Research on Collaborative Filtering of Food Information Security in E-Commerce Platform

Sheng Luo¹,²,*

¹School of Information and Statistics, Guangxi University of Finance and Economics, Nanning530003, China
²Guangxi Key Laboratory of Cross-border E-commerce Intelligent Information Processing, Guangxi University of Finance and Economics, Nanning530003, China

*315412323@qq.com

Abstract. Recommender systems are one of the most important technologies in the electronic commerce system. In a collaborative filtering recommendation algorithm, similarity calculation is the key to determining the efficiency of the recommendation algorithm. This paper analyzes the shortcomings of traditional similarity measurement methods in recommender systems and proposes a scoring-matrix-filling algorithm. Based on information categories and user interest similarity, the algorithm can reduce the negative influence of data sparsity on the recommendation result to some extent. The research results have certain practical and guiding significance.

Keywords: collaborative filtering, similarity, recommendation algorithm, degree of interest

1. Introduction

The rapid development of Internet information technology has greatly changed the way people access information. However, with the increasing amount of information on the Internet, the problem of information overload is becoming progressively more serious. As technical means for effectively solving the problem of Internet information overload, information recommender systems are attracting increasing attention in the industry. An information recommender system can learn the user's behavior to understand and master the user's preferences so that it can more effectively recommend to the user content in which he or she may be interested. At present, several major Internet industries, including e-commerce and social networks, have used the recommender system to varying degrees.
Among the existing recommendation algorithms, collaborative filtering recommendation algorithms are relatively mature and have been widely used in e-commerce [1]. At present, most collaborative filtering recommendation algorithms calculate the predicted score of a user without rating items and use this as the main basis for recommending information to the user. In a collaborative filtering recommendation algorithm, determining the nearest neighbor of the user is the key step. Therefore, calculating the similarity between items using a scoring matrix is the core issue of collaborative filtering algorithms. Traditional methods for computing similarity include the cosine similarity, correlation similarity, and modified cosine similarity algorithms [2]. According to the collaborative filtering theory, the more user score data are available, the higher the recommendation quality of a collaborative filtering recommendation algorithm. However, in an actual recommender system, the user's score data is often very sparse, and it is difficult to accurately measure the similarity between users by using a traditional similarity method, which leads to low recommendation accuracy [3]. As the electronic commerce system continues to expand in scale, especially in recent years, the number of users and commodities has increased dramatically. However, user-rated items usually account for only 1% of the total items, thereby resulting in extremely sparse data and low accuracy of traditional measurement methods. Therefore, Breese et al. [4] proposed a method for alleviating data sparseness by default voting and inverse user frequency. Adomavicius and Tuzhilin [5] proposed the combination of a collaborative filtering recommendation algorithm with a content-based recommendation algorithm, which effectively alleviated the impacts of new-user problems and new-project problems. On the basis of Sarwar et al., Huang et al. [6] further integrated user-based and item-based collaborative filtering recommendation algorithms, which solved the problem of data sparseness to some extent and improved the recommendation quality of the algorithm. However, in general, the existing research methods still have problems of low efficiency and low accuracy.

To alleviate the problem of recommendation accuracy reduction due to data sparsity, we propose a new recommendation algorithm based on item classification information. First, we use the domain nearest-neighbor method, which is based on classification information, to estimate the items that have not yet been scored. We no longer use a single similarity measurement method; instead, the user interest similarity is combined with traditional similarity to calculate user similarity. In addition, we improved the traditional user interest similarity measurement method and proposed a collaborative filtering recommendation algorithm that is based on user degree of interest. This method takes into account the differences in the number of items in each category and can more accurately describe the user's true preferences for different item categories, thereby making the user similarity measurement more accurate, which results in a higher-quality recommendation.

