Journal of Cases on Information Technology
Volume 24 • Issue 5

Application of E-Commerce Recommendation Algorithm in Consumer Preference Prediction

Wei Wang, School of Economics and Trade, Anhui Business and Technology College, China*

ABSTRACT

Through user characteristic information, user interaction behavior, commodity characteristic information, recommendation engine, and related technologies in data mining, this paper makes a more in-depth study and analyzes the problems of "big data volume," "cold start," and "data sparsity" in the recommender system in modern business websites. In response to these problems, this paper transforms the problem of large data volume into the problem of large user groups. Then, after using the k-means clustering algorithm to divide the large user group into homogeneous user groups to alleviate the problem, a combination of collaborative filtering algorithm and content-based recommendation algorithm in the homogeneous user group is proposed to alleviate this problem. The experimental precision and recall are both around 0.4, and when W=0.8, the F value is the largest.

KEYWORDS

Collaborative Filtering Recommendation, Electronic Commerce, Preference Prediction, Recommendation Algorithm

INTRODUCTION

With the advent of the era of big data, the volume of data on various platforms is growing, especially in e-commerce. E-commerce websites will continue to increase the types and quantities of commodities on the website to attract consumers and meet their diverse requirements. Faced with so many products, consumers spend more time searching for a target, a phenomenon known as "information overload." To satisfy users' passive acceptance of recommendations, solve the problem of information overload, and adhere to the principle of providing users with high-quality and better services, it is becoming more and more important to study consumer preference recommendation systems. Recommender systems can collect as much implicit and explicit data as possible about the interaction between users and websites, and recommender systems are considered a more personalized solution.

When shoppers do not know their needs, general business websites will display them to users based on the current popularity of mobile phones. However, when different users search for the word "mobile phone," the content appears the same, and users still need to filter them to obtain them. The recommendation system, in this case, acts as a shopping guide to select personalized products for users. In this way, it tailors a set of products that the user is most likely to be interested in; users no longer get lost in the vast sea of products, effectively solving the problem of information overload.

DOI: 10.4018/JCIT.306977

*Corresponding Author

Hybrid recommendation algorithms process the characteristics of users or items and the interaction data between users and websites, which is also the core of this recommendation system research.

This article studies the problems of "data sparsity," "cold start" and "big data" in the recommendation process through the hybrid recommendation algorithm, solves the problems of insufficient information acquisition, difficulties in generating personalized recommendations, and the scarcity of users' rating data when the recommendation system faces new users, making effective recommendations difficult.

RELATED WORK

With the rapid development of e-commerce, merchants put more categories on the shelves. Understanding consumer preferences and prompting consumers to quickly find their favorite products in many categories requires recommendation algorithms. Wu (2019) analyzed the characteristics of various data in different information sources, proposed a novel recommendation model, which can alleviate the sparsity problem by seamlessly integrating multi-relational data and visual content, and designed a computationally efficient learning algorithm MSRA to optimize the proposed model. The prediction method was introduced into a new vector to represent disease and applied the new vectorized data to a positive unlabeled learning algorithm to predict and rank long non-coding RNA (lncRNA) genes associated with disease (Peng et al., 2017). Bai et al. (2017) studied two main text representations for predicting cross-site purchase preferences, including shallow and deep text features learned by deep neural network models. Using extensive experiments on a large, linked dataset, they provide experimental results showing that leveraging social text to predict purchase preferences is promising. In listening experiments, fan noise signals were adjusted to the same loudness and the same preference compared to the normal reference sound by varying their levels in an adaptive program to quantify how changes in these two indices affect subjects' preference and loudness judgments (Töpken & van de Par, 2019).

Aiming at the two problems of malicious evaluation and the amount of usage affecting prediction accuracy, Zheng et al. (2020) proposed a collaborative filtering recommendation algorithm with item tag features to provide recommendations for users. Experiments show that the algorithm can solve the cold start problem associated with data wells, and the interpretation of the recommendation results is also convincing (Zheng et al., 2020). Collaborative deep learning and its parallelization method was the basis for an improved model for item content optimization, which improved SDAE (stacked denoising auto-encoder) based on CDL (collaborative deep learning) and added private network nodes (Yang et al., 2021). With sharing the network parameters of the model, a private bias term was added for each item, which solved the problem where the recommender system performance dropped sharply when the data is sparse (Yang et al., 2021). Bi et al. (2020) proposed a recommendation algorithm based on a deep neural network where users' basic data and the basic data for commodities are important auxiliary data used to build a regression model for predicting user ratings based on deep neural networks (Bi, 2020). The explosion of reviews has led to a serious problem, information overload. How to mine user interests and understand user preferences from these reviews is critical research and practice. Traditional recommender systems mainly use structured data to mine user interest preferences, such as product categories, user tags, and other social factors. Ma et al. (2018) used the LDA+Word2vec model to mine user interests and proposed a social user emotion measurement method. Finally, they integrated three factors of user topic, user emotion, and interpersonal influence into a recommender system (RS) based on probabilistic matrix factorization. Then, they conducted a series of experiments on the Yelp dataset, and the experimental results show that the proposed method outperforms the existing methods (Ma et al., 2018). Biagi and Falk (2017) provided new empirical evidence on the impact of ICT/e-commerce activities on labor demand. A key feature of the empirical analysis was the use of several advanced ICT activities, and the study's main finding is that increases in ICT/e-commerce activities did not lead to job losses. This applied to manufacturing and services and SMEs and large companies (Biagi & Falk, 2017). These studies are instructive to a certain extent, but there are cases where the demonstration is insufficient or not accurate enough, which can be further improved.

RECOMMENDATION ALGORITHM FOR PREFERENCE PREDICTION

The commonly used recommendation algorithms include those based on demographics, association rules, collaborative filtering recommendation algorithms, and hybrid recommendation algorithms (Cosma & Simha, 2018). Using information retrieval technology, the topic content is cleaned, word segmentation is processed, and finally, each topic is modeled using the feature value, the correlation is calculated, and the recommendation is formed. The interaction information between users and websites such as browsing, collection, and other unstructured data, is relatively large, and the ability to store and calculate these data has become an indicator to measure the quality of a recommendation system (Yuan et al., 2021). We will briefly introduce these algorithms below.

RULE-BASED RECOMMENDATIONS

The analysis of association rules between many data items using rule-based recommendation algorithms can be traced back to the field of data mining (Vinodhini & Chandrasekaran, 2017). The rule-based recommendation takes the purchased item as the rule header and the recommendation object as the rule body. Mining user behaviors generate the relationships and rules between unique items, and items are recommended to users (Robinson, 2017). For example, after analyzing many users, a rule was found: “People who buy beer often buy diapers.” Then when a user buys beer, the system will recommend diapers to him. The rule-based recommendation algorithm mines a large amount of user behavior data and has a more comprehensive grasp of user information, generating a better recommendation model. When a user generates a behavior, they can make recommendations to the user according to the rules. As a result, recommendation speeds are fast and effective because they are developed based on past mining of a large amount of data; mining rules in the early stage have a high time complexity and are insufficiently flexible. With the continuous expansion of the data volume, rules mining becomes difficult, eventually leading to a decrease in the recommendation speed (Min, 2017).

CONTENT-BASED RECOMMENDATIONS

As the name suggests, content-based recommendations (CB) are based on the attributes or characteristics of users or items (Mero, 2018). CB recommendation is the earliest recommendation algorithm, recommending other similar items to users according to the items that users liked in the past; in algorithm operation, the core and key problem is the problem of similarity measurement. CB recommendation differs from demographics-based acquaintance measurement. Because users’ basic information is limited, only limited features must be quantified during quantification. However, the products’ attributes are processed according to the description information of the products, and it is not easy to extract the attributes. CB is proposed to solve related problems in information retrieval to use information retrieval methods in the attribute extraction phase (Figure 1).

Figure 1. Content-based recommendation diagram

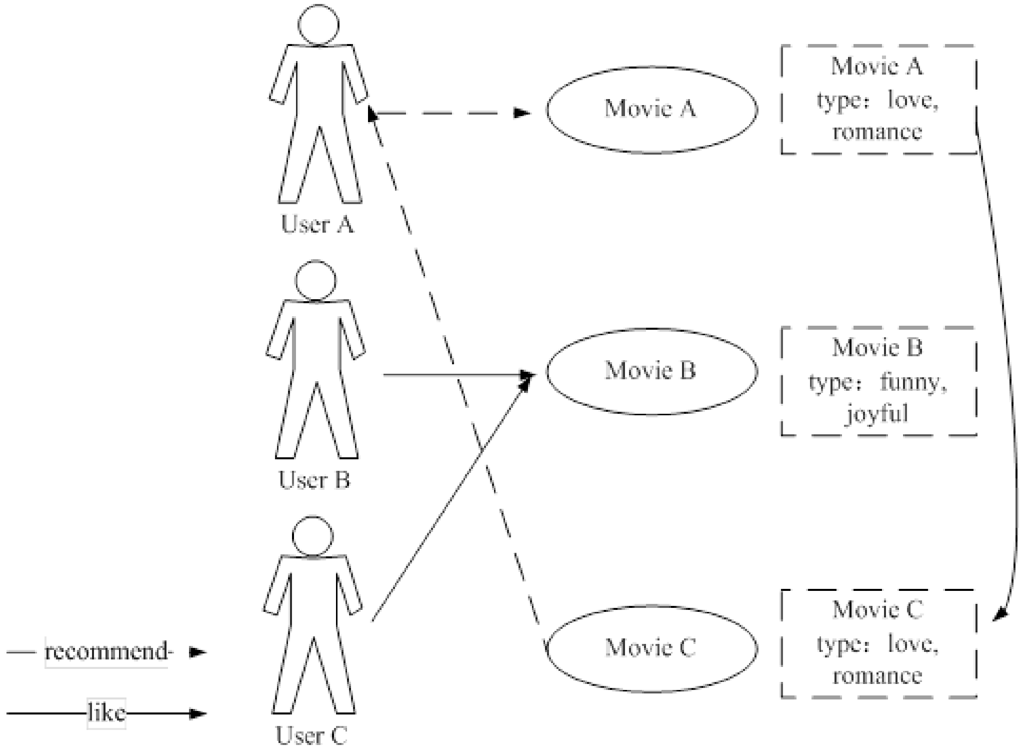
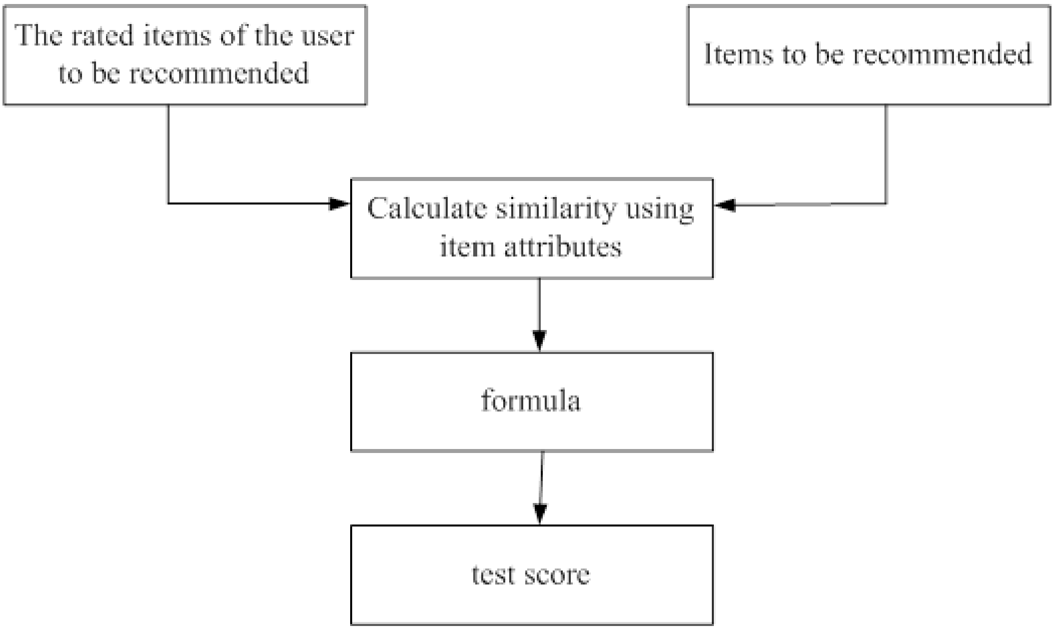


