

## Research Article

# Communication-Based Book Recommendation in Computational Social Systems

Long Zuo,<sup>1</sup> Shuo Xiong ,<sup>2</sup> Xin Qi,<sup>3</sup> Zheng Wen,<sup>3</sup> and Yiwen Tang<sup>2</sup>

<sup>1</sup>Chang'an University, Xi'an, China

<sup>2</sup>Huazhong University of Science and Technology, Wuhan, China

<sup>3</sup>Waseda University, Tokyo, Japan

Correspondence should be addressed to Shuo Xiong; [xiongshuo@hust.edu.cn](mailto:xiongshuo@hust.edu.cn)

Received 4 December 2020; Revised 4 January 2021; Accepted 11 January 2021; Published 30 January 2021

Academic Editor: Wei Wang

Copyright © 2021 Long Zuo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper considers current personalized recommendation approaches based on computational social systems and then discusses their advantages and application environments. The most widely used recommendation algorithm, personalized advice based on collaborative filtering, is selected as the primary research focus. Some improvements in its application performance are analyzed. First, for the calculation of user similarity, the introduction of computational social system attributes can help to determine users' neighbors more accurately. Second, computational social system strategies can be adopted to penalize popular items. Third, the network community, identity, and trust can be combined as there is a close relationship. Therefore, this paper proposes a new method that uses a computational social system, including a trust model based on community relationships, to improve the user similarity calculation accuracy to enhance personalized recommendation. Finally, the improved algorithm in this paper is tested on the online reading website dataset. The experimental results show that the enhanced collaborative filtering algorithm performs better than the traditional algorithm.

## 1. Introduction

*1.1. Research Background.* With the progress of technological development, people have come from the era of information sparsity to information overload. Many experts and scholars have proposed solutions for this, the most famous of which is the classified catalog and search engine. A classified directory's basic approach is to classify websites according to their usage scenarios and characteristics, and users can find related websites through classification. However, this solution has significant problems. With the increasing number of websites, it is undoubtedly complicated to classify them accurately. Only popular websites can be covered, and it is increasingly difficult to meet the needs of users [1].

With the current increasingly severe information overload, most of the time, users are not actively obtaining information but passively accepting it. Few people especially type a few keywords into the search box in that column. For

example, imagine that a user wanted to read a book; however, they did not know which one they wanted to choose, looking casually when they were free. In this scenario, it is difficult for a search engine to meet the needs of the user. Therefore, how to provide users with or recommend what they are interested in is a significant problem that needs to be addressed [2, 3].

For this problem, a personalized recommendation system provides a solution. In use, it does not need users to give specific keywords; it can mainly push relevant information. In practice, personalized recommendation technology has been widely used in large and medium-sized websites. In particular, for some news sites, e-commerce sites, and social media, it has become a standard [4].

Although personalized recommendation technology has been widely used in various fields, it still faces many actual application scenarios. To recommend more interesting items to users, improve the satisfaction in recommendations, achieve true personalization, and achieve "thousands of

people, thousands of needs,” a lot of research is still required [5–7]. In this case, computational social system theory would be a useful method. Computational social science refers to the academic subdisciplines concerned with computational approaches to the social sciences. This means that computers are used to model, simulate, and analyze social phenomena. Fields include computational economics, computational sociology, cliodynamics, culturomics, and the automated analysis of contents in social and traditional media. It focuses on investigating social and behavioral relationships and interactions through social simulation, modeling, network analysis, and media analysis [8, 9].

For many book e-commerce websites, compared with other articles, books have the following differences. First, the number is enormous; there are tens of thousands of books under each category, and there are many books with the same name or similar name. Second, unlike in the past, a large number of new books are put on the shelves every day. Third, unlike listening to songs and watching movies, reading itself is an extremely costly thing. Fourth, there is a strong demand for personalized book recommendations. Often, users do not know which book they want to read. In this case, book recommendation is a typical computational social system issue; how to recommend books they like and have not read before is undoubtedly very important.

Therefore, personalized recommendation algorithm research has important practical significance [10]. On the one hand, it can effectively address the problem of “book shortage” and find good books for users. On the other hand, book producers can make more relevant books become more prominent [11].

In human communication or computational social systems, personalized recommendation is mostly focused on personalized news recommendation, while book recommendation is seldom studied. News recommendation primarily considers the timeliness of news and the priority of popular information, which differs from book recommendation. Books, as a product, are relatively stable in a period. It will be affected by the time factor, but books and news have something in common in terms of recommendation: both need significant personalization, and the number of products is relatively large. Therefore, personalized book recommendation research can be used for reference and inspiration for other personalized recommendation research fields.

## 2. Related Personalized Recommendation Algorithm

**2.1. Personalized Recommendation.** Personalized recommendation generally refers to a service mode in which a website or network application collects and analyzes a user’s explicit or implicit behavior records and models their preferences. It then actively pushes information to users according to the results of the modeling. At present, no matter what business model, e-commerce, information, social networking, games, and other fields all hope to capture users’ attention through personalized recommendation [12].

Along with personalized recommendation, non-personalized recommendation is also used, such as ranking

popular items of each website and the latest updated list of items. This is only based on simple item rating data, listing time, collection, and click behavior information to achieve a summary of the item information distribution [13]. All users who visit the website will see the same information with no personalization. Personalized recommendation includes the following process as shown in Figure 1.