2. Traditional collaborative filtering recommendation algorithm

2.1 Calculation method of user similarity

To calculate the predicted scores of non-rated items, we need to calculate the score similarities between the target user and other users according to the users’ historical score records. Similarity reflects the degree of similarity between the users’ preferences for each item. There are usually three main types of similarity: correlation similarity, cosine similarity, and modified cosine similarity.
2.1.1. Correlation Similarity
Correlation similarity is also called Pearson correlation. Suppose that the set of items that have been rated by users \( m \) and \( n \) is \( I_{m,n} \); \( A_{m,j} \) and \( A_{n,j} \) represent the ratings of the item \( j \) by users \( m \) and \( n \), respectively, and \( \bar{A}_m \) and \( \bar{A}_n \) represent the average scores of users \( m \) and \( n \) on all items, respectively. Then, the correlation similarity between users \( m \) and \( n \) is:

\[
Sim(m, n) = \frac{\sum_{j \in I_{m,n}} (A_{m,j} - \bar{A}_m)(A_{n,j} - \bar{A}_n)}{\sqrt{\sum_{j \in I_{m,n}} (A_{m,j} - \bar{A}_m)^2} \sqrt{\sum_{j \in I_{m,n}} (A_{n,j} - \bar{A}_n)^2}}
\]

(1)

2.1.2. Cosine Similarity
Cosine similarity is also called vector similarity. Each user's score record is represented as a vector in the \( n \)-dimensional item space. If the user fails to score an item, the user's rating for the item is set to zero. The similarity between two users is defined as the cosine of the angle between the scoring vectors of the two users. Suppose that the score vectors of users \( m \) and \( n \) are \( M \) and \( N \), respectively. Let \( I_m \) and \( I_n \) represent the sets of items that have been scored by users \( m \) and \( n \). The cosine similarity between the users \( m \) and \( n \) is:

\[
Sim(m, n) = \cos(M, N) = \frac{\sum_{j \in I_{m,n}} A_{m,j}A_{n,j}}{\sqrt{\sum_{j \in I_{m,n}} A_{m,j}^2} \sqrt{\sum_{j \in I_{m,n}} A_{n,j}^2}}
\]

(2)

2.1.3. Modified Cosine Similarity
The cosine similarity measure ignores the rating scale problem of different users. However, Modified cosine similarity can make up for this problem. That is modified cosine similarity minus the average score of the user on the item. Suppose that the set of items that have been rated by both users \( m \) and \( n \) is \( A_{m,n} \). Let \( \bar{A}_m \) and \( \bar{A}_n \) represent the sets of items that have been scored by users \( m \) and \( n \), respectively. The modified cosine similarity between users \( m \) and \( n \) is:
2.2. Score Prediction

A predicted score reflects the extent to which the current user may be interested in an item. It is calculated based on the ratings of the item by other users who are similar to the current user. Score prediction is the main method that most recommender systems use to generate recommendations. When calculating the predicted score, $K$ users are often used as the neighboring nodes of the current user to calculate. Suppose the set $R_m$ represents the nearest-neighbor set of users $m$. To calculate the predicted score of the item $j$ by user $m$, the following formula is used:

$$\sum_{m \in R_m} \frac{\sum_{m \in R_m} \text{Sim}(m, m') (A_{m,j} - \overline{A_m})}{\sum_{m \in R_m} \text{Sim}(m, m')}$$

where $A_{m,j}$ represents the non-empty score of user $m$ on the item $j$, $\overline{A_m}$ represents the average score in the set of common scored items of $m$ and $m$, and $\overline{A_m}$ represents the average score of user $m$ on all projects.