Figure 1 intuitively expresses the idea of the algorithm based on content filtering. First, the recommendation system will analyze the list of all movie attributes according to the content of the movie itself and the attributes of each movie. Then, we calculate the similarity between movies. It can be observed that movie A has the highest similarity with movie C. Because movie A was once liked by user A, the similar movie C is recommended to user A. According to the content-based characteristics, CB is divided into the recommendation algorithm based on the project and user content according to the content filtering algorithm's objective. The user-based recommendation algorithm uses the user's preference for the item and then uses the item's attribute to calculate and generate the user's characteristics. The item-based content recommendation is to calculate the attribute recognition degree between items. This process is like the IBCF processing method, except that the similarity between items is determined by the item's content (feature or attribute). The formula is:

$$\Pr ed(u,p) = \genfrac{}{}{0pt}{}{\sum_{i \in ratedItems} simp,i \times r_{u,i}}{\sum_{i \in ratedItems} simp,i} \qquad (1)$$

The implementation steps of the CB algorithm in this paper are shown as a flowchart (Figure 2).

Figure 2. Content-based recommendation process steps


The rated items of the user to be recommended
Items to be recommended
Calculate similarity using item attributes
formula
test score

The following is an example to illustrate the implementation steps of CB. Table 1 represents the scores of five items by four users, and the scoring range is 1-5; Table 2 shows the attributes that exist in the four items. A zero means none exists, and one means some exist.

Table 1. A matrix of four user ratings for five movies

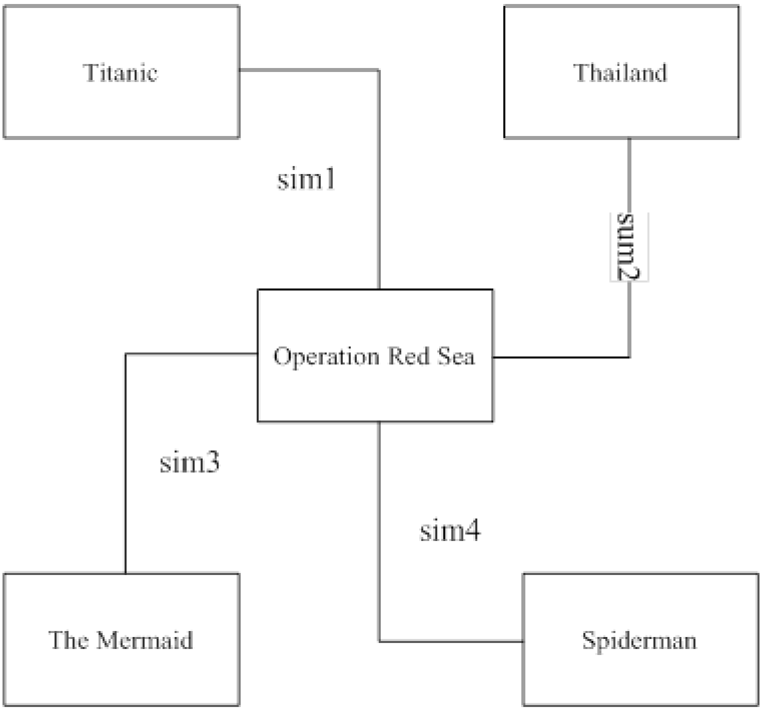
	Thailand	The Mermaid	Spiderman	Operation Red Sea	Titanic
Person A	6	2	4	3	?
Person B	3		5		6
Person C		4		4	2
Person D	5		3	2	3

Table 2. The attribute matrix showing the items for the five movies

	shootout	comedy	romantic	love	action	adventure
Thailand			1	1	1	
The Mermaid		1			1	
Spiderman		1	1	1		
Operation Red Sea					1	1
Titanic	1				1	1

As can be seen from Table 1, to predict the user Person C's preference for the movie *Operation Red Sea*, the first step is to calculate similarities between *Titanic*, *Thailand*, *The Mermaid*, *Spider-Man*, and *Operation Red Sea*. The between-movie similarity scores are shown in Figure 3.

Figure 3. Feature association



The Euclidean distance is generally used to calculate the attribute similarity, and there is a distance between the item attribute vectors a $\left(x_1, x_2, x_3, \cdots, x_n\right)$ and b $\left(y_1, y_2, y_3, \cdots, y_n\right)$:

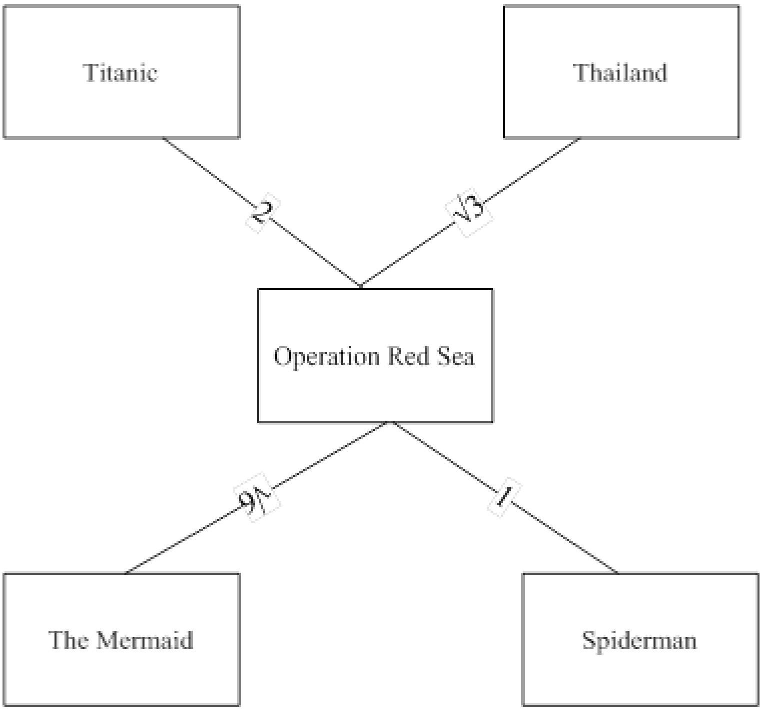
$$O(a,b) = \sum_{a=0}^{n} \left(x_a - y_a\right)^2 \tag{2}$$

The distance also needs to be normalized to the similarity for calculation.

$$Sim(a,b) =_{1+Oa,b}^{1} \tag{3}$$

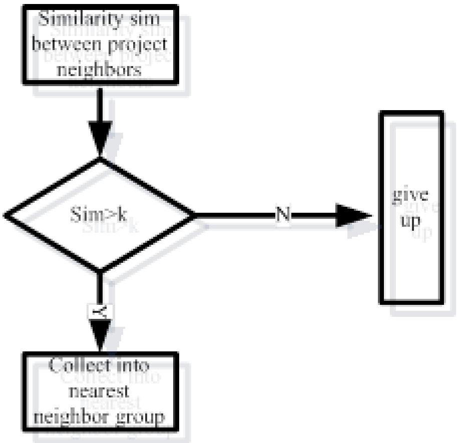
where 'sim1' in Figure 3 represents the similarity between the eigenvectors [0,1,1,0,1,0] of *Titanic* and the eigenvectors [0,0,0,1,1,1] of *Operation Red Sea*. First, it is necessary to calculate the Euclidean distance between the two vectors in combination with formula (2): $O_1 = 0 - 0^2 + 1 - 0^2 + 1 - 0^2 + 0 - 1^2 + 1 - 1^2 + 0 - 1^2 = 2$; similarly, $O_2 = 3; O_3 = 6; O_4 = 1$ can be obtained.

Figure 4. Euclidean distance representations



As the user has interacted with multiple products on an e-commerce website, there will be many similarities of this nature for the recommendation system to use. To develop more accurate recommendations, we must set a similarity threshold k to exclude the influence of distant items on user preferences (Wang et al., 2018). As shown in Figure 4, if a threshold greater than sim3 is set, then only sim1, sim2, and sim4 must be processed. After the processing is completed, the nearest neighbor will be obtained, as shown in Figure 5.