This process can be more clearly defined as follows: for user  $u$ , in a specific scenario  $c$ , a function is constructed, that is, the recommendation method  $f(u, i, c)$  is built to predict the user’s interest in the candidate itemset  $i$ . Then, all the candidate items are sorted according to the degree of interest, and finally, a recommendation list is generated. We can divide this process into two parts, that is, in actual practice, we have to solve two problems. The first is the problem of data and information, that is, user information, item information, scene information, what this information refers to, and how to process it. The second is the problem of algorithm selection as there are many algorithms, and it is not clear which one should be chosen. This is the core of personalized recommendation because different algorithms may produce different recommendations [14, 15]. Therefore, before evaluating any recommendation system, the first thing to evaluate is its recommendation algorithm. Whether in academia or industry, personalized recommendation systems are mostly focused on optimizing recommendation algorithms. At present, the common recommendation algorithms mainly include the following (Figure 2).

The basic idea of traditional collaborative filtering [16, 17], whether based on user collaboration or item-based collaboration [18], is to find the nearest neighbor of a user or item first and then make a score prediction or top- $N$  recommendation. The core of the algorithm is the calculation of similarity. As the user and item datasets of these algorithms increased, collaborative filtering algorithms also face related problems: data sparsity, cold start, and scalability [19].

### 2.2. Research Issue

**2.2.1. Data Sparsity.** With the expansion of the scale of websites, the number of users and items is increasing. Meanwhile, the proportion of items scored is becoming smaller and smaller, which leads to the sparseness of the user-item rating matrix. For example, the sparsity of the MovieLens dataset, Netflix dataset, NYTimes dataset, and YouTube dataset is 93.7%, 98.8%, 99.65%, and 99.72%, respectively. However, in an actual commercial recommendation system, the user’s evaluation of the recommended item is usually less than 1%. It is difficult for any two users to score the same item. When looking for the nearest neighbor or user, this may not be accurate or even be found, so the recommendation performance is not ideal. For example, user  $I$  and user  $J$  have similar interests and tastes, and user  $J$  and user  $K$  have similar hobbies and high correlation. However, if user  $I$  and user  $W$  have not rated the same product, the system will think that the correlation between the two is low, thus missing similar users.

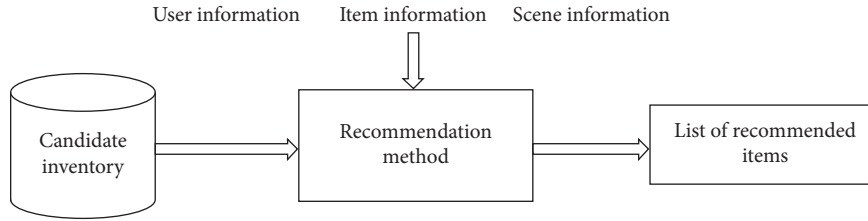


FIGURE 1: Personalized recommendation process.

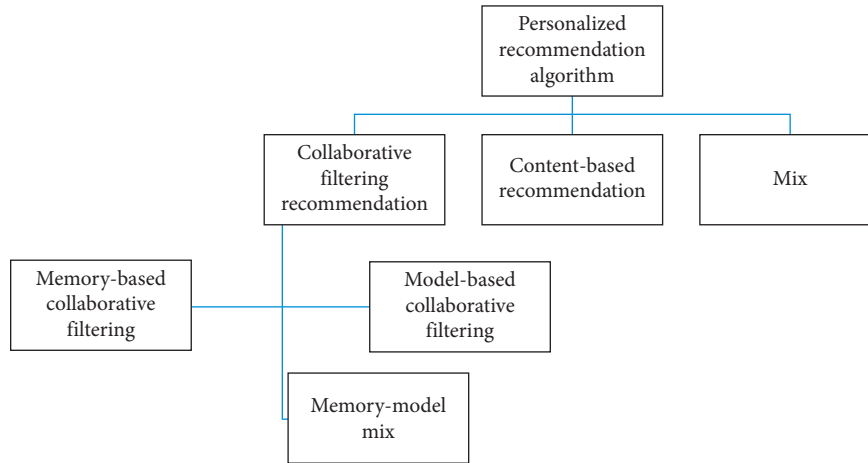


FIGURE 2: Various types of common collaborative filtering algorithms.

In this case, the amount of data is more extensive. The score is sparser, making the error in the calculation of similarity larger, which directly affects the quality of the recommendation results.

**2.2.2. Cold Start.** Cold start is also a hot topic in the field of personalized recommendation. It mainly refers to providing a personalized recommendation service for users if they do not have enough user data or item data. In the extreme case, when a new website has just opened, it is not clear how to make recommendations. This is a classic problem in personalized recommendation and generally includes three parts: user cold start, item cold start, and system cold start.

User cold start means that the system does not have historical behavior data for a new user and cannot find users with similar interests and tastes, so it is unable to make a personalized recommendation. The leading solution for the cold start problem is to recommend new items. When a new item is added to the system, it is not clear how to recommend it to the appropriate user. System cold start is the problem of how to make personalized recommendations for a new website with no users or items.