3. Collaborative filtering algorithm based on user interest degree

3.1. Scoring prediction based on category

Among the abovementioned similarity calculation methods, the cosine similarity algorithm is a method for calculating the similarity between users through distance measurement. The correlation similarity algorithm calculates the similarity among users by statistical means. Traditional similarity calculation methods have some disadvantages. If user score data are extremely sparse, it is difficult to apply the traditional similarity measurement method. In the cosine similarity algorithm, the scores of items that are not scored by users are set to 0, and the score that is assigned by user $i$ to evaluation item $j$ is $a_{i,j}$. Then, when constructing the user rating data matrix $A(m,n)$, the score that is assigned by user $i$ to project $j$ is $A_{i,j}$:

$$A_{i,j} = \begin{cases} 0, & \text{if user i did not rate item j} \\ a_{i,j}, & \text{if user i rated item j} \end{cases}$$
Although this method can improve the computational efficiency, the credibility of the scoring matrix is not high in the case of sparse data. The users score the items that have not yet been scored, and the scores are not exactly the same. In addition, the modified cosine similarity and the correlation similarity take into account the common scored items among users. If there are a few common scored items between two users, even if the user scores on the small set are very similar, there is no guarantee of high similarity between them. In summary, the traditional similarity measurement methods (correlation similarity, cosine similarity, and modified cosine similarity) cannot effectively measure the similarity between users in the case of extremely sparse data and lead to degraded recommendation quality. To solve the sparsity problem of user rating data, the traditional method is to set the user’s non-rated item to a fixed default value (this value is generally the intermediate value of the scoring domain; for example, for the 5-point scoring system, it is set to 3), or to the average evaluation value of the user. However, when each user evaluates non-rating items, the results will not be exactly the same; therefore, setting a fixed default value cannot fundamentally solve the problem of scoring sparse data. To solve the problem of extreme sparsity of user rating data, this paper uses the set of scored items to calculate the similarity between users. In addition, based on the category of information, we use the "domain nearest neighbor" method to fill the union of score item, so that the nearest-neighbor selection is more accurate.

In this paper, the nearest-neighbor set of the user is obtained by the "domain nearest neighbor" method. Assume that $U_{mn}$ is the union of items that were scored by users $m$ and $n$. To predict the score of item $j$ (which was not rated) in $U_{mn}$. In the scoring matrix $A(e, g)$, the average number of items in the category $C_u$ of item $j$ is selected, thereby constituting the domain scoring matrix $A_j$. Then, user $m$’s nearest neighbor set is computed based on $A_j$, and the predicted score of $j$ is obtained.

In practical applications, all items are divided into several categories. Take the MovieLens dataset as an example, which contains 100,000 score data of 947 users for 1,679 movies, and divides the movies into 19 categories. Denote target user $m$’s score set as $I_m$ and target user $n$’s score set as $I_n$. The categories of the items in $I_m$ and $I_n$ belong to $C_m$ and $C_n$, respectively. Let $C_j = C_m \cap C_n$. There are two cases, as shown in Tab. 1.

|       | Action | Comedy | Horror |
|-------|--------|--------|--------|
| $I_1$ | 2      |        |        |
| $I_2$ | 4      |        |        |
| $I_3$ |        | 1      |        |
| $I_4$ |        | 5      |        |
| $I_5$ |        |        | 3      |
| $I_6$ |        |        |        |
| $I_7$ |        |        |        |

**Tab.1** User-rated item distribution
(1) There exists \( C_j \in (C_m \cup C_n) \) such that \( C_j \notin C_i \); see Fig. 1. That is, in \( (C_m \cup C_n) \), there is at least one user who fails to score all the items in a category. For example, in Tab.1, items \( I_3 \) and \( I_4 \) of user \( m \) in the “Comedy” class are not rated, whereas user \( n \) has rated the items in this class. Then, we take the average score of the user in the comedy class as the score of user \( m \) for items \( I_3 \) and \( I_4 \). That is the scores of the non-scored items in \( C_j \) are set as the average score of the class.

