samples is from the MovieLens data set [45–47] provided by GroupLens. We select the ml-100K version of MovieLens. The traditional collaborative filtering algorithm based on the nearest neighbor, GSCF algorithm, and UPCF algorithm is run on the data sets train1 and test1 and data sets train2 and test2, and then compare and analyze the difference between the MAE value and RMSE value of the three algorithms [48–50]. According to the analysis of the three algorithms recommendation results, the specific steps of the experimental design are as follows:

The first step, the division of the data set: the data set will be divided into two parts according to the proportion of 9:1 in accordance with the principle of completely random. One class is called the training set train, and the less is called the test set test. The ml-100K data set is divided into several times, and we obtain the training sets train1, train2, train3 and so on, as well as the test set corresponding to the training sets test1, test2, test3 and so on.

The second step, the prediction of scoring probability: the probability of the user's score for each type is calculated according to Section 3.1, and a score probability matrix of 943×18 is obtained. Nine hundred forty-three lines represent 943 users, and 18 columns represent 18 types of movies.

The third step, the similarity calculation: the similarity S1 is calculated by the score matrix, and the similarity S2 is calculated by the score probability. We will get a similarity matrix by combining the two similarities; the similarity matrix S is as follows:

$$S = \begin{bmatrix} S_{11} & S_{12} & \dots & S_{1m} \\ S_{21} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ S_{m1} & \dots & \dots & S_{mm} \end{bmatrix}$$

S_{m1} is the similarity between user m and user 1 in the matrix.

The fourth step is to choose the nearest neighbor according to the similarity.

The fifth step is to calculate the scoring error MAE, RMSE.

4.3 Experimental results and analysis

This section is compared with the traditional user-based propulsion algorithm UBCF, traditional item-based algorithm IBCF, a recommendation algorithm GSCF based on graph structure and project type. For the UPCF algorithm, we carry out experimental analysis. By comparing the influence of different neighbors on the MAE and RMSE of these four recommendation algorithms, the number of neighbors is the same, the difference between the MAE value and the RMSE value of the four recommendation algorithms is obtained. The number of neighbors which selected 10 to 80 variables was shown in the data sets train1 and test1 and data sets train2 and test2 of these four algorithms (UBCF, IBCF, GSCF, UPCF) in the MAE and RMSE performance. In the data sets train1 and test1, the three recommendation algorithm's MAE value of the comparison is shown in Table 1.

In Table 1, UBCF is based on the user's algorithm, IBCF is a conventional item-based algorithm, GSCF is an algorithm based on graph structure and project type, and UPCF is an algorithm based on user score preference and project type. When the number of neighbors is the same, the MAE value of the improved algorithm UPCF is the smallest, that is, the prediction error is the smallest and the recommendation performance is the best. When the number of neighbors is different, the MAE value of the four algorithms decreases first and then increases with the growth of the neighbors. It shows that the MAE is affected by the neighbors, in other words, the performance is affected by the neighbors. In order to make the comparison of the

Table 1 Comparison table for MAE value

	Number of neighbors							
	10	20	30	40	50	60	70	80
UBCF	0.792	0.777	0.773	0.772	0.772	0.772	0.772	0.773
IBCF	0.863	0.839	0.829	0.824	0.820	0.818	0.817	0.815
GSCF	0.779	0.767	0.763	0.763	0.763	0.764	0.764	0.765
UPCF	0.770	0.761	0.758	0.757	0.757	0.758	0.759	0.762

When the number of neighbors is the same, the MAE value of the improved algorithm UPCF is the smallest, that is, the prediction error is the smallest and the recommendation performance is the best. When the number of neighbors is different, the MAE value of the four algorithms decreases first and then increases with the growth of the neighbors. It shows that the MAE is affected by the neighbors, in other words, the performance is affected by the neighbors

four algorithms more obvious, this chapter draws the MAE values into line graphs, as shown in Fig. 4.

We take the number of neighbors as variables; with the increase of the number of neighbors, the MAE values of the four algorithms are reduced first and then flattened. This shows that the number of neighbors has a certain impact on the scoring error, when the number of neighbors is enough, this effect gradually weakened. In the case of the same number of neighbors, the MAE value of the algorithm UPCF is lower than the algorithm GSCF, which is lower than the UBCF and IBCF. This chapter shows that the error between the real score and the predicted score of the improved algorithm is the lowest, and the prediction of the user is more accurate. On the data sets train1 and test1, the RMSE values of the four recommendation algorithms are compared as shown in Table 2.