**2.2.3. Scalability.** The scalability problem mainly refers to the real-time calculation in recommendation systems. When more and more items are added to the system, the complexity of computing the similarity of users or items that

collaborative filtering relies on increases. Hence, it is not straightforward to make real-time recommendations [20].

In addition, in the calculation of user similarity, the core process of user-based collaborative filtering recommendation is to calculate the similarity between users [21], adjust the  $K$  value of  $K$  similar users, determine items with a high score for similar users, and then recommend these items [22]. The calculation for collaborative filtering based on articles is similar to that of collaborative filtering, where the similarity between items is calculated.

To some extent, user similarity determines what kind of items to recommend for the target user and is critical in the whole recommendation process. However, there are some problems in the current methods for user similarity calculation.

First, if there is only one item shared by two users, the Jaccard user similarity measure will produce inferior results. In this case, if the cosine similarity formula is used, the result is always 1. Imagine that there are two user vectors  $A$  and  $B$ , respectively,  $(1, 1, 1, 1)$  and  $(5, 5, 5, 5)$ . In this case, we find that the similarity between the two users  $A$  and  $B$  is very low. The similarity based on the Pearson correlation coefficient cannot be calculated, and the cosine value of the included angle is always 1. Neither of these two calculation methods can accurately measure the real similarity of users.

Second, the Pearson correlation coefficient measures the linear correlation of user vectors. When user vectors with high similarity are used, the opposite result may be obtained when using the Euclidean distance calculation. In addition, when

measuring some nonlinear cases, the actual performance of cosine similarity and Pearson correlation coefficient will become weaker. When the number of items is small, it is difficult to show an accurate similarity between them.

Third, the above similarity calculation methods do not consider the impact of popular items on similarity calculation. For example, suppose two users have bought the Xinhua Dictionary and scored the item. It does not mean that their interests are similar because many people have bought many popular items. In this paper, the Douban reading dataset is a popular top 250 book; they significantly impact the calculation results. In this paper, it is believed that minority items or unpopular items can better reflect the similarity of user interests.

Fourth, as the typical score as a similarity calculation index is too simple, the introduction of user community relationships, as a trust mechanism, better measures the similarity between users.

### 3. Collaborative Filtering Based on the Communication Method

#### 3.1. Communication Theory

**3.1.1. Network Community and Identity.** The term “community” evolved from Latin. It refers to a group of people who live in the same region and are more consistent in ideology and behavior. With the development of the internet, new changes have taken place in the community’s outer edge. According to Rheingold [23], a British scholar, the network community is that “when there are enough people to participate in a public discussion for a long time, put in enough emotions, and form a network composed of personal relationships in cyberspace, it will produce the social clustering phenomenon on the network.” Preece believed that, in the community generated in the network world [24], community members generally share the same interests, common behavior scope, and even values. Community members will conduct interactive and mutually beneficial behaviors, share their resources, and establish a sound community management system.

In the beginning, the community members did not have a sense of belonging and identity in the community itself. However, they gradually came into being through a period of communication and interaction and mutual understanding. Prahalad and Ramaswamy proposed that, in this interaction process [25], all network communities will have a common contract and responsibility. These make the network community members have a certain sense of identity to the network community itself. Because of the same interests and hobbies, people connect through the network and form an online community. Members of the network community can establish their identity and sense of belonging to their network community through communication, interaction, resource sharing, and other activities.

**3.1.2. Identity and Trust.** According to Prahalad, trust is a personal characteristic, and trust comes from people’s beliefs, expectations, or feelings. Generally speaking, there are

two kinds of trust among members: trust between members in the system and trust in the long term. Through a series of empirical studies, it has been proved that social identity has a significant positive impact on members’ trust. The higher the degree of identification with the community, the higher the trust in the community members.

**3.1.3. The Relationship between Trust and Similarity.** The critical problem for collaborative filtering recommendation is the problem of similarity calculation. Specifically, for users, it is the selection of users’ nearest neighbors. Therefore, we need to verify the relationship between trust and similarity, whether trust and user’s similarity preference exist.

Cai Nicolas Ziegler used mobile phone user rating data from consulting websites and the trust data between users to derive the following formula [26]:

$$\forall x \in A \frac{\sum_{y \in \text{trust}(x)}^{\text{sim}}(x, y)}{|\text{trust}(x)|} \gg \frac{\sum_{z \in A \setminus \text{trust}(x)}^{\text{sim}}(x, y)}{(|A \setminus \text{trust}(x)|)}. \quad (1)$$

In this formula,  $A$  represents all the user sets,  $\text{trust}(x)$  represents the set of users trusted by  $x$  users in the system, and  $\text{sim}(x, y)$  represents the specific size of the similarity between user  $x$  and user  $y$ , and this is a common measurement method. For instance, given that the agent is interested in Sci-Fi and AI, chances that  $y$  trusted by  $x$  also likes these two topics are much higher than for peer  $z$  not explicitly trusted by  $x$ . Various social processes are involved, such as participation in those social groups that best reflect our own interests and desires. Some recommendation and reputation systems based on trust have already been proposed, exploiting latter expected correlation between trust and interest similarity, but none have provided clear evidence that trust does correlate to profile similarity.