Figure 5. IBCF nearest neighbor selection flowchart


Similarity sim between project neighbors
Sim>k
N
give up
Y
Collect into nearest neighbor group

After obtaining the nearest neighbors, formula (1) is used to predict Person C's score for the movie *Operation Red Sea*: $P =^{20.5+4_3^1+3\times1}_{0.5+\frac{1}{3}+1}= 3.05$

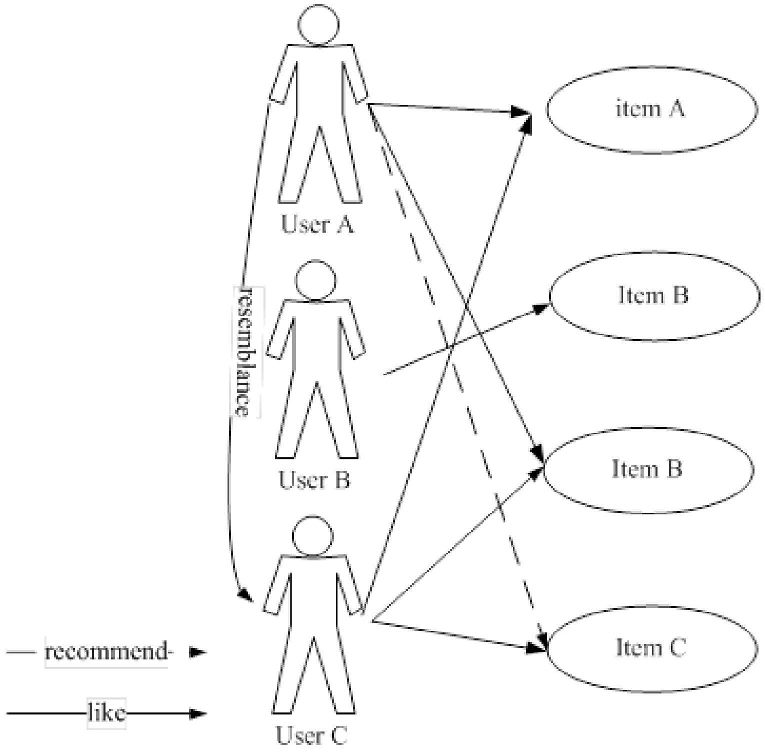
COLLABORATIVE FILTERING ALGORITHM RECOMMENDATION

Collaborative filtering algorithm recommendation is currently the most researched, most widely used, and most successful recommendation algorithm that has been at the forefront of recommender system research (Arnold et al., 2018). It is based on two assumptions; first, users with similar preferences are assumed to have similar preferences for other aspects. Second, there will not be a substantial preference change. Since its inception, the collaborative filtering algorithm has been used in various recommendation systems and is known as the most successful recommendation method.

(1) User-based recommendation

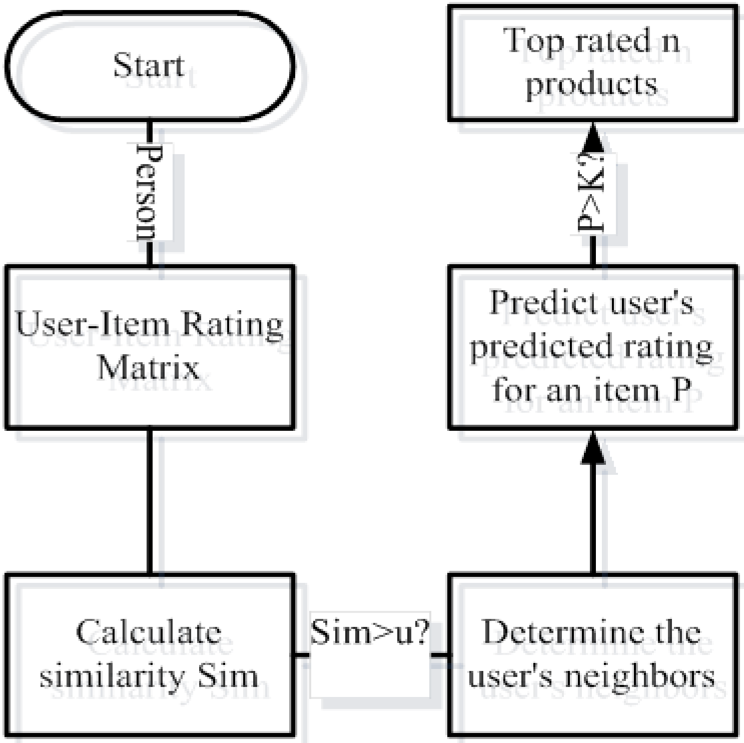
The basic concept of the user-based recommendation system is that users will ask friends with similar tastes to understand what their favorite shoes are. Then they will be less unclear about what type of shoes are good as they will understand their friends' recommendations.

Figure 6. Principle of a user-based collaborative filtering algorithm



As shown in Figure 6, user A likes items A and C; user B likes item B; user C likes items A and C. Therefore, user A's interests are the same as C's, so user C's favorite item D will be recommended to user A. The user-based collaborative filtering algorithm implementation steps are explicated and are depicted in Figure 7.

Figure 7. The user-based collaborative filtering (UBCF) process recommendation diagram



The following is an example to illustrate the implementation steps of the user-based collaborative filtering algorithm. Suppose a scoring matrix is shown in Table 3, where Step1 is to calculate the similarity and determining the nearest neighbor.

Table 3. An example of a user-item rating matrix

	Item 1	Item 2	Item 3	Item 4	Item 5
Person A	5	3	2	4	?
Person B	2	4	3	3	4
Person C	4	4	2	2	4
Person D	1	5	5	2	2
Person E	3	2	1	2	2

Table 3 shows the five users' preference values for five items as a rating matrix. Using these data, we must predict the predicted rating of Person C for item five. According to Figure 7, we must first compute and identify the nearest neighbor. Here, the Pearson correlation coefficient (Pearson) is used to calculate the degree of acquaintance. The Pearson correlation formula is as follows:

$$SimU_i,U_j = \frac{\sum_{a\in atems} R_{i,a} - \overline{R_i} R_{j,a} - \overline{R_j}}{\sum_{a\in atems} R_{i,a} - \overline{R_j}^2 \sum_{a\in atems} R_{j,a} - \overline{R_j}^2} \tag{4}$$

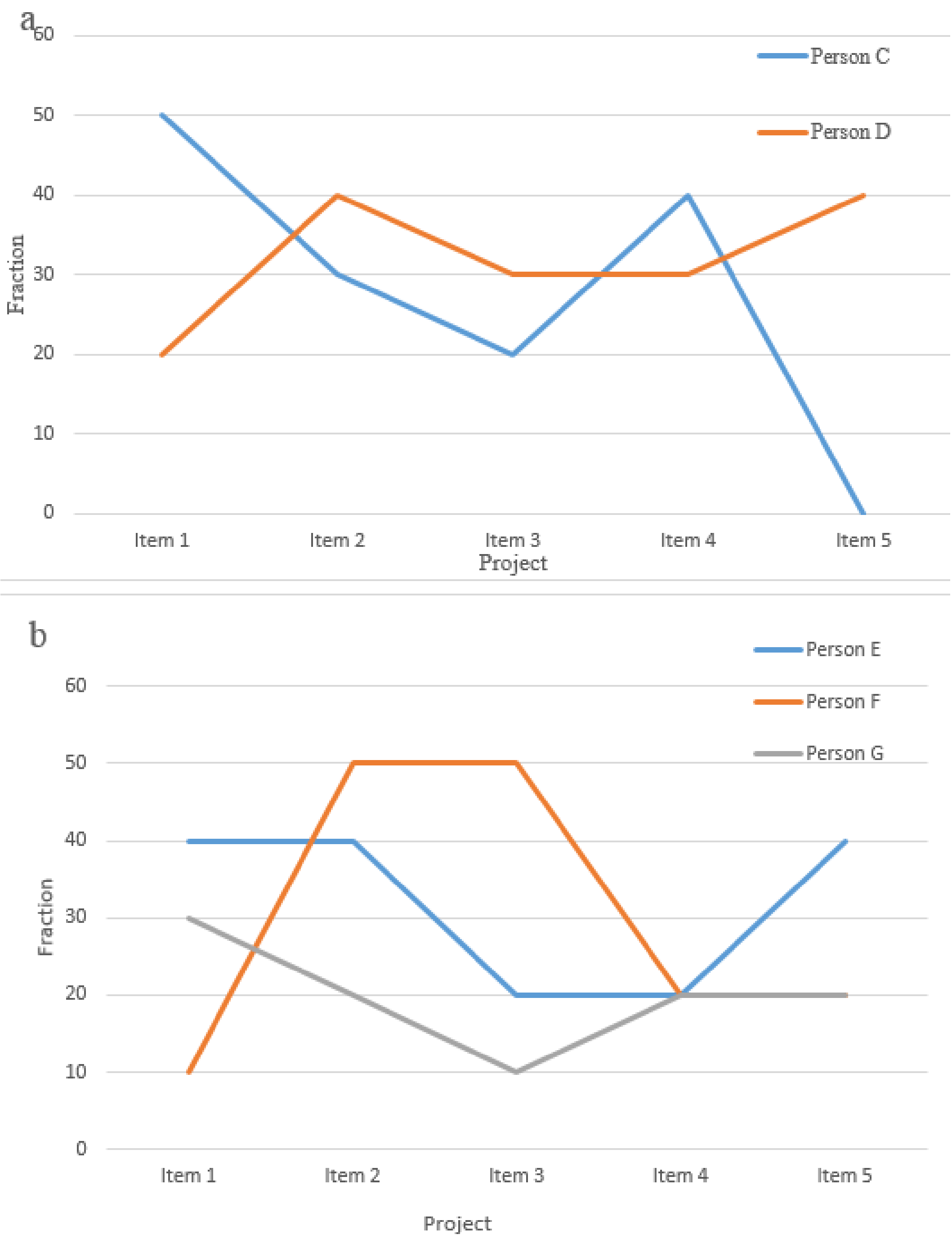
where items represent the total set of items rated by users; $R_{a,i}$ represents the rating of item i by user a; and, $\overline{Ra}$ represents the average rating of user a. Using formula (4), the correlation between Person C and Person D is calculated. The mean of Person C scores is $\overline{\mathrm{Ri}} = \frac{5+3+2+4}{4} = 3.5$. The mean of Person D scores is $\overline{R}\mathrm{j} = \frac{2+4+3+3+4}{5} = 3.2$. After calculation, the similarity between Person C and Person D is $\overline{R}\mathrm{j} = \frac{2+4+3+3+4}{5} = 3.2$. The Pearson correlation between users can be obtained using this calculation method with the results shown in Table 4.