As can be seen from Table 2, the RMSE value of the algorithm UPCF is always smaller than the other three recommendation algorithms (under the same number of neighbors). As the number of neighbors increases, the RMSE value becomes smaller first and then bigger. When the number of neighbors is about 40, the value of RMSE tends to be the smallest, that is, the error of the prediction score is the smallest.

According to the data in Table 2, the RMSE values of the four contrast algorithms are plotted as histograms as shown in Fig. 5.

As can be seen from Fig. 5, with the increase of the number of nearest neighbors, the RMSE (root mean square error) value of the four algorithms is gradually reduced and then tends to be gentle. In the case of the same number of neighbors, the RMSE value of the UPCF algorithm has been lower than the other three kinds of recommendation algorithms, that is, the error between the real score and the prediction score of UPCF is the smallest, and the prediction is more accurate.

In order to exclude the impact of the data set on the results of the algorithm, the following experiments will be performed on second data sets train2 and test2 generated randomly.

In the data sets of train2 and test2, with the nearest neighbor as a variable, MAE and RMSE are the evaluation criteria to analyze and compare these four recommendation algorithms.

Figure 6 is the contrast effect diagram of the MAE value, with the increase in the number of nearest neighbors, the MAE value of the three algorithms decreases first and then increases. When the nearest neighbor number is about 40, the MAE value is the smallest, which shows that the collaborative filtering algorithm is affected by the nearest neighbor number.

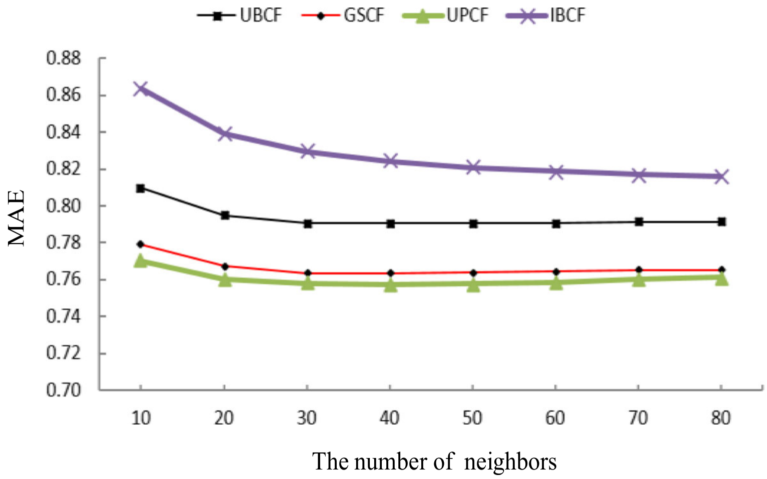

Fig. 4 Comparison of the MAE values of the four algorithms on the data set train1. When the number of neighbors is the same, the MAE value of the improved algorithm UPCF is the smallest, that is, the prediction error is the smallest and the recommendation performance is the best. When the number of neighbors is different, the MAE value of the four algorithms decreases first and then increases with the growth of the neighbors. It shows that the MAE is affected by the neighbors, in other words, the performance is affected by the neighbors

Table 2 Comparison of RMSE value

	Number of neighbors							
	10	20	30	40	50	60	70	80
UBCF	1.045	1.027	1.022	1.020	1.019	1.019	1.019	1.019
IBCF	1.079	1.048	1.035	1.028	1.024	1.022	1.020	1.019
GSCF	1.036	1.019	1.016	1.013	1.014	1.013	1.013	1.013
UPCF	1.026	1.016	1.013	1.011	1.012	1.012	1.012	1.013

The RMSE value of the algorithm UPCF is always smaller than the other three recommendation algorithms (under the same number of neighbors). As the number of neighbors increases, the RMSE value becomes smaller first and then bigger. When the number of neighbors is about 40, the value of RMSE tends to be the smallest, that is, the error of the prediction score is the smallest

When the nearest neighbor number is the same, the MAE of UPCF algorithm is the smallest, that is, the predicted score is close to the real value of the user and it provides the best recommendation result. It fully illustrates the importance of the user to the subjective behavior of commodity score.

Figure 7 is the contrast effect diagram of the RMSE value, with the increase in the number of nearest neighbors, the value of RMSE decreased first and then increased. When the nearest neighbor number is about 40, the value of RMSE is the smallest. In the case of the same number of near neighbors, the RMSE value of the UPCF algorithm is the smallest, that is, the error between the real score and the prediction score is the smallest, and the performance is the best.