**3.2. Algorithm Based on the Trust Relation of the Network Community.** According to the above theoretical basis, we believe that multiple users connect through the internet based on common interests or other factors, and each member of the network community can generate an identity through good interaction and other communication behaviors. Empirical research shows that this kind of community identity has a significant positive effect on trust among members. Therefore, members of the same network community will form a trust relationship based on interests, which means a relationship created at the direction of an individual, in which one or more people hold the individual’s property subject to certain duties to use and protect it for the benefit of others. The current research results confirm that this trust relationship has a significant impact on the calculation of user similarity.

In the measurement of trust, this paper selects two indicators. First, the number of users joining the same community: the more the users join the same community, the stronger the trust relationship is and the more similar the users are. Second, the proportion of unpopular communities reduces the weight of popular communities. The more the

users from popular communities join, the more similar they are.

Based on the above ideas for improvement and the analysis of the relationship between trust and similarity, this paper proposes an improved formula for the calculation of user similarity:

$$W_{uv} = \frac{k * \sum_{i \in N(u) \cap N(v)} 1 / (1 + |N(i)|)}{\sqrt{|N(u)||N(v)|}} * \frac{|T(u) \cap T(v)|}{\sqrt{|T(u)||T(v)|}} \quad (2)$$

In the formula,  $N(u)$  is the set of items scored by user  $u$ ,  $N(v)$  is the set of items that user  $v$  has scored,  $N(i)$  is the set of items that have been scored on item  $i$ ,  $T(u)$  is the set of communities that user  $u$  joined, and  $T(v)$  is the collection of communities that user  $v$  joined;  $k$  is a constant parameter equal to  $\sqrt{2}$ . In this formula,  $1/(1 + |N(i)|)$  is used to penalize the influence of popular items in the common rating list of  $u$  and  $v$  on the calculation of similarity. For this idea, we evaluate the algorithm.

## 4. Experiment Analysis

**4.1. Data Mining and Experiment Plan.** For this research, the Douban reading website (<https://book.douban.com/>) was crawled. The top 250 books have been evaluated and scored, and all user information, including the user's unique ID, the group concerned, the site concerned, the books they want to read, the books they are reading, and the books they have read, and their scores are stored in a MongoDB database. Python's scratch framework is adopted, and Redis is used for scheduling and distributed crawling. To prevent repeated crawling, a bloom filter is used as a URL to reassemble parts. All the rating data in the dataset are the real scores of the users of the website for the books, among which there are 329443 user IDs, 203321 books, and 16144337 scores.

In order to evaluate the performance of the improved collaborative filtering algorithm for the experimental dataset, the dataset is divided into a training set and test set; the ratio is 4:1, and cross-validation is conducted. On this basis, four control experiments were designed:

- (i) Experiment 1: select the traditional collaborative filtering algorithm based on the Pearson correlation coefficient to calculate the user similarity, train the user's interest preference model on the training set, and predict the test set. The experimental indicators are accuracy, recall, and coverage.
- (ii) Experiment 2: select the traditional collaborative filtering algorithm based on the Jaccard formula to calculate the user similarity, train the model on the training set, and predict the test set. The experimental indicators are accuracy, recall, and coverage.
- (iii) Experiment 3: select the traditional collaborative filtering algorithm based on cosine similarity to calculate the user similarity, train the model on the training set, and predict the test set. The experimental indicators are accuracy, recall, and coverage.

- (iv) Experiment 4: the improved collaborative filtering algorithm is selected and trained on the training set. The experimental indexes are accuracy, recall, and coverage.

### 4.2. Algorithm Evaluation Index

**4.2.1. Accuracy.** We have function

$$\text{accuracy} = \frac{n_{\text{correct}}}{n_{\text{total}}} \quad (3)$$

where  $n_{\text{correct}}$  means the number of correct samples and  $n_{\text{total}}$  is the total number of samples. When evaluating a recommendation system, this index can reflect the extent to which it can correctly predict the target users' behavior. It is intuitive and interpretable.

**4.2.2. Precision and Recall.** The precision refers to the number of samples that we predict to be correct in our results, which are also correct in practice. Recall refers to how many of our original sample data are correctly judged. The former is mainly used to detect judgment accuracy, while the latter mostly detects incomplete judgment.

For example, imagine that there are 1000 data samples divided into boys and girls according to gender. There are 600 boys and 400 girls. We need to find all the boys, and find 500 through some algorithm but only 400 of them are boys. We can determine that the accuracy is 80%, and the recall is about 66%. The precision and recall can be defined as follows.

For user  $u$ , item  $n$  is recommended (denoted  $R(u)$ ), and assume that the set of items that the user is interested in for the test set is  $T(u)$ ; the recall and precision are

$$\begin{aligned} \text{recall} &= \frac{\sum_u |R(u) \cap T(u)|}{\sum |T(u)|}, \\ \text{precision} &= \frac{\sum_u |R(u) \cap T(u)|}{\sum_u |R(u)|}. \end{aligned} \quad (4)$$

Because this paper recommends a personalized list for users, the so-called top-N recommendation, accuracy, and recall will be used to evaluate each algorithm's actual performance. The prediction is made on the test set by training the algorithm model in the training set, and the corresponding values are calculated.

**4.2.3. Coverage.** This indicator refers to the proportion of recommended data in the total number of items. In general, it is used to assess the ability to discover small groups of items. The following formula can express the coverage of the recommendation system:

$$\text{coverage} = \frac{|\cup_{u \in U} R(u)|}{|I|} \quad (5)$$

$U$  is the set of all users,  $u$  means some of them,  $R(u)$  refers to the list of items recommended by the system, and  $I$  is the set of all items.