Table 4. Pearson correlation coefficient calculation results

	Person C	Person D	Person E	Person F
Person D	-0.63452			
Person E	0.508248	0.2182		
Person F	-0.79642	0.4791	0.195180	
Person G	0.835414	-0.428	-0.10910	-0.7985

The rating curves of the five users are shown in Figure 8, and they allow us to compare the effect of subtracting the average value of user ratings in the Pearson correlation calculations, according to Table 4.

Figure 8. The five-person rating track



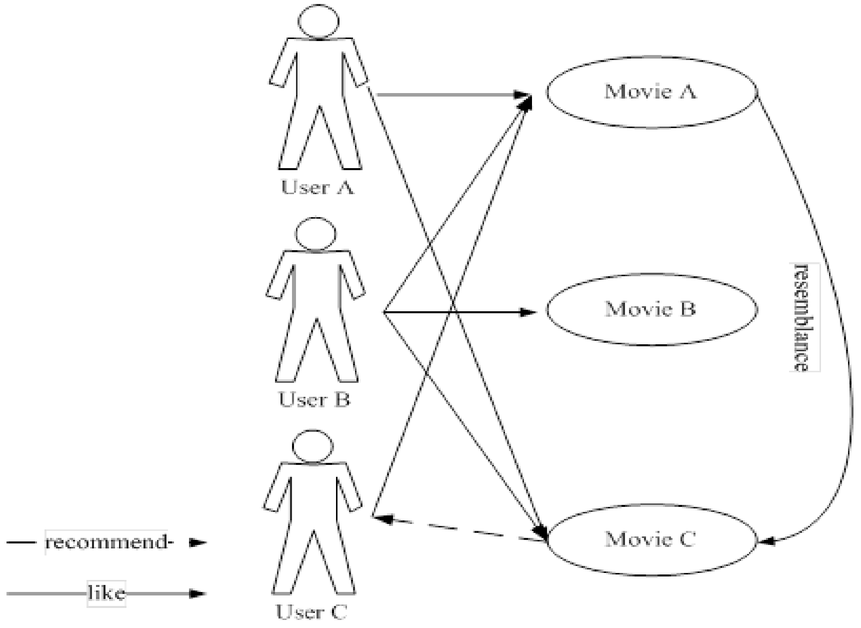
This issue can also be reflected in Figure 8; although Person C and Person G have significant differences in the scores, the score curves are the same. It shows that the degree of item preference

of Person C and Person G are related, and the two are a class of people. It can also be seen from the data that the Pearson correlation between Person C and Person G is 0.835414, the largest correlation. After calculating the similarity, the next step is to divide the nearest neighbors. Generally, the nearest neighbors are divided according to the threshold of acquaintance.
Calculating the predicted score: With 0 as the threshold, the predicted score of Person C for item5 can be solved using the following formula:

$$Pip = \overline{R_i} + \frac{\sum_{j \in n} simi,j \times Rjp - \overline{R_j}}{\sum_{j \in n} simi,j} \tag{5}$$

When predicting the score of Person C, it can be seen from step 1 that by subtracting the average value of the item score, the influence of the user's difference in the item score on the recommendation result is removed. Using formula (5), it can be known that the predicted score of Person C for item 5 is: $pi, p = 3.5 +^{0.408248*4-{}_{5}^{4+4+2+2+4}+0.935414*2-{}_{5}^{3+2+1+2+2}}_{5} = 4.0$
Step 3: Setting the score threshold and generating the recommended topN. Since there is only one vacancy in Person C in Table 4, we must predict only one score for item five. However, on an e-commerce website, the preference matrix is sparser than Table 4, and it is necessary to use predicted scores to fill all the table vacancies. Furthermore, the recommendation generation stage is not for a single item but for all the user's vacant items. Therefore, it is unrealistic and unnecessary to recommend every item in the predicted rating items. In general, a score threshold K is determined according to the size of the recommended pit N, and only items with a predicted score greater than K can fill the recommended pit; finally, the items in the pit are recommended to the user.

Figure 9. Item-based collaborative filtering algorithm



(2) Item-based recommendation

As shown in Figure 9, since user C likes item A, then item C acquainted with the item A is recommended to User C. This is the recommended based on IBCF; the implementation steps of IBCF are similar to UBCF, and the difference is in the selection of the nearest neighbors. IBCF selects the nearest neighbors of the items. In the calculation of item similarity, Table 4 is used for analysis. First, the cosine degree of acquaintance (cosine) is used to calculate the degree of acquaintance between item1 and item5. The formula is:

$$Sim\ (\mathrm{a},b) = \cos a,b = \frac{\vec{a}\bullet\vec{b}}{\vec{a}\ \left\|\vec{b}\right\|} \tag{6}$$

In formula (6), the vector a and the vector b respectively represent the preference vector of the item. Therefore, Item1 can be represented as a vector a1=[2,4,1,3], and similarly, the vector of Item5 can be represented as a5=[4,4,2,2]. Consequently, the degree of acquaintance between the two items can be computed as:

$$\mathrm{s}im(item1,item5) = \cos(a_1,a_5) = \frac{2*4+4*4+1*2+3*2}{2^2+4^2+1^2+3^2*4^2+4^2+2^2+2^2} = 0.92 \tag{7}$$

The principle of using cosine in the degree of acquaintance uses the cosine value of the angle between two vectors to represent the degree of acquaintance, which extends the cosine theorem; the theorem still applies when extending a two-dimensional vector to an N-dimensional vector. Therefore, the value range of Sim(a,b) is (-1,1). The smaller the angle between the two vectors, the more consistent the two vectors are, the more similar the items represented by the vectors, and the higher the cosine. Cosine similarity does not account for the impact of user rating differences on the calculation results of acquaintance. Therefore, the adjusted cosine similarity (acosine) is used for comparison in the calculation of item similarity, as shown in formula (8):

$$Sim\ \left(a,b\right) = \frac{\sum_{u\in U}(R_{u,a}-\overline{R_i})(R_{u,b}-\overline{R_a})}{\sum_{u\in U}(R_{u,a}-\overline{R_i})^2(R_{u,b}-\overline{R_a})^2} \tag{8}$$

where $\overline{R_a}$ and $\overline{R_\mathrm{b}}$ distinguish the Pearson average; while the former calculates the average of all user preferences for the item, the latter calculates the average of user ratings. The former is for projects, and the latter is for users. Acosine ignores the effect of user preferences on items by subtracting the average of the items being rated. Compared to cosine, user preferences have been corrected (Zinko et al., 2021). The revised preference information is as follows.

Table 5. Revised score sheet

	Item 1	Item 2	Item 3	Item 4	Item 5
Person C	1.5	-0.5	-1.5	0.5	?
Person D	-1.2	0.8	-0.2	-0.2	0.8
Person E	0.8	0.8	-1.2	-1.2	0.8
Person F	-2	2	2	-1	-1
Person G	0.8	-0.2	-1.2	0.8	-0.2

Table 5 calculates the degree of acquaintance between users, Item1, and Item5 again. At this time, Item1 and Item5 can be expressed as [-1.2, 0.8, -2, 0.8] and [0.8, 0.8, -1, -0.2]. The modified cosine similarity between the two items is computed as $sim(item1, item5) = \cos(I_1, I_5) = 0.38$

Five people rated the two items to show the correction effect of acosine (Figure 10).

Figure 10. Scoring curves for two items

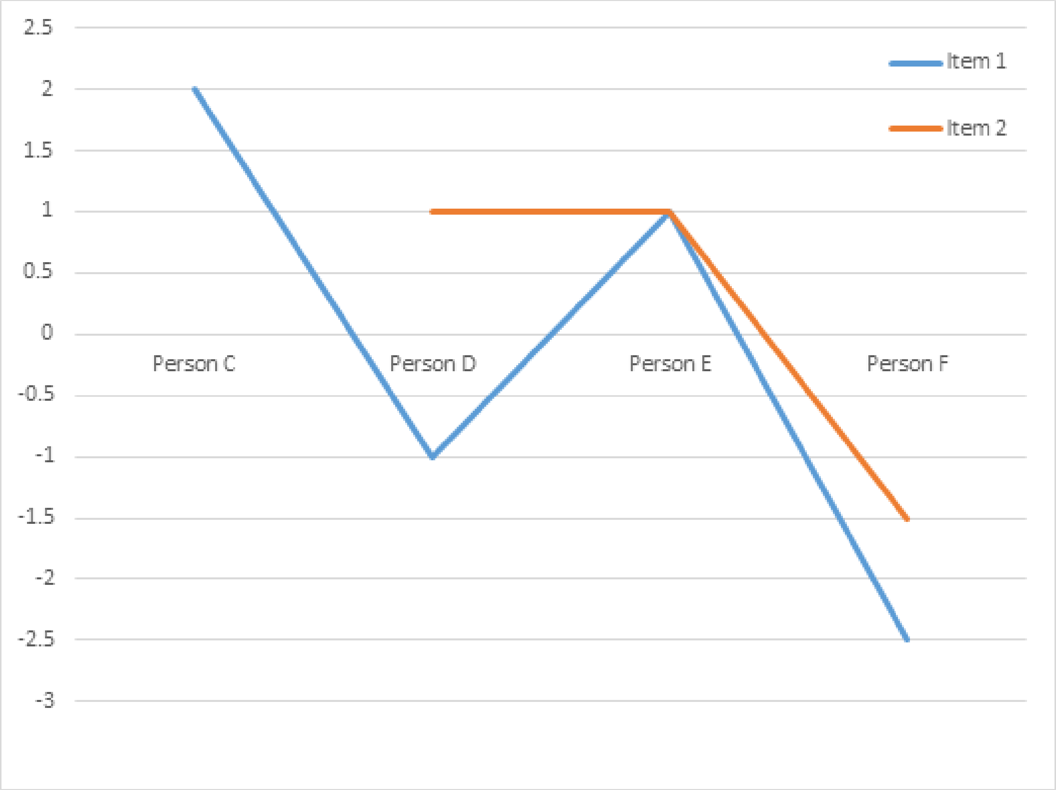


Figure 10 depicts the graphs of Item1 and Item5 being rated by five users. By observing the degree of curvature of the curves, we observe the trend graphs of the two curves are distinct. The similarity between the two items calculated by cosine is 0.92, so the two items are highly similar; this observation is because of the different scoring standards of the five people on the items. For example, Person C gives five points for their favorite items, and Person D gives only four points for their favorite items, which results in an inaccurate similarity calculation. The acquaintance calculated by acosine is 0.38, closer to the real similarity than cosine. Because acosine uses the method of subtracting the average value of the item's rating, it ignores the impact of the user's rating criteria on the item. The calculated item similarity is closer to the latent item similarity, so acosine is used to calculate the score similarity in the experimental calculation process. After calculating the degree of acquaintance, it is necessary to calculate the predicted score of Person C for item 5. The formula is:

$$\Pr edu, p = \genfrac{}{}{0pt}{}{\sum_{a\in ratedItems} simp,a\times r_{u,a}}{\sum_{a\in ratedItems} simp,a} \tag{9}$$