![Fig. 1 User scoring set](image)

(2) There exist non-scored items of class \( C_j \). For example, in Table 1, users \( m \) and \( n \) have scored items from both the “Horror” class and the “Action” class. Then, we use the nearest-neighbor method to compute the predicted scores of user \( n \) for \( I_1 \) and \( I_5 \); the formula is as follows:

\[
F_j = \overline{A}_m + \frac{\sum_{m \in R_m} Sim(m, m')(A_{m', j} - \overline{A}_m)}{\sum_{m \in R_m} Sim(m, m')} \tag{6}
\]

where \( R_m \) is the nearest-neighbor set of user \( m \) in category \( M \), \( Sim(m, m') \) represents the similarity between user \( m \) and user \( m' \), \( A_{m', j} \) represents the score of user \( m' \) on item \( j \), and \( \overline{A}_m \) and \( \overline{A}_{m'} \), respectively, represent the average scores of user \( m \) and user \( m' \) in category \( M \).

By the above method, we can estimate the non-scored items’ scores in the scored-item union of two users. Then, we can use the similarity measurement method to calculate the similarity between the two users.
3.2. Similarity measurement method based on interestingness degree

The traditional collaborative filtering algorithm only considers the user's single scoring similarity. The algorithm of collaborative filtering based on the user's interests considers the categories of the item. The scoring results reflect the user's preference similarity for each item category. This improves the similarity measure between users, which improves the recommendation quality of the algorithm to some extent. At present, the collaborative filtering recommendation algorithm that is based on multi-similarity among users is becoming increasingly popular in practical applications [13].

In the network environment, the user's preference for the project category provides an important basis for information recommendation. If a user reviews more of the item, it shows that the user is more interested in the item. When two users have the same interests, we can assume they have high similarity. Therefore, we define the "User-Item" category scoring matrix $N$:

$$
N = \begin{bmatrix}
N_{11} & \cdots & N_{1k} \\
\vdots & \ddots & \vdots \\
N_{s1} & \cdots & N_{sk}
\end{bmatrix}
$$

(7)

where $s$ represents the number of users, and $N_{sk}$ represents the number of users that have evaluated the items in category $k$. The degree of interest of user $x$ in items of class $a$ can be expressed as:

$$
I_{xa} = \frac{N_{xa}}{N_x}
$$

(8)

where $N_{xa}$ represents the total number of users $x$ who have evaluated an item of $a$, and $N_x$ represents the total number of evaluations of user $x$. If a user reviews more of the item, it shows that the user is more interested in the item. Then, the interest similarity of the two users can be calculated by cosine similarity:

$$
Sim(m,n) = \frac{\sum_{a=1}^{k} A_{m,a}A_{n,a}}{\sqrt{\sum_{a=1}^{k} A_{m,a}^2} \sqrt{\sum_{a=1}^{k} A_{n,a}^2}}
$$

(9)
where $k$ is the number of categories of the project, and $A_{m,a}$ represents the interest degree of user $m$ in items of class $a$. Therefore, the overall similarity between users can be obtained by combining the user score similarity and the user interest similarity:

$$Sim(m,n) = (1 - \omega)sim_1(m,n) + \omega sim_2(m,n)$$

(10)

where $sim_1(m,n)$ represents the item scoring similarity between user $m$ and user $n$, $sim_2(m,n)$ represents the interest similarity degree between user $m$ and user $n$, and $0 < \omega < 1$.

However, the number of items in each category is not taken into account in formula (8), so there may be a big difference in the number of items in each category. Take GROUP1 in Tab.3 as an example. Tab.2 lists the number of different types of movies in GROUP1.

| Unknown | Action | Adventure | Animation | Children’s | Comedy | Crime | Documentary |
|---------|--------|-----------|-----------|------------|--------|--------|-------------|
| 4       | 267    | 142       | 55        | 145        | 548    | 132    | 53          |
| Drama   | Fantasy| Film-Noir | Horror    | Musical    | Mystery| …      |             |
| 874     | 26     | 33        | 94        | 68         | 73     |        |             |

Tab.2 Number of movies in each category

In Tab.2, there are big differences in the number of movies in different categories. Of the 19 categories, 14 have more than 50 movies. There are 10 categories with less than 100 movies; the highest value was 725 and the lowest value was 2.