The number of times each item appears in the recommendation list is also essential for a more fine-grained evaluation recommendation algorithm. Generally, there are two indicators for further evaluation of coverage: Gini coefficient and information entropy.

$$G = \frac{1}{n-1} \sum_{i=1}^n (2j-n-1)p(i_j). \quad (6)$$

$i_j$  is the  $j$ -th item in the list of items sorted according to their popularity. Through this formula, it can be found that the more uneven the recommendation times are, the closer the Gini coefficient is to 1.

$$H = - \sum_{i=1}^n p(i) \log_p(i). \quad (7)$$

Here,  $p(i)$  is the popularity of item  $i$  plus all items' popularity.

**4.2.4. RMSE and MAE.** In addition to the above indicators, the root mean square error (RMSE) and mean absolute error (MAE) are also important indicators to evaluate a recommendation algorithm. Both measure the deviation between the user's real value and the algorithm's predicted value. Suppose the prediction score on the test set is  $\{y_1, y_2, \dots, y_n\}$ . The actual score of the user is  $\{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n\}$ . Then, the root mean square error is

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}}. \quad (8)$$

Also, the MAE is given in equation (9). Generally speaking, the average absolute error MAE and RMSE are used to evaluate the performance of personalized recommendation algorithms when predicting the user's rating of the items.

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n}. \quad (9)$$

**4.3. Experiment Result.** Experiment 1: using the Pearson correlation coefficient for the user similarity calculation for different  $K$  values; the value of  $K$  is  $k$  neighbors with the highest similarity to the target user interest. The results can be seen in Table 1.

Experiment 2: the performance of collaborative filtering based on the Jaccard user similarity calculation for different  $K$  values, as shown in Table 2.

Experiment 3: the performance of collaborative filtering based on the cosine similarity calculation for different  $K$  values, as shown in Table 3.

Experiment 4: the performance of improved collaborative filtering based on the trust relationship and penalization of popularity for different  $K$  values, as shown in Table 4.

From the series of experimental results, compared with the traditional collaborative filtering algorithm, the

TABLE 1: User collaborative filtering under different  $K$  values based on the Pearson correlation coefficient.

$K$	Precision (%)	Recall (%)	Coverage (%)
5	5.22	6.13	33.23
10	5.43	6.77	27.74
20	7.21	6.83	22.10
40	7.39	6.71	20.29
80	6.32	6.11	15.77

TABLE 2: User collaborative filtering under different  $K$  values based on Jaccard.

$K$	Precision (%)	Recall (%)	Coverage (%)
5	4.31	5.42	25.22
10	4.92	6.31	22.66
20	5.11	6.49	20.45
40	6.39	7.71	20.21
80	5.90	7.00	18.33

TABLE 3: User collaborative filtering under different  $K$  values based on cosine similarity.

$K$	Precision (%)	Recall (%)	Coverage (%)
5	8.22	10.22	35.23
10	8.50	15.37	27.24
20	8.66	15.55	25.33
40	8.21	14.28	22.34
80	7.31	10.33	20.31

TABLE 4: User collaborative filtering under different  $K$  values based on the trust relationship by the communication method.

$K$	Precision (%)	Recall (%)	Coverage (%)
5	10.11	20.29	39.49
10	11.12	25.30	26.62
20	12.17	26.35	22.12
40	10.70	21.12	20.30
80	10.21	21.50	20.01

proposed algorithm has significantly improved the performance of multiple evaluation indicators for multiple  $K$  values. This is due to the use of the trust relationship in the network community and penalizing popular items. For the accuracy index, compared with the Pearson correlation measure, the performance is improved by about 5% and by 6% compared with the Jaccard measure. The recall increased by 20%, 19%, and 11% compared with the Pearson, Jaccard, and cosine similarity, respectively. For the coverage index, there was an increase of 6% compared to the Pearson measure, 14% compared to the Jaccard measure, and 4% compared to the cosine similarity measure.

The following details the experimental performance of different user similarity calculation methods under multiple  $K$  values.

Specific to each experiment: from the results of experiment 1 (see Figures 3–5 for details), when using the Pearson correlation coefficient to calculate the similarity between users, the accuracy and recall rate of the  $K$  value in a particular range (5, 20) will increase with the increase of  $K$  value and reach the highest point when  $K=20$ . When the value of  $K$  exceeds 20, the performance of these two indexes in the experiment is gradually weakened. The index of coverage will increase progressively with the  $K$  value and show a downward trend progressively. This is easy to understand. The formula for calculating the coverage mentioned above explains this point well.

According to the results of experiment 2 (see Figures 6–9 for details), when the Jaccard formula is used to calculate the similarity between users, the performance of accuracy rate and recall rate is gradually improved in the range of  $K$  value (5, 40), and the performance reaches the best when  $K=40$ . Once the value of  $K$  exceeds 40, the experimental performance of these two indexes will decline.