The numerator in (9) calculates the multiplication set of the degree of acquaintance between the item to be recommended and the item that the user has preferred, and the rating value of the item that the user has preferred. The denominator represents the sum of item similarities. The calculation reduces the contribution of items with a low acquaintance to the predicted score. Therefore, it is necessary to set a project nearest neighbor threshold and projects larger than the threshold can take part in the scoring contribution.

(3) Comparison of the two algorithms

UBCF and IBCF are relatively basic recommendation algorithms. UBCF is recommended based on the acquaintance between users. IBCF is recommended based on the similarity between items. As a result, UBCF is generally used in news and friend recommendations. IBCF is used in areas where project content changes slowly. IBCF is biased toward personalized recommendations because it predicts the likely preferences of future users based on the user's historical preferences (Jin et al., 2017).

PROBLEMS IN COLLABORATIVE FILTERING ALGORITHMS

Data Sparsity Problem. Compared with the number of all items, the number of user interactions becomes small (Li & Ku, 2017); consequentially, there are sparse user-item rating vectors, which leads to sparse user-item rating matrices, which can lead to inaccurate similarity calculations (Jing et al., 2018).

At present, for the problem of data sparseness, there are the following methods: direct filling method, calculating the average score of users, and filling the vacancies in the score matrix with this average user. This method is relatively simple to implement, but the processing process is relatively rough. Although this solution is simple and easy to implement, it solves the sparsity problem to a certain extent and improves the recommendation quality relatively. The cluster filling method uses the clustering method to compute the center points of all the items that the user has preferred and uses the preferences of the center points to fill the matrix (Hajli & Featherman, 2017).

Cold start problem. When a new user or a new product appears in the recommendation system, since no information about interaction with the website can be obtained, it is natural that the user's preference cannot be extracted (Liao & Ho, 2021). Therefore, solving the cold start problem greatly improves website user loyalty and sales. The most common method uses registration information, popular product recommendations, and item feature vector recommendations (Chen, 2018).

HYBRID RECOMMENDATION

Through the research on commonly used recommendation algorithms, we know the advantages and disadvantages of a single algorithm. However, if a recommendation system only applies CF, then although it can get novel recommendations, when a new product is introduced but not discovered by users, the product will never be recommended (Yang et al., 2017). Therefore, single algorithms are often combined to complement each other and optimize the recommendation effect while overcoming defects associated with single recommendation algorithms. A review of common combinations follows.

There are five popular combination methods. First, the weighting method. It is generally used in recommendation systems with multiple engines. It is necessary to integrate the results of multiple recommendation engines when generating a recommendation for a user. There is no relationship between recommendation engines, but when generating the recommendation list, each engine weights the recommendation results according to the weights assigned by the system. In the result recommendation stage, the weighted recommendation items of each engine are rated and sorted (Yang & Sun, 2017).

Second, the switching method. It is necessary to use the advantages and disadvantages of various algorithms for different situations to select the dominant algorithm to deal with the problem.

Multiple recommendation engines are required to apply this hybrid recommendation system, and we must select an appropriate algorithm for each recommendation engine to ensure the accuracy of the recommendation system (Zhao et al., 2017). Otherwise, the algorithm is used in confusion, which will reduce the recommendation effect.

Third, the partitioning method. Amazon uses this method effectively, dividing the presentation layer of the system into different areas and using different algorithms according to the product characteristics of each area. For example, if UCF is used, then the requirement of the area may be "goods liked by similar people;" if ICF or CB is applied, then the requirement may be "goods liked" or "these goods are very similar to preferences." The partitioning method provides users with a more comprehensive recommendation (Kant & Mahara, 2018). The weighting method, the switching method, and the partitioning method are all parallel combination methods of the recommendation engine, and the recommendation engine is not related in the calculation stage.

Fourth, is integral mixing. This approach fuses the implementation steps of two or more algorithms.

Fifth, the hierarchical hybrid approach is when two or more recommendation mechanisms are used in sequence. We use the result of one recommendation mechanism as the input of the other to synthesize the advantages and disadvantages of each recommendation mechanism to enhance recommendation accuracy.

APPLICATION RESEARCH OF E-COMMERCE RECOMMENDATION ALGORITHM IN CONSUMER PREFERENCE PREDICTION

We can summarize the three common problems in recommendation engines by researching commonly used recommendation algorithms. The first is the problem of large data volume. In the context of more and more users accustomed to online shopping, the number of users and products is very large. The role of the recommendation engine is to use this data to generate rating predictions. It addresses how the recommendation system should deal with such a large amount of data.

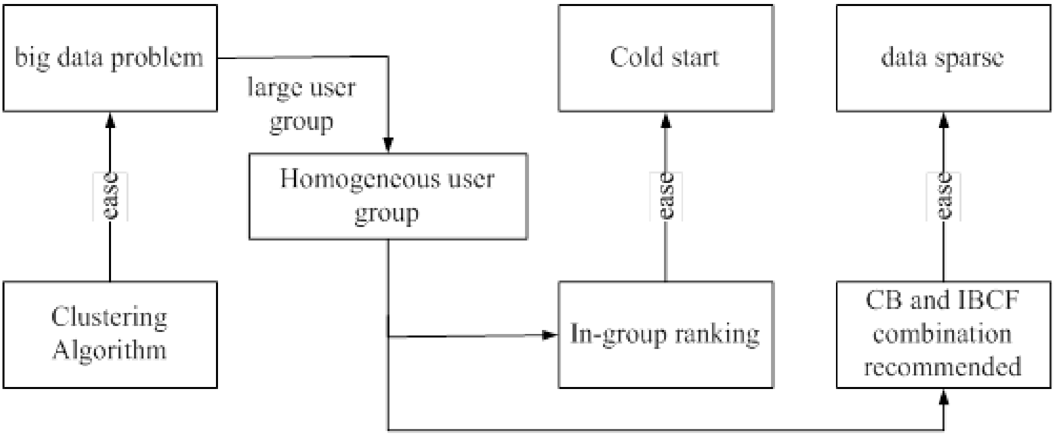
The second is the cold start problem. It refers to how new users can be recommended to generate personalized recommendations due to the lack of interaction data with the website, and it is impossible to model user interaction behaviors and generate a scoring matrix. This interaction is another problem that recommendation engines face.

Third is the data sparsity problem. In real shopping applications, the M×N order preference matrix formed by user items is particularly sparse. The data sparsity may only reach 0.01 or less. If the matrix is too sparse, the recommendation effect of the recommendation system will be worse. This sparsity is the primary reason for the insufficient accuracy of recommender systems.

USE GROUPING TO ALLEVIATE THE COLD START PROBLEM

This paper takes the user group as the starting point when faced with the problem of a large volume of data. On the problem of grouping, this paper uses theories from economics and psychology. When faced with a large heterogeneous market, it is first necessary to divide the market into multiple homogeneous consumer groups so that each small consumer group has similar basic characteristics. First, divide users into different groups because of the cold start problem where the user has no behavior record. However, all registered users of the mall have basic registration information. Therefore, consider using this registration information to model users and use the correlation algorithm in cluster analysis to divide these users into different homogeneous user groups. Second, this paper adopts the method of scoring their interactive items by homogeneous user groups to recommend to new users to ease the "cold start" problem faced by the recommendation engine. For the "data sparsity" problem, this paper finds that the similarity calculation is inaccurate because the scoring matrix is too sparse. Therefore, the IBCF and CB algorithm combination is considered to improve recommendation accuracy. Figure 11 depicts a flowchart of the solution to common problems in recommender systems.

Figure 11. Solution flow diagram



Since we divide this grouping according to users' basic information, it does not depend on preferences and is aligned with the characteristics of new users. Therefore, it is appropriate to use this approach to solve the cold start problem for new system users. Furthermore, we only need to recommend items with higher ratings in the new user's group to the new user. Therefore, we use the K-means clustering algorithm to first group the large user groups, which can reduce the dimension of the preference matrix and alleviate the cold start problem. The following steps describe the K-means method's specific process to ease the cold start problem.
Step 1: Build a user vector.
Step 2: Set the scoring threshold, take out the user items of the homogeneous group, and form a recommendation.

CLUSTERING ALGORITHM DESCRIPTION

The first step in designing the hybrid recommendation engine in this paper is to group the huge user groups in the recommendation system, and we can use the related algorithms of cluster analysis in the grouping.

There are also many classifications of clustering algorithms, such as partition, hierarchical, model-based, and grid-based methods. For example, the k-means clustering algorithm used in the algorithm preprocessing stage is a method based on partition. The following introduces the K-means clustering algorithm often used in the reclustering analysis.