4.4 Comparative analysis of algorithms

The recommendation algorithm GSCF based on graph structure and item type is based on the conventional algorithm, which makes full use of the indirect neighbor and commodity type when computing similarity.

However, the algorithm UPCF thinks that the user is the product of the user's subjective behavior; it is a kind of implicit user preferences. We make full use of the user score behavior that can find the root cause of data sparsity, as it causes no marks of the product for users. In order to better analyze and compare the two improved algorithms in this paper, we will compare the MAE value and RMSE value of the two algorithms.

Figure 8 is the algorithm UPCF and algorithm GSCF, which has the contrast line chart of the data set train1 on the MAE value (the average absolute error). It can be seen from the graph that the MAE value of the two algorithms decreases first and then increases, and finally tends to be gentle. With the near neighbor number as the variable, When the near neighbor number is about 40, the MAE values of the two algorithms are the smallest, that is, the error is the smallest. Under the condition of the same neighborhood, the numerical MAE of the UPCF algorithm is lower than the GSCF algorithm. The MAE value is smaller, which means that the scoring error of UPCF is lower than GSCF, that is, the recommendation effect of the algorithm UPCF is more accurate than the GSCF algorithm. Because it analyzes the root causes of data sparsity, it makes full use of the implicit information implied by user rating behavior, namely, the user's implicit preference.

Figure 9 is the RMSE (RMS error) contrast histogram between the algorithm GSCF and algorithm UPCF. The left is the algorithm GSCF, and the right is the algorithm UPCF. With the near neighbor number as the variable, the RMSE values of the two kinds of recommendation algorithms decrease first and then increase, and finally tend to be gentle. When the near neighbor number reaches about 40, the RMSE value of the two algorithms is

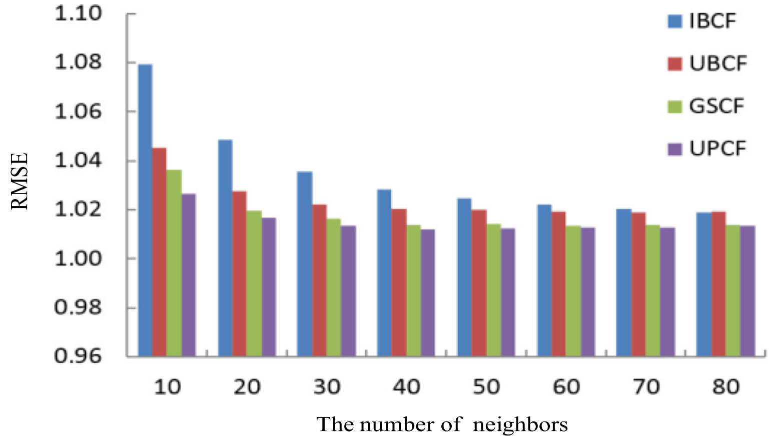

Fig. 5 The comparison of RMSE values between four algorithms in data set of train1. The RMSE value of the algorithm UPCF is always smaller than the other three recommendation algorithms (under the same number of neighbors). As the number of neighbors increases, the RMSE value becomes smaller first and then bigger. When the number of neighbors is about 40, the value of RMSE tends to be the smallest, that is, the error of the prediction score is the smallest