From the process and results of experiment 3 (see Figure 7 for details), the experimental performance of the user similarity calculation method based on cosine similarity is better than the first two methods in accuracy, recall, and coverage. Based on cosine similarity, the user's interest preference vector is weighted to avoid some abnormal vectors' interference on the calculation results. When the  $K$  value is in the range of (5, 20), the accuracy rate and recall rate are gradually improved and optimal when  $K=20$ .

Figure 8 shows the performance of user similarity calculation based on cosine similarity under multiple  $K$  values.

Finally, we analyze the results of experiment 4 (see Figure 9 for details). The trust relationship and popularity penalty of the network community are used to calculate the user similarity. This improved collaborative filtering algorithm performs better in the experiment, and the effect of performance improvement is obvious. When the value of  $K$  is equal to 20, that is, the user's nearest neighbor is 20, the performance is optimal.

According to the experiment result, the contributions of this paper are as follows: (1) the traditional collaborative filtering algorithm is improved. As existing collaborative filtering algorithms have problems in computing similarity accurately, this paper introduces two indicators: trust relationship based on the community and popularity penalization. This optimizes the recommendation strategy.

(2) Using new, authentic, and more ground-based datasets. In the past, academic research on collaborative filtering algorithms was usually based on datasets such as MovieLens and Netflix. Although these datasets are accurate and reliable, they are relatively old, and there is rarely this number of dimensions. In addition, in order to fit these datasets, many models make the accuracy, recall, and coverage of the results more "beautiful," which will lead to overfitting.

In this paper, Python is used to capture more than 10 million rating data for the Douban website and more than 10

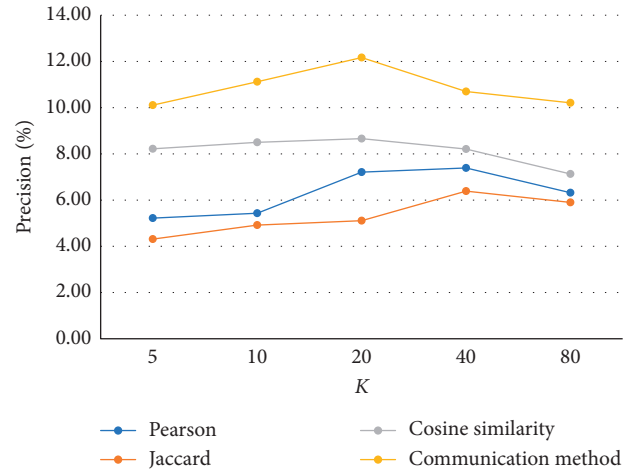


FIGURE 3: Precision of different user similarity calculation methods under multiple  $K$  values.

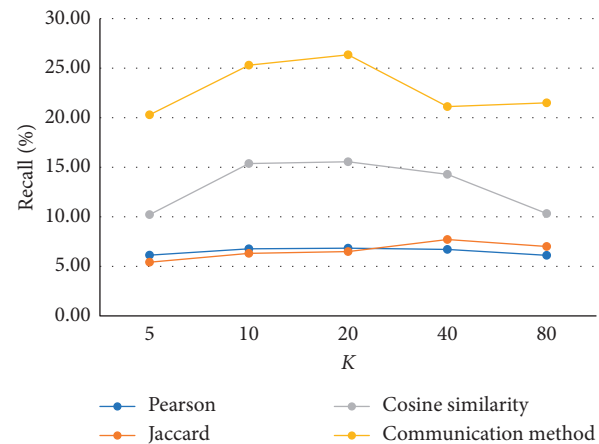


FIGURE 4: Recall of different user similarity calculation methods under multiple  $K$  values.

million users' social relationships as the verification dataset. On this basis, the requirements for the algorithm model will be higher.

There is still much work to be done on the approach proposed in this paper:

(3) The problem of data sparsity: as an essential problem for collaborative filtering recommendation algorithms, data sparsity has not been solved. The improved personalized recommendation algorithm based on collaborative filtering proposed in this paper has not made significant progress in addressing this problem.

(4) The improvement of the algorithm on the considered dataset is not large. Compared with the traditional collaborative filtering personalized algorithm, the proposed algorithm makes a certain amount of improvement in terms of accuracy, recall, and coverage. The specific range of improvement is shown and explained in the previous section. It can be seen from the data that the improvement is not

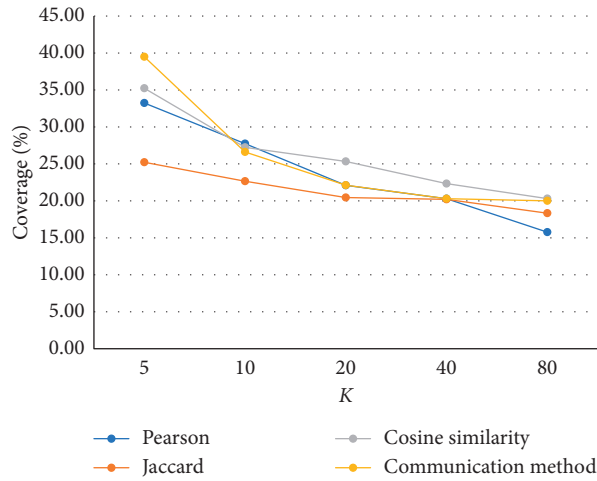


FIGURE 5: Coverage of different user similarity calculation methods under multiple  $K$  values.