In the K-means algorithm, K represents the division of a group into K categories, and 'means' represents the mean. Consequently, it is sometimes referred to as 'the mean algorithm.' The basic idea is to select any K objects as the center point of the original cluster in the data with n data objects. Then, we compute their distance from these initial cluster center points for other objects to be clustered. After assigning them to the class with the largest similarity cluster center, we then calculate the clustering from each point to the center point in these clusters and re-divide a new cluster center for each cluster. Iterate the process until the normalization function converges. The commonly used method is the squared error, defined as:

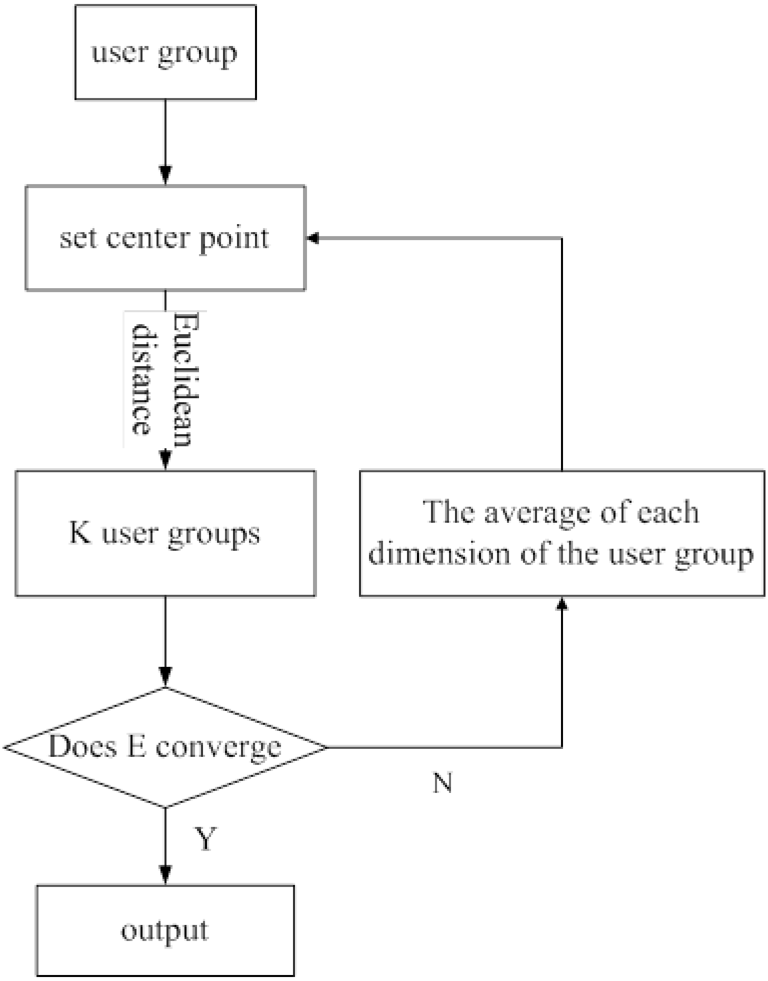
$$E=\sum\nolimits_{a=1}^{k}\sum\nolimits_{p\in C_{\mathrm{a}}}\left|p-m_{a}\right|^{2} \tag{10}$$

where E represents the sum of scoring errors for all objects, P is a point in space, and m_a is the mean of C_a. For each object in the cluster, find the square of the distance from the object to its cluster center and then sum it up. This criterion makes the K results produced with relatively high cohesion. The six implementation steps are:

1. Select K elements in the target data set as the cluster center.
2. Calculate the distance from each point to the centroids, and cluster each point into the nearest cluster.
3. According to the division result of the previous step, calculate the average value of each point in each dimension in the new cluster to form a new cluster center point.
4. Repeat the second step.
5. The clustering results no longer change.
6. Output the result.

We assume that there is a vector set composed of N users in the system, and we assume that each user has M eigenvalues. We arbitrarily take K M-dimensional vectors as the center point of clustering and then use the Euclidean distance to calculate the distance between user's vector U and the center point. Next, divide the users into the groups closest to the center point, calculate each user's average value in each dimension, take the M averages as the new center point, and repeat the steps until the clusters no longer change. After the steps, we divide users into N different groups, as shown in Figure 12.

Figure 12. User grouping flowchart



DESIGN OF COMBINATION RECOMMENDATION ALGORITHM

Commonly used algorithms in recommendation engines include user-based collaborative filtering algorithms, item-based collaborative filtering algorithms, demographic-based collaborative filtering algorithms, and content-based recommendation algorithms.

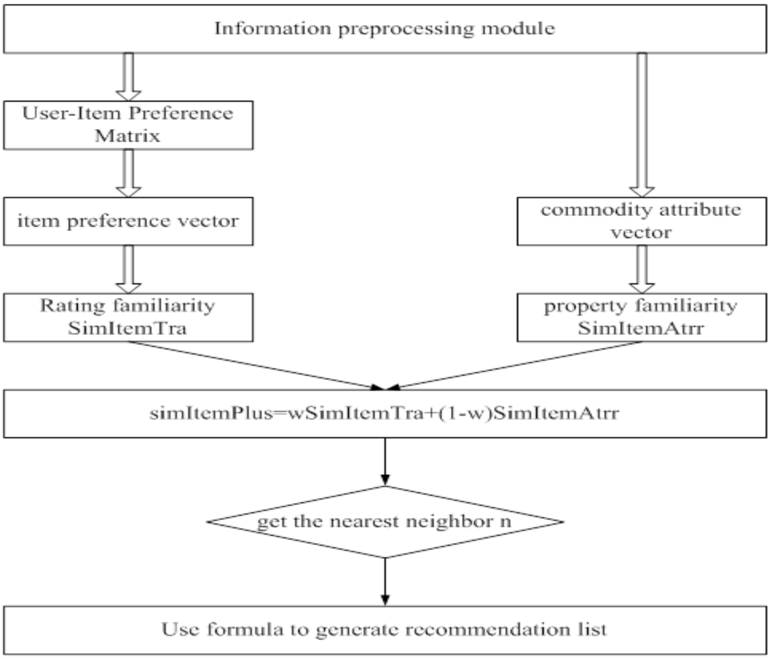
Considering that the recommendation system will affect the accuracy of the recommendation system because of the sparse data in the scoring matrix, two combined recommendation schemes are designed and introduced here.

Option 1: The idea of combination is to use an integral combination of CB and IBCF to combine the computational similarity parts of the two algorithms. First, the item-user preference vector is used to calculate the rating acquaintance $SimItemTraa,b$ between items, and then we calculate the attribute acquaintance $SimItemA\mathrm{trr}(a,b)$ between items using the item's attribute vector. Next, we set a dynamic weight w for these two similarities and finally calculate the comprehensive acquaintance $SimItemP\mathrm{lus}(a,b)$ between the items. The calculation is:

$$SimItemPlus(a,b) = w * SimItemTraa,b + 1 - w * SimItemTrr(a,b) \tag{11}$$

After obtaining the comprehensive similarity, follow the same steps as IBCF to predict the predicted score of the item. The algorithm flow is shown in Figure 13.

Figure 13. The flow of the overall hybrid scheme of the two algorithms


Information preprocessing module
User-Item Preference Matrix
item preference vector
commodity attribute vector
Rating familiarity SimItemTra
property familiarity SimItemAtrr
simItemPlus=wSimItemTra+(1-w)SimItemAtrr
get the nearest neighbor n
Use formula to generate recommendation list

Scheme 2: As shown in Figure 13, this scheme is based on the data sparsity of the scoring matrix, which will affect the recommendation effect. Therefore, the method of converting the sparse matrix into a dense matrix is considered first, and a scheme of hierarchically combining the two algorithms is proposed. There are three steps in the scheme. First, use a content-based recommendation algorithm to predict the ratings for the gaps in the rating matrix and fill the gaps in the sparse matrix. Second, calculate the scoring acquaintance between the filled matrix items. Third, predicted scoring. We give an example to describe the idea of Option 2. Suppose we have a scoring matrix, as shown in Table 6, and an attribute matrix in Table 7.

Table 6. User item rating matrix

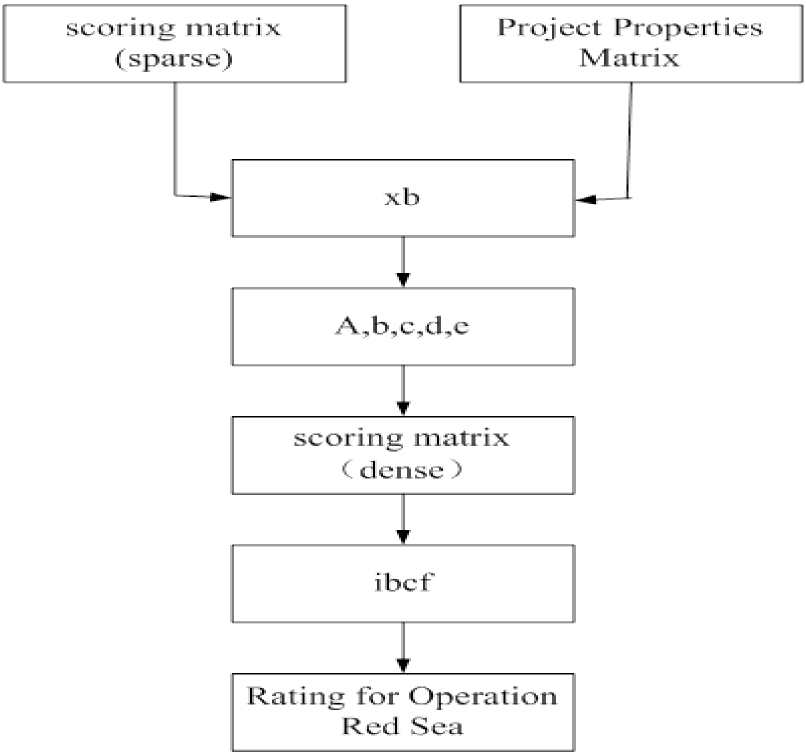
	Thailand	*The Mermaid*	*Operation Red Sea*	*Titanic*
Person C	1	4	6	?
Person D	A	5	D	1
Person E	3	B	1	E
Person F	2	C	D	5

Table 7. Project properties matrix

	Action	Comedy	Romantic	Love
Thailand			1	1
The Mermaid	1	1		1
Operation Red Sea		1	1	1
Titanic	1			

According to the design of the second scheme, the user Person C's preference for the *Operation Red Sea* operation is predicted. For the matrix vacancy A to be filled, the steps of CB are used first to calculate the predicted score of Person C to the Titanic, and it is known that Person E's rating score of *Thailand* is 3. Person C has a rating of 6 for *Operation Red Sea*, so we need to calculate the similarity between *Thailand* and RMS *Titanic*. The example stage here uses cosine to calculate the degree of acquaintance to facilitate the calculation: $Sim1 = (1\times 0 + 0\times 1 + 1\times 1 + 1\times 0)/1\times 2 = 1/2$. In the same way, the attribute recognition between *Operation Red Sea* and the *Titanic*: Sim2=0. Therefore, the predicted score A of the movie *Titanic* by Person E calculated by CB can be computed using Formula (1) A=4. In the same way, we can fill in all the vacancies in the matrix, then feed the filled matrix into the IBCF engine to get a predicted score for *Operation Red Sea*. Figure 14 is a flowchart of a hierarchical hybrid scheme of two algorithms.