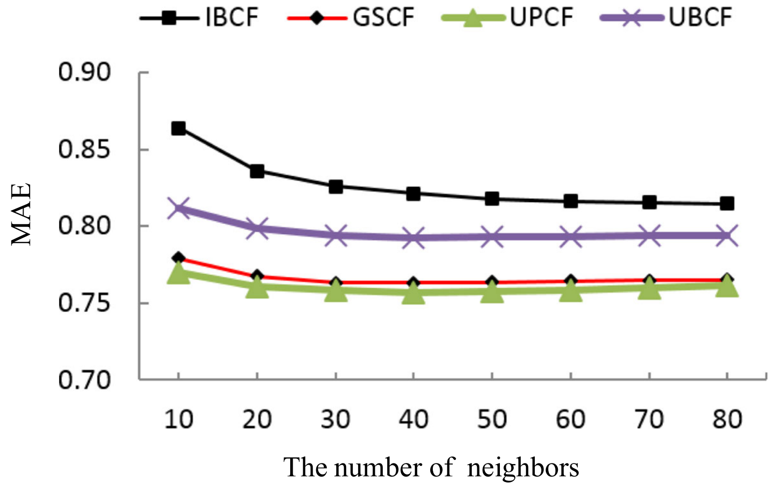

Fig. 6 The comparison of the MAE values between four algorithms in data set of train2. The MAE value of the three algorithms decreases first and then increases. When the nearest neighbor number is about 40, the MAE value is the smallest, which shows that the collaborative filtering algorithm is affected by the nearest neighbor number. When the nearest neighbor number is the same, the MAE of UPCF algorithm is the smallest, that is, the predicted score is close to the real value of the user and it provides the best recommendation result. It fully illustrates the importance of the user to the subjective behavior of the commodity score

the smallest, that is, the algorithm has the least error and the highest precision. In the case of the same number of near neighbors, the RMSE value of the algorithm UPCF is lower than that of the algorithm GSCF, that is, the error is smaller.

The experimental results show that the numerical values about MAE and RMSE of the UPCF algorithm are lower than the GSCF algorithm, namely, the error score is lower between the real and prediction score, so that the recommendation algorithm of UPCF is better than the GSCF algorithm.

5 Conclusions

The primary cause of the sparse data is that the user does not take the initiative to score the commodities. Due to the continuous updating of the commodity information, the user has no ability and energy to purchase and rate each commodity. The subjective behavior of the user to select a product and score it is an invisible embodiment of the user's interest preference. Users will only choose the products that they are interested in. If the user is satisfied with the commodities, he will give them high scores. On the basis

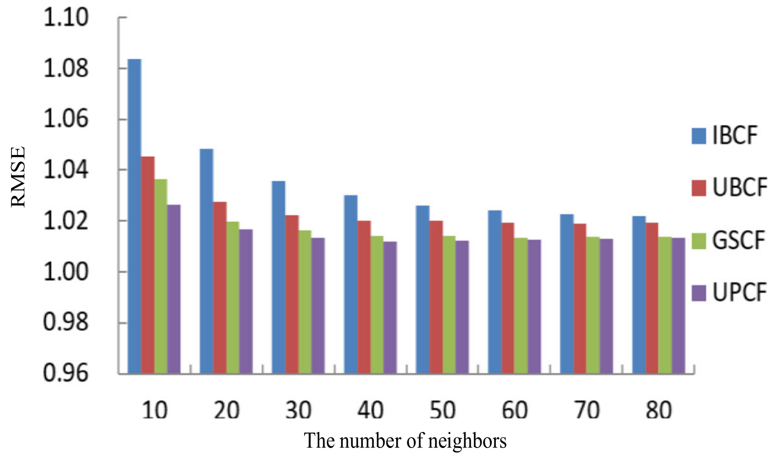

Fig. 7 The comparison of RMSE values between four algorithms in train2. With the increase in the number of nearest neighbors, the value of RMSE decreased first and then increased. When the nearest neighbor number is about 40, the value of RMSE is the smallest. In the case of the same number of near neighbors, the RMSE value of the UPCF algorithm is the smallest, that is, the error between the real score and the prediction score is the smallest, and the performance is the best

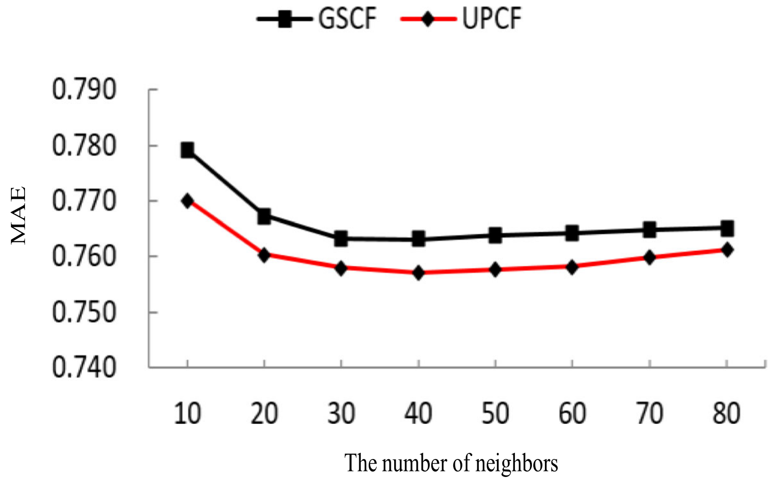

Fig. 8 Comparison of MAE values of two improved algorithms in a data set of train1. It can be seen from the graph that the MAE value of the two algorithms decreases first and then increases, and finally tends to be gentle. With the near neighbor number as the variable, when the near neighbor number is about 40, the MAE values of the two algorithms are the smallest, that is, the error is the smallest. Under the condition of the same neighborhood, the numerical MAE of the UPCF algorithm is lower than the GSCF algorithm. MAE value is smaller, which means that the scoring error of UPCF is lower than GSCF, that is, the recommendation effect of the algorithm UPCF is more accurate than the GSCF algorithm