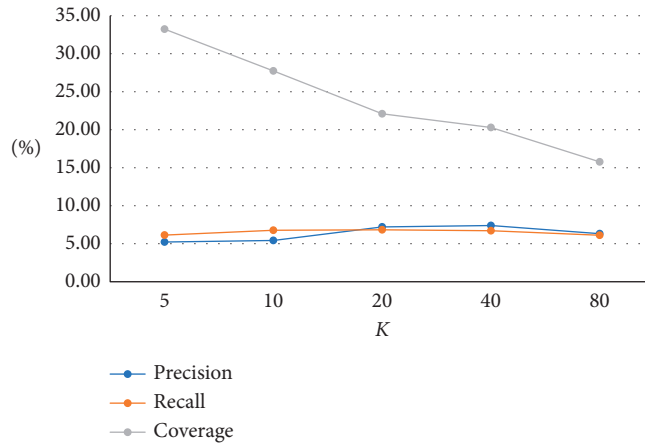


FIGURE 6: Based on the Pearson correlation coefficient, user similarity calculation is performed under multiple  $K$  values.

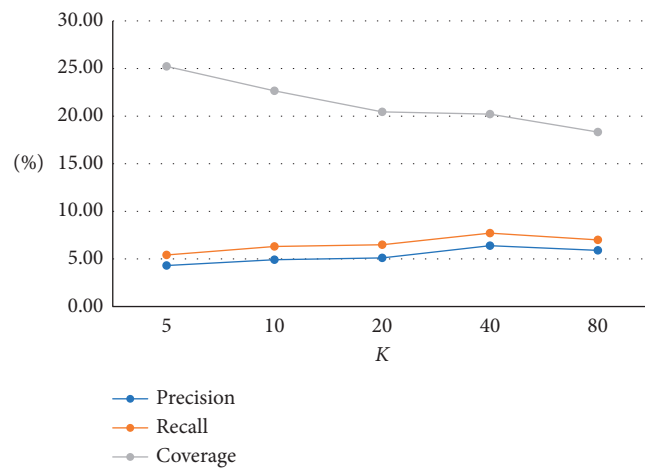


FIGURE 7: Based on the Jaccard coefficient, user similarity calculation is performed under multiple  $K$  values.



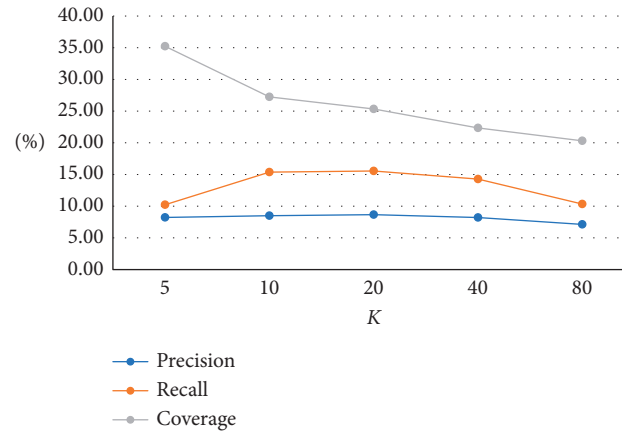


FIGURE 8: Based on cosine similarity, user similarity calculation is performed under multiple  $K$  values.

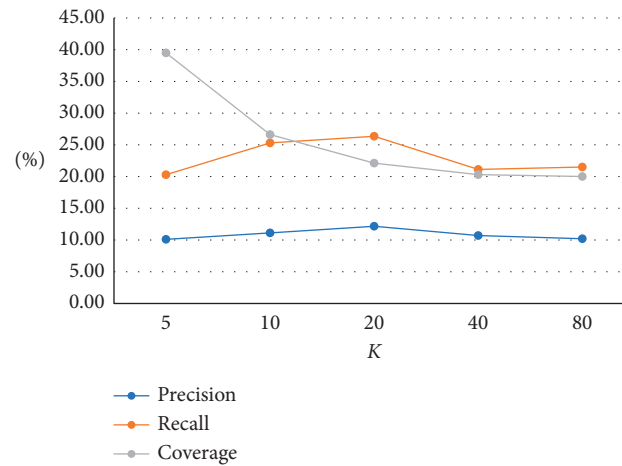


FIGURE 9: Based on the communication method, user similarity calculation is performed under multiple  $K$  values.

substantial enough. Although it has met or even exceeded the requirements, there is still a lot of room for improvement in the future.

## 5. Conclusion

With the continuous development of the internet, there is a massive amount of information coming to users. The problem of information overload is becoming increasingly prominent. With too much information, it is difficult for people to distinguish and find the information they like or deem useful. Based on the above background, it is believed that personalized recommendation approaches can effectively alleviate this problem. Focusing on the core personalized recommendation algorithm of a personalized recommendation system, this paper considers several main algorithms. It summarizes the contribution of the academic community in improving the performance of these algorithms in recent years. This paper then improves the traditional recommendation algorithm based on collaborative filtering and introduces a community-based trust relationship and popularity penalty as measures for calculating user similarity. The performance of the proposed algorithm on

the data in this study demonstrates that compared with traditional collaborative filtering, it has certain advantages in terms of accuracy, recall, coverage, and recommendation quality. Also, we would like to extend this research method from the book recommendation system to the video recommendation [27, 28].