Figure 14. Hierarchical hybrid flow chart of two algorithms



PREDICTION INDICATORS OF RECOMMENDER SYSTEM ACCURACY

Accuracy measures the predictive ability of a recommender system, and it is an offline evaluation indicator. Many papers discuss this metric and use it to evaluate algorithmic improvements and algorithmic innovations. Evaluating this indicator requires two datasets, one as the basis for recommendation and the other as the basis for testing. We can calculate this metric based on the degree of coincidence of the recommended scores in the test set. It is generally calculated based on precision, recall, F-value, and MAE (mean absolute error). Here are some commonly used methods of predicting accuracy.

The evaluation index can evaluate the recommendation effect of the recommendation algorithm in the recommendation system. There are many evaluation indicators with different evaluation standards from different perspectives. In the experiment, the scoring indicators are precision and recall. The accuracy rate represents the user's probability of liking the given recommendation list. The recall rate represents the probability that they recommend the product that the user likes in the recommendation list given by the system. When the recommendation system recommends products that have no preference with the target user, there are generally four situations: 1. The user likes the product recommended by the recommendation system. 2. The recommendation system recommends the product, but the user does not like it. 3. The user likes the product, but the recommendation system does not recommend it. 4. The user does not like the product, and the recommendation system does not recommend it. N_{tr} , N_{tn} , $N_{\mathrm{f}r}$, N_{fn} represents the number of commodities in four cases, respectively. Let the total number recommended by the system be N, and the total number users like is M.

$$N = N_{tr} + N_{\mathrm{f}r}; M = N_{tr} + N_{tn} \tag{12}$$

Then the formula for calculating the accuracy rate:

$$\mathrm{Pr}\,ecision = \frac{N_{tr}}{N} \tag{13}$$

The formula for calculating recall rate:

$$\mathrm{Re}\,ca;; = \frac{N_{tr}}{M} \tag{14}$$

We can see from the two Formulas that the accuracy rate is the proportion of the target results in the evaluation recommendation results, and the recall rate is the proportion of accurate recommendations in the number of recommended recommendations. Therefore, to comprehensively consider the two factors, the weighted harmonic mean of precision and recall is used to evaluate the quality of the recommendation algorithm. The calculation is:

$$F = \frac{2*precision*recall}{\mathrm{pre}cision+recall} \tag{15}$$

Mean absolute error (MAE). The evaluation index mentioned in the previous section judges the recommendation system's accuracy from the recommendation pits' perspective. MAE is calculated based on the coincidence of the predicted scores with the test set data. The formula for calculating MAE value is:

$$Mae = \frac{\sum_{u,i\in T} r_{ui} - \bar{r}_{ui}}{T} \tag{16}$$

where i represents the item, u represents the user, r_{ui} represents the actual rating of the item, and $\bar{r}_{ui}$ represents the predicted rating of the recommendation engine.

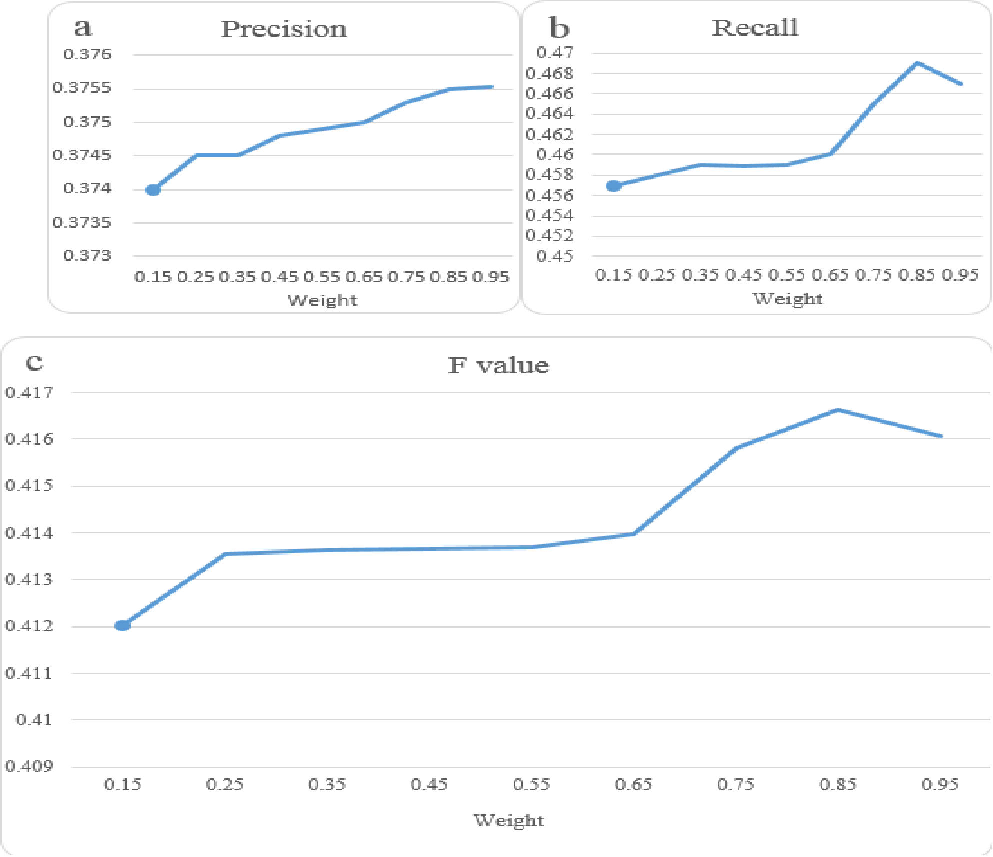
EXPERIMENTAL PROCESS AND EXPERIMENTAL RESULTS

Experimental settings:

Test the com1-author combination algorithm on the movie_lens 1M data set; the pace is 0.1, and the training set is a test set = 8:2.

1. Evaluation indicators: precision rate, recall rate, and F value.
2. Experimental purpose: to determine the optimal weights.
3. Experimental results: we show the precision and recall curves under different weights in Figure 15.

Figure 15. Precision, recall, and generated F-value curves



From the change curve of the weighted average F of the correct rate and the recall rate in the above figure, it can be found that when w=0.8, the F value is the maximum value of 0.416902133, and the recommendation effect is the best. Therefore, when calculating the similarity of the combined recommendation algorithm, the weight is 0.8. It is shown by the change curve of the weighted average F of the accuracy rate and recall rate in Figure 15. Therefore, when the weight is 0.8, it is the optimal comprehensive similarity, so the calculation of the optimal comprehensive similarity algorithm is provided by:

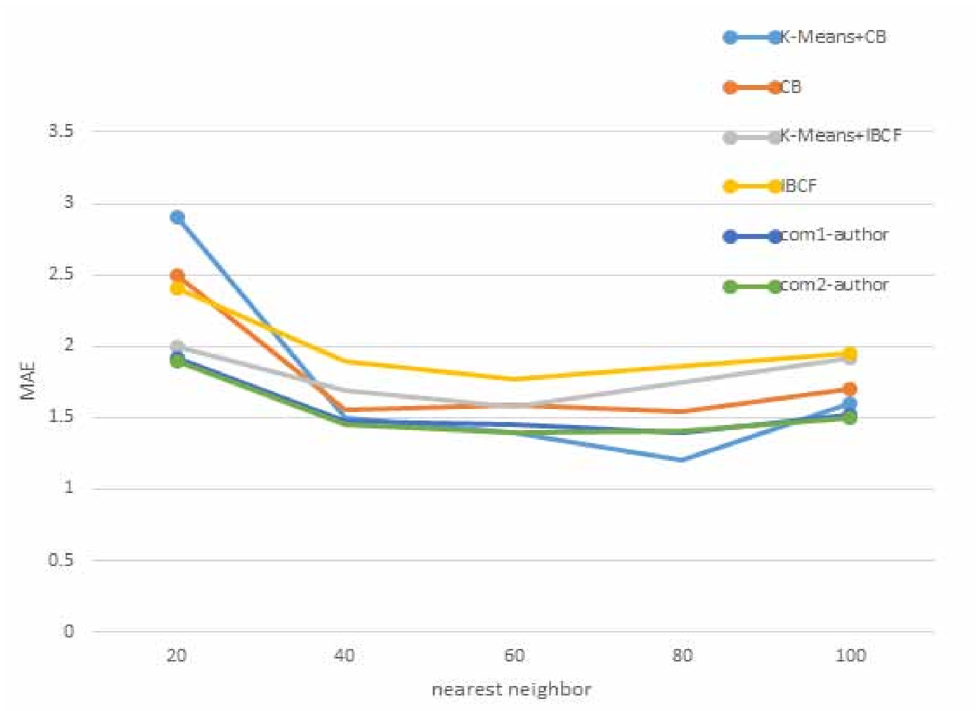
$$SimItemPlusi, j = 0.8SimItemTrai, j + 0.2SimItemAttr\text{(i,j)} \tag{17}$$

The optimal weight w=0.2 of com1-author is brought into experiment 2 to conduct the experiment.

Experiment 2. The performance comparison of the algorithm and the nearest neighbor selection experiment are set as follows.