of the conventional algorithm, this paper integrates the subjective behavior of users to score the commodities and makes full use of the implicit information in user behavior. The improved algorithm firstly calculates the scoring probability of the user to a product and then incorporates the commodity type and the scoring probability of the user to the product on the basis of the traditional similarity calculation method. Compared with the traditional algorithm based on a neighbor recommendation, GSCF algorithm, and UBCF algorithm, in the case of the same number of neighbors, MAE value and RMSE value UPCF algorithm are the lowest, which fully illustrates the usability of user preference information implied by the subjective behavior of user score.

This paper presents the UPCF algorithm. The project type is added to the traditional collaborative filtering recommendation algorithm to alleviate the cold start and

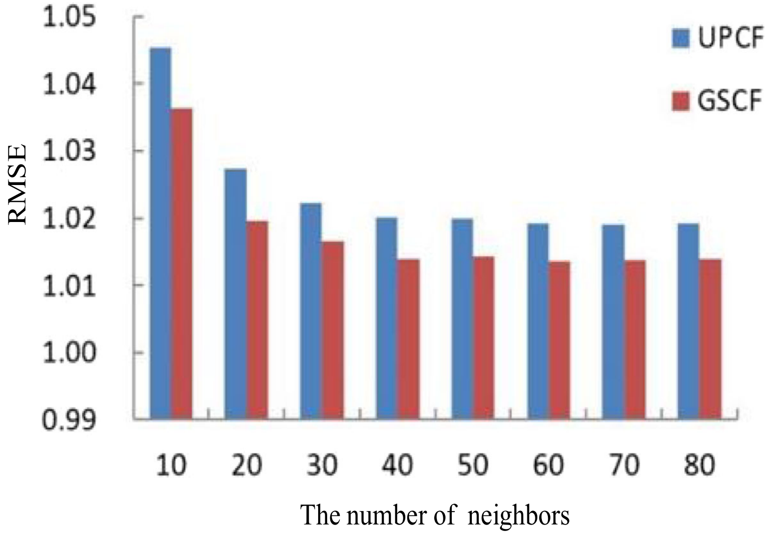

Fig. 9 Comparison of RMSE values of two improved algorithms in a data set of train1. With the near neighbor number as the variable, the RMSE values of the two kinds of recommendation algorithms decrease first and then increase, and finally tend to be gentle. When the near neighbor number reaches about 40, the RMSE value of the two algorithms is the smallest, that is, the algorithm has the least error and the highest precision. In the case of the same number of near neighbors, the RMSE value of the algorithm UPCF is lower than that of the algorithm GSCF, that is, the error is smaller

score sparsity problem. The primary reason of data sparsity problem is that users do not score the commodities, which is the users' subjective behavior. Based on the GSCF algorithm, this paper combines the user's willingness to score the commodity, and an algorithm based on user score probability and project type is proposed. The difference between UPCF and the traditional algorithm is analyzed, and the recommendation system dataset is used for experimental verification and data analysis. The experimental results show that the algorithm based on user score probability and project type alleviates the problem of data sparsity, which has a better effect than the conventional algorithm.

6 Future works

The current collaborative filtering recommendation technology research has been more mature, but there is still room for improvement in the recommendation accuracy and user experience. The improved algorithm proposed in this paper is only for the data sparsity situation. In order to solve other shortcomings of the traditional recommendation algorithm, the future work mainly around the followings:

(1) The use of social network to solve the cold start problem: The social information and display information (circle of friends, QQ space) of social network users are used to supplement and improve the recommendation algorithm user behavior information, so that we can get better predictions of user preference and enhance the recommendation performance of the recommendation algorithm.

(2) The use of time sequence to solve the problem of user interest drift: Because the interest of the user changes over time, the time is added to the recommendation algorithm to study the impact of this objective factor on the recommendation accuracy.