In addition, in practical applications, the numbers of items in different categories have an impact on the measurement results. Taking the MovieLens dataset as an example, suppose that user $x$ has scored 20 films from each of the action and horror classes. Using the formula (8), it is concluded that user $x$ has the same preference for action movies and horror movies. However, the actual situation is that the number of action movies is far greater than the number of horror movies, and horror movies have smaller audiences than action movies. Thus, since user $x$ has scored the same number of action and horror movies, user $X$ may be a horror film enthusiast. Therefore, we improve formula (9) so that it can more accurately calculate the user's interest in practical applications. The improved calculation formula is as follows:

$$Sim(m,n) = \frac{\sum_{a=1}^{k} T_{m,a} T_{n,a}}{\sqrt{\sum_{a=1}^{k} T_{m,a}^2} \sqrt{\sum_{a=1}^{k} T_{n,a}^2}}$$

(11)
where, after improvement, the measurement formula of the interest degree of user $m$ in items of class $a$ is as follows:

$$T_{m,a} = \frac{N_{m,a}}{G_a} \quad (12)$$

Here, $G_a$ represents the number of items that belong to class $a$, and $N_{m,a}$ represents the total number of evaluations of items in class $a$ by user $m$.

Formula (11) takes full account of the differences in the number of items in each category and the popularity of each category. Formula (12) considers the number of items, to make the calculation as close to the actual recommendation as possible. If the value of $G$ is too low, that is, if the number of items in the category is too small, the value of $H$ will be too high, which will lead to the results of the calculation being inaccurate. Taking the MovieLens dataset as an example, the number of movies that are in the Fantasy category is 22. If formula (12) is used to calculate the user's interest, a high value will be obtained, which is impractical. In this case, we set a threshold $M$. If $G_a > M$, formula (12) is used to calculate the user's interest in an item; if $G_a \leq M$, formula (8) is used to calculate the user's interest in the item. The method of measuring user interest similarity is:

$$Sim_I(m,n) = \begin{cases} \frac{\sum_{a=1}^{k} A_{m,a} A_{n,a}}{\sqrt{\sum_{a=1}^{k} A_{m,a}^2} \sqrt{\sum_{a=1}^{k} A_{n,a}^2}}, & G_a \leq M \\ \frac{\sum_{a=1}^{k} T_{m,a} T_{n,a}}{\sqrt{\sum_{a=1}^{k} T_{m,a}^2} \sqrt{\sum_{a=1}^{k} T_{n,a}^2}}, & G_a > M \end{cases} \quad (13)$$

Recommender systems tend to recommend popular products to users. This phenomenon is called "popularity bias". Popularity bias easily leads to the Matthew effect: the recommender system tends to recommend popular products, which increases the popularity of popular goods, so their probability of being recommended also grows and unpopular products are considered less by users. In this paper, the interest degree measurement method is improved. By calculating the user interest similarity and combining it with the score similarity, the overall similarity between users is finally obtained. This method takes full account of the popularity of each item in the classification, which makes the measurement of interest more reasonable and, to some extent, overcomes the “popularity bias” problem, thereby making the recommendation results more diverse and novel.

3.3. Recommendation process
By combining user interest degree, the similarity between users can be calculated by formula (10) and formula (13). Then, the nearest-neighbor set of the target user can be found, and the predicted scores of non-scored items can be calculated according to formula (4). Finally, the $N$ items with the highest predicted scores constitute the $Top - N$ recommendation result set [18]. For the target user $u$, the appropriate project set $p$ is recommended. The recommendation process is shown in Fig.2.

![Fig.2 Process of the collaborative filtering recommendation algorithm](image)

(1) Algorithm 1: Collaborative filtering algorithm based on user interest degree

Input: "User-Item" matrix; Item attribute matrix; Number of nearest-neighbor users $k$; Size of the recommendation set $N$.