1. Test the com1-author combination algorithm, com2-author combination algorithm, k-means+IBCF algorithm, k-means+CB algorithm, IBCF algorithm, and CB algorithm on the movie_lens 1M dataset. Select the nearest neighbor [0-50], the number of steps: 10, and the number of clusters K=10.
2. Evaluation index: mean absolute error MAE (the smaller the accuracy, the higher the accuracy).
3. Experimental purpose: To test the recommendation quality of six recommendation engines.
4. Experimental results: The variation trends of the MAE values of the six algorithms under different neighbors are as follows.

Figure 16. The change curve of Mae value for one algorithm



According to the horizontal comparison in Figure 16, it is found that the combination recommendation method of com1-author has the best recommendation quality when the nearest neighbor is selected as 30. Through analysis, this is also understandable because the CB curve generally recommends higher quality when the nearest neighbor is selected as 40. The IBCF curve has a higher recommendation efficiency when the nearest neighbor is selected as 30 because com1-author adds a weight of 0.8 to the calculation of the comprehensive acquaintance, so it is like IBCF in the problem of nearest neighbor selection. Observing the curve of com2-author, we can find that the recommendation quality is the best when the nearest neighbor is 40.

DISCUSSION

The role of the recommendation system of modern e-commerce websites is to process this information and analyze the list of products that users may be interested in. This paper conducts experiments with two schemes of the hybrid recommendation algorithm. One is to select the optimal solution from the calculation results of multiple recommendation methods and recommend it to the user by setting certain rules; the second is to combine the recommended results to optimize the recommended results. The degree of acquaintance is calculated by cosine, and the experimental results show that the mixed recommendation quality of com1-author and com2-author is the highest.

Commodity information refers to structured and unstructured drawing data, and interactive behavior refers to different behaviors of users interacting with the website, such as browsing and collecting. This paper adopts the hybrid recommendation method to make the recommendation system more accurate for the personalized recommendation formed by each user and uses the k-means clustering algorithm to divide the large user group into homogeneous user groups to alleviate problems. The user-based collaborative filtering algorithm uses the recommendation algorithm to generate recommendations within the group and update the recommendation list frequently. In this paper, the information of the product is divided into structured and unstructured information to quantify the product's information. The recommendations made this way are more precise.

CONCLUSION

This research presents a content-based, rule-based, and collaborative recommendation algorithm. The advantages and drawbacks of each are explained, and we provide experimental verification of efficacy. From the results, we draw several important conclusions. First, this research shows it is feasible to decompose the large user group into small user groups and then use the small user groups to construct a matrix. Second, two recommendation schemes are proposed on this basis and are demonstrated to improve the recommendation quality. Finally, it is a significant research contribution to finding a suitable combined recommendation algorithm by combining theory and practice and demonstrating its efficacy. Overall, the recommendation system research has two primary aspects: investigating how to optimize the recommendation algorithm and integrating recommendation algorithms and practical applications.

FUNDING AGENCY

The Open Access Processing fee for this article was covered in full by the authors of the article.

ACKNOWLEDGMENT

This work was supported by the One Belt, One Road Strategy for Anhui’s manufacturing industry, a study on the promotion of the global value chain of the Anhui manufacturing industry in 2019, based on the perspective of intra-industry specialization, Project No: SK2019ZD68.

This work was supported by the general project of the Anhui Philosophy and Social Sciences Planning Project “Research on innovation mechanism of blockchain Technology Assisting Anhui financial precision poverty alleviation,” Project No: ahsky2019d092.

REFERENCES

Arnold, F., Cardenas, I., Sörensen, K., & Dewulf, W. (2018). Simulation of B2C e-commerce distribution in Antwerp using cargo bikes and delivery points. *European Transport Research Review*, *10*(1), 1–13. doi:10.1007/s12544-017-0272-6

Bai, T., Dou, H.-J., Zhao, W. X., Yang, D.-Y., & Wen, J.-R. (2017). An experimental study of text representation methods for cross-site purchase preference prediction using the social text data. *Journal of Computer Science and Technology*, *32*(4), 828–842. doi:10.1007/s11390-017-1763-6

Bi, J. W., Liu, Y., & Fan, Z. P. (2020). A deep neural networks based recommendation algorithm using user and item basic data. *International Journal of Machine Learning and Cybernetics*, *11*(4), 763–777. doi:10.1007/s13042-019-00981-y

Biagi, F., & Falk, M. (2017). The impact of ICT and e-commerce on employment in Europe. *Journal of Policy Modeling*, *39*(1), 1–18. doi:10.1016/j.jpolmod.2016.12.004

Chen, L. F. (2018). Green certification, e-commerce, and low-carbon economy for international tourist hotels. *Environmental Science and Pollution Research International*, *26*(18), 17965–17973. doi:10.1007/s11356-018-2161-5 PMID:29785607

Cosma, A. C., & Simha, R. (2018). Machine learning method for real-time non-invasive prediction of individual thermal preference in transient conditions. *Building and Environment*, *148*, 372–383. doi:10.1016/j.buildenv.2018.11.017

Hajli, N., & Featherman, M. S. (2017). Social commerce and new development in e-commerce technologies. *International Journal of Information Management*, *37*(3), 177–178. doi:10.1016/j.ijinfomgt.2017.03.001

Jin, S., Li, H., & Li, Y. (2017). Preferences of Chinese consumers for the attributes of fresh produce portfolios in an e-commerce environment. *British Food Journal*, *119*(4), 817–829. doi:10.1108/BFJ-09-2016-0424

Jing, N., Tao Jiang, T., Du, J., & Sugumaran, V. (2018). Personalized recommendation based on customer preference mining and sentiment assessment from a Chinese e-commerce website. *Electronic Commerce Research*, *18*(1), 1–21. doi:10.1007/s10660-017-9275-6

Kant, S., & Mahara, T. (2018). Merging user and item based collaborative filtering to alleviate data sparsity. *International Journal of System Assurance Engineering & Management*, *9*(1), 1–7. doi:10.1007/s13198-016-0500-9

Li, C. Y., & Ku, Y. C. (2017). The power of a thumbs-up: Will e-commerce switch to social commerce? *Information & Management*, *55*(3), 340–357. doi:10.1016/j.im.2017.09.001

Liao, S., & Ho, C. (2021). Mobile payment and mobile application (app) behavior for online recommendations. *Journal of Organizational and End User Computing*, *33*(6), 1–26. doi:10.4018/JOEUC.20211101.oa2

Ma, X., Lei, X., Zhao, G., & Qian, X. (2018). Rating prediction by exploring user's preference and sentiment. *Multimedia Tools and Applications*, *77*(6), 6425–6444. doi:10.1007/s11042-017-4550-z

Mero, J. (2018). The effects of two-way communication and chat service usage on consumer attitudes in the e-commerce retailing sector. *Electronic Markets*, *28*(2), 1–13. doi:10.1007/s12525-017-0281-2

Min, J. K.MIN JUNG KIM. (2017). How to promote e-commerce exports to China: An empirical analysis. *KDI Journal of Economic Policy*, *39*(2), 53–74. doi:10.23895/kdijep.2017.39.2.53

Peng, H., Lan, C., Liu, Y., Liu, T., Blumenstein, M., & Li, J. (2017). Chromosome preference of disease genes and vectorization for the prediction of non-coding disease genes. *Oncotarget*, *8*(45), 78901–78916. doi:10.18632/oncotarget.20481 PMID:29108274

Robinson, C. (2017). Disclosure of personal data in ecommerce: A cross-national comparison of Estonia and the United States. *Telematics and Informatics*, *34*(2), 569–582. doi:10.1016/j.tele.2016.09.006

Töpken, S., & van de Par, S. (2019). Determination of preference-equivalent levels for fan noise and their prediction by indices based on specific loudness patterns. *The Journal of the Acoustical Society of America*, *145*(6), 3399–3409. doi:10.1121/1.5110474

Vinodhini, G., & Chandrasekaran, R. M. (2017). A sampling based sentiment mining approach for e-commerce applications. *Information Processing & Management*, *53*(1), 223–236. doi:10.1016/j.ipm.2016.08.003

Wang, C.-D., Deng, Z.-H., Lai, J.-H., & Yu, P. S. (2018). Serendipitous recommendation in e-commerce using innovator-based collaborative filtering. *IEEE Transactions on Cybernetics*, *49*(7), 2678–2692. doi:10.1109/TCYB.2018.2841924

Wu, B. (2019). A collaborative filtering recommendation algorithm for multi-source heterogeneous data. *Journal of Computer Research and Development*, *56*(5), 1034–1047.

Yang, D., & Sun, J. (2017). BM3D-Net: A convolutional neural network for transform-domain collaborative filtering. *IEEE Signal Processing Letters*, *25*(1), 55–59. doi:10.1109/LSP.2017.2768660

Yang, X., Liang, C., Zhao, M., Wang, H., Ding, H., Liu, Y., Li, Y., & Zhang, J. (2017). Collaborative filtering-based recommendation of online social voting. *IEEE Transactions on Computational Social Systems*, *24*(1), 1–13. doi:10.1109/TCSS.2017.2665122

Yuan, C., Wu, C., Wang, D., Yao, S., & Feng, Y. (2021). Review of consumer-to-consumer e-commerce research collaboration. *Journal of Organizational and End User Computing*, *33*(4), 167–184. doi:10.4018/JOEUC.20210701.oa8

Zhao, F., Yan, F., Jin, H., Yang, L. T., & Yu, C. (2017). Personalized mobile searching approach based on combining content-based filtering and collaborative filtering. *IEEE Systems Journal*, *11*(1), 324–332. doi:10.1109/JSYST.2015.2472996

Zheng, G., Yu, H., & Xu, W. (2020). collaborative filtering recommendation algorithm with item label features. *International Core Journal of Engineering*, *6*(1), 160–170.

Zinko, R., de Burgh-Woodman, H., Furner, Z. Z., & Kim, S. J. (2021). Seeing is believing: The effects of images on trust and purchase intent in eWOM for hedonic and utilitarian products. *Journal of Organizational and End User Computing*, *33*(2), 85–104. doi:10.4018/JOEUC.20210301.oa5

Wei Wang was born in Huainan, Anhui Province, China in 1980. He received a master’s degree from Hefei University of technology. Now, he works in Anhui Business and Technology College. His research interests include e-commerce and cross-border e-commerce.