Abbreviations

GSCF: A collaborative filtering recommendation algorithm based on the graph structure; IBCF: A collaborative filtering recommendation algorithm based on item; MSE: Mean square error; RMSE: Root mean square error; UBCF: A collaborative filtering recommendation algorithm based on user; UPCF: A collaborative filtering recommendation algorithm based on user score probability and project type

Acknowledgements

The authors would like to appreciate all anonymous reviewers for their insightful comments and constructive suggestions to polish this paper in high quality. This research was supported by the National Key Research and Development Program of China (No. 2018YFC0810204), National Natural Science Foundation of China (No.61502220), Shanghai Science and Technology Innovation Action Plan Project (16111107502, 17511107203) and Shanghai key lab of modern optical system. In this paper, Fan Yang is the corresponding author.

About the authors

CHUNXUE WU received the Ph.D. degree in Control Theory and Control Engineering from of mining and technology, Beijing, China, in 2006.He is a Professor with the Computer Science and Engineering and software engineering Division, School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, China. His research interests include, wireless sensor networks, distributed and embedded systems, wireless and mobile systems, networked control systems.

JING WU is a graduate student of computer technology at School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, China. His research interests include Internet of Things, embeded system development, Deep Learning.

CHONG LUO (1991-) is a postgraduate student of computer technology, School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, China. His research interests include, networks communication, big data and machine learning.

QUNHUI WU is a Shanghai Hao long environmental technology Co., Ltd. system integration engineer. 2013 graduated from Xi'an Jiaotong University in computer science and technology. At present the main research direction for the computer system integration, computer control systems.

CONG LIU received the Ph.D. degree in computer application from the East China Normal University, Shanghai, China, in 2013. He is currently a Lecturer with the Department of Computer Science and Engineering, School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, China. His research interests include Evolutionary Computation, Machine Learning, and Image Processing.

YAN WU is currently a postdoctoral associate at the school of public and environmental affairs, Indiana University Bloomington. He obtained his PhD degree in Southern Illinois University Carbondale, with concentrations in environmental chemistry and ecotoxicology. His research involves elucidations of environmental fate of contaminants using chemical and computational techniques, as well as predictions of their associated effects on wildlife and public health. Data Processing and Analysis in Environmental Related Fields.

FAN YANG is an Associate Prof in School of Information, Zhongnan University of Economics and law. She received her PhD degree in School of Computer Science, Wuhan University, China, 2007, and M.S. degree in Dept. of Computer Engineering, Hubei University, China, 2004. Now Dr. Yang is doing some research on wireless communication security. Her research interest includes security analysis and improvements for Block Chain related technology.

Funding

This research was supported by the National Key Research and Development Program of China (No. 2018YFC0810204), National Natural Science Foundation of China (No.61502220), Shanghai Science and Technology Innovation Action Plan Project (16111107502, 17511107203) and Shanghai key lab of modern optical system.

Authors' contributions

The idea arose from the discussion between CW and JW. CLuo, QW, and YW performed the experiments. JW helped in finalizing the solution and amending the manuscript. CLiu and QW completed the writing and formatting of the paper. FY took charge of all the works for the submission of the paper. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Author details

[1]School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China. [2]Shanghai Haolong Environmental Science and Technology Co., Ltd, Shanghai 201110, China. [3]Public and Environmental Affairs, Indiana University Bloomington, Bloomington, IN 47405, USA. [4]School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China.

Received: 29 June 2018 Accepted: 27 February 2019
Published online: 29 March 2019