Output: Top-$N$ recommendation set of target user $u$.

Step 1: Establishing a revised "User-Item" scoring matrix. For the union $U$ of target user $u$ and other users, use the nearest-neighbor method to fill in the scores of non-scored items in $U$. The revised "User-Item" scoring matrix was obtained.

Step 2: Establishing a revised "User-Category" scoring matrix. According to the user rating matrix and the item attribute matrix, the number of items in each category of the target user is calculated.

Step 3: The overall similarity between target users and non-target users is calculated. Based on the revised "User-Item" scoring matrix $A$, by formula (3), the similarity based on user score is calculated. By formula (12), the similarity based on user interest degree is calculated. Then, a weight factor $\omega$ is selected, and the final similarity matrix $Sim(u)$ between user $u$ and the other users is calculated according to formula (10).

Step 4: Predicting scores and generating recommendation results. According to the similarity matrix, the neighbor set $S$ of the target user is obtained. Then, according to $S$, the predicted score of target $u$ for item $j$ is calculated by formula (4). Finally, the predicted scores are arranged from high to low, and the largest $N$ items are used as the $Top - N$ recommendation result set.

(2) Algorithm 2: Computation of time complexity.

Hypothesis: There are $m$ users and $n$ goods, and we need to find $k$ neighboring users.
Step 1: Find the nearest-neighbor set of item \( j \) among only items that belong to category \( c \). The time complexity is:

\[
O(m \times nc_j) \approx O(m) < O(m \times n)
\]  

(14)

where \( nc_j \) represents the total number of users who scored item \( j \) in category \( c \).

Step 2: Offline calculation; the time complexity is: \( O(l) < O(m \times n) \).

Step 3: In the revised "User-Item" scoring matrix, the similarity between target users and \( m \) basic users is calculated; the time complexity is \( O(m) \).

Step 4: Sort the \( m \) similarities and find the nearest neighbors; the time complexity is \( O(m \log m) \).

Generate recommendations through the \( k \) nearest neighbors; the time complexity is \( O(k \times m) \). Since \( k \) is much smaller than \( m \), the time complexity is \( O(m) \).

By analyzing the whole process of Algorithm 1, the time complexity of generating the recommendation result set for user \( u \) is \( O(m) \), and the total time complexity is \( O(m^2) \).

4. Conclusion

This paper considers the category information of user-rated items and uses the category information to predict the scores of unscored items in the set of user-rated items. Moreover, when calculating the interest similarity between users, the item classification information is taken into account, which makes the similarity calculation more consistent with the real situation and alleviates the data sparseness problem to some extent.

Acknowledgments

This research work was supported by Middle-age and Young Teachers’ Basic Ability promotion project of Guangxi(2019KY0659), The central government guides the local science and technology development fund project subsidization(GuikeAD20238072), and Doctoral research project of Guangxi University of Finance and economics(K99999150000072)under Grant.

Reference

[1] Gey F, Hearst M and Tong R 1999 Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval International Acm Sigir Conference on Research & Development in Information Retrieval (ACM )

[2] Ahn H J 2008 A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem J. Information ences 178(1) pp 37-51

[3] Jiang L D 2013 Study on the sparsity problem of collaborative filtering algorithm (Chongqing:
Chongqing University)

[4] Breese J S, Heckerman and Kadie C 2013 Empirical Analysis of Predictive Algorithms for Collaborative Filtering J. Uncertainty in Artificial Intelligence 98(7) pp 43-52

[5] Gediminas A A 2005 Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions J. IEEE Transactions on Knowledge & Data Engineering 17(6) pp734-749

[6] Cao G H 2010 Collaborative filtering recommendation algorithm based on uncertain nearest neighbors J. Journal of Computer Science 33(8) pp1369-1377