References

1. L. Jianguo, Z. Tao, W. Binghong, Research progress of personalized recommendation system. Adv. Nat. Sci. **19**(1), 1–15 (2009)
2. D. Ailin, Z. Yangyong, S. Bole, Collaborative filtering recommendation algorithm based on item score prediction. Softw. J. (9), 1621–1628 (2003)
3. Z. Xiangyu, *Study on top-N collaborative filtering recommendation technology* (Beijing Institute of Technology, Beijing, 2014)
4. J. Xiao, M. Luo, J.M. Chen, J.J. Li, in *Advanced Intelligent Computing Theories and Applications. ICIC 2015. Lecture Notes in Computer Science*, ed. by D. S. Huang, K. Han. An Item Based Collaborative Filtering System Combined with Genetic Algorithms Using Rating Behavior, vol 9227 (Springer, Cham, 2015), pp. 453–460
5. L. Shijia, *Research and application of collaborative filtering algorithm based on coupling similarity* (Zhejiang University, Hangzhou, 2016)
6. F. Bo, C. Jiujun, Among multiple users similarity collaborative filtering algorithm. Comput. Sci. **39**(1), 23–26 (2012)
7. W. Jing, Y. Jian, An optimized item-based collaborative filtering recommendation algorithm. Mini Comput. Syst. **31**(12), 2337–2342 (2010)
8. C. Yanping, W. Sai, Hybrid collaborative filtering algorithm based on user item. Comput. Technol. Dev. **24**(12), 88–91 (2014)
9. L. Qihua, Z. Liyi, Research progress on the diversity of personalized recommendation system. Libr. Inf. Work **57**(20), 127–135 (2013)
10. Z. Tao, Ten challenges of personalized recommendation technology. Programmers **6**, 107–111 (2012)
11. J. Xu, *Personalized recommendation algorithm based on comments and ratings* (Zhejiang University, Hangzhou, 2013)
12. Y. ting, Personalized recommendation based on collaborative filtering (Beijing Institute of Technology, Beijing, 2015)
13. C.X. Jia, R.R. Liu, Improve the algorithmic performance of collaborative filtering by using the interevent time distribution of human behaviors. Physica A **436**, 236–245 (2015)
14. M. Elahi, F. Ricci, N. Rubens, A survey of active learning in collaborative filtering recommender systems. Comput. Sci. Rev. **20**(C), 29–50 (2016)
15. W. Kong, *Research on the key issues of collaborative filtering recommendation system* (Huazhong Normal University, Wuhan, 2013)
16. L. Qiang, *Research on key algorithms in collaborative filtering recommendation system* (Zhejiang University, Hangzhou, 2013)
17. H. Liu, Z. Hu, A. Mian, et al., A new user similarity model to improve the accuracy of collaborative filtering. Knowl.-Based Syst. **56**(3), 156–166 (2014)
18. J. Lee, M. Sun, G. Lebanon, *A comparative study of collaborative filtering algorithms. arXiv preprint arXiv:1205*, vol 3193 (2012)
19. Y. Zeng, C.J. Sreenan, N. Xiong, L.T. Yang, J.H. Park, Connectivity and coverage maintenance in wireless sensor networks. J. Supercomput. **52**(1), 23–46 (2010)
20. C.L. Liao, S.J. Lee, A clustering based approach to improving the efficiency of collaborative filtering recommendation [J]. Electron. Commer. Res. Appl. **18**, 1–9 (2016)
21. X. Peiyong, *Research on collaborative filtering algorithm in personalized recommendation technology [D]* (Ocean University of China, Qingdao, 2011)
22. H. Chuangguang, Y. Jian, W. Jing, et al., Uncertain of the nearest neighbor collaborative filtering recommendation algorithm. J. Comput. **33**(8), 1369–1377 (2010)
23. N. Xiong, A.V. Vasilakosb, L.T. Yang, C. Wang, R. Kannane, C. Chang, Y. Pan, A novel self-tuning feedback controller for active queue management supporting TCP flows. Inf. Sci. **180**(11), 2249–2263 (2010)
24. G. Shenhua, A collaborative filtering algorithm based on singular value decomposition and temporal weight. Comput. Appl. Softw. **27**(6), 256–259 (2010)
25. X. Yang, J. Yu, T. Ergen, et al., The collaborative filtering model combined singularity and diffusion process. Softw. J. (8), 1868–1884 (2013)
26. Z. Qinqin, L. Kai, W. Bin, SPCF: a memory based collaborative filtering recommendation algorithm [J]. J. Comput. Sci. **36**(3), 671–676 (2013)
27. Y. Ar, E. Bostanci, A genetic algorithm solution to the collaborative filtering problem. Expert Syst. Appl. **61**, 122–128 (2016)
28. C. Zhou, S. Huang, N. Xiong, S.H. Yang, H. Li, Y. Qin, X. Li, Design and analysis of multimodel-based anomaly intrusion detection systems in industrial process automation. IEEE Trans. Syst. Man Cybern. Syst. **45**(10), 1345–1360 (2017)
29. R. He, N. Xiong, L.T. Yang, J.H. Park, Using multi-modal semantic association rules to fuse keywords and visual features automatically for web image retrieval. Inf. Fusion **12**(3), 223–230 (2011)
30. Z. Wang, T. Li, N. Xiong, Y. Pan, A novel dynamic network data replication scheme based on historical access record and proactive deletion. J. Supercomput. **62**(1), 227–250 (2012)
31. N. Xiong, J.W.A.V. Vasilakos, Y.R. Yang, A. Rindos, Y. Zhou, in *A self-tuning failure detection scheme for cloud computing service*. W.Z. Song, in 2012 IEEE 26th Parallel & Distributed Processing Symposium (IPDPS) (2012)
32. J. Yin, W. Lo, S. Deng, Y. Li, Z. Wu, N. Xiong, Colbar: a collaborative location-based regularization framework for QoS prediction. Inf. Sci. **265**, 68–84 (2014)
33. C. Wu, J. Yuan, B. Shi, Stability of initialization response of fractional oscillators. J. Vibroengineering **18**(6), 4148–4154 (2016)
34. G. Li, Z. Zhang, L. Wang, et al., One-class collaborative filtering based on rating prediction and ranking prediction. Knowl.-Based Syst. **124**, 46–54 (2017)
35. B. Lin, W. Guo, N. Xiong, G. Chen, A.V. Vasilakos, H. Zhang, A pretreatment workflow scheduling approach for big data applications in multi-cloud environments. IEEE Trans. Netw. Serv. Manag. **13**(3), 581–594 (2016)
36. Z. Wan, N. Xiong, N. Ghani, A.V. Vasilakos, L. Zhou, Adaptive unequal protection for wireless video transmission over IEEE 802.11 e networks. Multimed. Tools Appl. **72**(1), 541–571 (2014)
37. Y. Sang, H. Shen, Y. Tan, N. Xiong, in *Efficient protocols for privacy preserving matching against distributed datasets*. International Conference on Information and Communications Security (2006), pp. 210–227
38. R. Han, Y. Gao, C. Wu, An effective multi-objective optimization algorithm for spectrum allocations in the cognitive-radio-based internet of things. IEEE Access **6**, 12858–12867 (2018)
39. H. Zheng, W. Guo, N. Xiong, A kernel-based compressive sensing approach for mobile data gathering in wireless sensor network systems. IEEE Trans. Syst. Man Cybern. Syst **8**(99), 1–13 (2017) https://doi.org/10.1109/TSMC.2017.2734886
40. Z. Yuxiao, L. Linyuan, Review of evaluation index of recommendation system. J. Univ. Electron. Sci. Technol. China **41**(2), 163–175 (2012)
41. L. Jianguo, Z. Tao, G. Qiang, et al., Review of evaluation methods for personalized recommendation systems. Soc. Syst. Complex. Sci. **6**(3), 1–10 (2009)
42. H. Shanshan, *Study on the key issues of collaborative filtering recommendation algorithm* (Shandong University, Ji'nan, 2016)
43. L. Qingwen, *Study on the recommendation algorithm based on collaborative filtering* (University of Science & Technology China, Hefei, 2013)
44. Q. Liu, E. Chen, H. Xiong, et al., Enhancing collaborative filtering by user interest expansion via personalized ranking. IEEE Trans. Syst. Man & Cybern. B Cybern. A Publication of the IEEE Systems Man & Cybernetics Society **42**(1), 218–233 (2012)
45. Y. Zhou, D. Zhang, N. Xiong, Post-cloud computing paradigms: a survey and comparison. Tsinghua Sci. Technol. **22**(6), 714–732 (2017)
46. X. Liu, S. Zhao, A. Liu, N. Xiong, A.V. Vasilakos, Knowledge-aware proactive nodes selection approach for energy management in internet of things. Future Generation Computer Systems. (2017) https://doi.org/10.1016/j.future.2017.07.022
47. N. Xiong, A.V. Vasilakos, L.T. Yang, L. Song, Y. Pan, R. Kannan, Y. Li, Comparative analysis of quality of service and memory usage for adaptive failure detectors in healthcare systems. IEEE J. Sel. Areas Commun. **27**(4), 495–509 (2009)
48. C. Lin, N. Xiong, J.H. Park, T. Kim, Dynamic power management in new architecture of wireless sensor networks. Int. J. Commun. Syst. **22**(6), 671–693 (2010)
49. J. Li, N. Xiong, J.H. Park, C. Liu, M.A. Shihua, S.E. Cho, Intelligent model design of cluster supply chain with horizontal cooperation. J. Intell. Manuf. **23**(4), 917–931 (2012)
50. W. Fang, Y. Li, H. Zhang, N. Xiong, J. Lai, A.V. Vasilakos, On the throughput-energy tradeoff for data transmission between cloud and mobile devices. Inf. Sci. **283**, 79–93 (2014)