## Data Availability

The data can be freely accessed in author's GitHub and run it. The data were from "Douban Dushu," a famous Chinese book reading comment website. Python data mining and analysis program can be run without restriction.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This paper was supported by the Huazhong University of Science and Technology Special Funds for Development of Humanities and Social Sciences (HUST, no. 450/

5003450017) and Fundamental Research Funds for the Central Universities (CHD, no. 300102240105).

## References

- [1] Y. Tian, B. Zheng, Y. Wang, Y. Zhang, and Q. Wu, "College library personalized recommendation system based on hybrid recommendation algorithm," *Procedia CIRP*, vol. 83, pp. 490–494, 2019.
- [2] S. Missaoui, F. Kassem, M. Viviani, A. Agostini, R. Faiz, and G. Pasi, "Looker: a mobile, personalized recommender system in the tourism domain based on social media user-generated content," *Personal and Ubiquitous Computing*, vol. 23, no. 2, pp. 181–197, 2019.
- [3] W. Wang, F. Xia, H. Nie et al., "Vehicle trajectory clustering based on dynamic representation learning of internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.
- [4] W. Wang, X. Zhao, Z. Gong, Z. Chen, N. Zhang, and W. Wei, "An attention-based deep learning framework for trip destination prediction of sharing bike," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.
- [5] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.
- [6] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," 2013, <https://arxiv.org/ftp/arxiv/papers/1301/1301.7363.pdf>.
- [7] W. Wang, N. Kumar, J. Chen et al., "Realizing the potential of internet of things for smart tourism with 5g and ai," *IEEE Network*, vol. 34, no. 6, pp. 295–301, 2020.
- [8] W. Wang, T. Tang, F. Xia, Z. Gong, Z. Chen, and H. Liu, "Collaborative filtering with network representation learning for citation recommendation," *IEEE Transactions on Big Data*, no. 1, p. 1, 2020.
- [9] R. M. Alvarez, *Computational Social Science*, Cambridge University Press, Cambridge, UK, 2016.
- [10] W. Wang, J. Ren, M. Alrashoud, F. Xia, M. Mao, and A. Tolba, "Early-stage reciprocity in sustainable scientific collaboration," *Journal of Informetrics*, vol. 14, no. 3, Article ID 101041, 2020.
- [11] L. Ravi, S. Vairavasundaram, S. Palani, and M. Devarajan, "Location-based personalized recommender system in the internet of cultural things," *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 5, pp. 4141–4152, 2019.
- [12] N. Anthony, A. Asem, and M. Badr Senousy, "Hybrid recommender system via personalized users' context," *Cybernetics and Information Technologies*, vol. 19, no. 1, pp. 101–115, 2019.
- [13] Z. Guo, Yu Shen, M. Aloqaily, Y. Jararweh, and K. Yu, "Probabilistic inference-based modeling for sustainable environmental systems under hybrid cloud infrastructure," *Simulation Modelling Practice and Theory*, vol. 107, Article ID 102215, 2021.
- [14] Z. Guo, Yu Shen, K. B. Ali et al., "Robust spammer detection using collaborative neural network in internet of thing applications," *IEEE Internet of Things Journal*, 2020.
- [15] Q. Lin, X. Wang, B. Hu et al., "Multiobjective personalized recommendation algorithm using extreme point guided evolutionary computation," *Complexity*, vol. 2018, Article ID 1716352, 18 pages, 2018.
- [16] B. Sarwar, K. George, K. Joseph, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, pp. 285–295, Hong Kong, China, April 2001.
- [17] L. Getoor and M. Sahami, "Using probabilistic relational models for collaborative filtering," in *Proceedings of the workshop on web usage analysis and user profiling (WEBKDD'99)*, pp. 1–6, San Diego, CA, USA, August 1999.
- [18] M. Deshpande and G. Karypis, "Item-based top- N recommendation algorithms," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 143–177, 2004.
- [19] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H.-P. Kriegel, "Probabilistic memory-based collaborative filtering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 1, pp. 56–69, 2004.
- [20] H. Lyle, D. P. Foster, Clustering methods for collaborative filtering," in *Proceedings of the AAAI workshop on recommendation systems*, pp. 114–129, Menlo Park, CA, USA, 1998.
- [21] N. Manouselis and C. Costopoulou, "Experimental analysis of design choices in multiattribute utility collaborative filtering," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 21, no. 2, pp. 311–331, 2007.
- [22] Y.-L. Chen, L.-C. Cheng, and C.-N. Chuang, "A group recommendation system with consideration of interactions among group members," *Expert Systems with Applications*, vol. 34, no. 3, pp. 2082–2090, 2008.
- [23] R. Howard, "A slice of life in my virtual community," *Global Networks: Computers and International Communication*, pp. 57–80, 1993.
- [24] J. Preece, "Sociability and usability in online communities: determining and measuring success," *Behaviour & Information Technology*, vol. 20, no. 5, pp. 347–356, 2001.
- [25] K. P. Coimbatore and V. Ramaswamy, "The co-creation connection," *Strategy and Business*, pp. 50–61, 2002.
- [26] C.-N. Ziegler and J. Golbeck, "Investigating interactions of trust and interest similarity," *Decision Support Systems*, vol. 43, no. 2, pp. 460–475, 2007.
- [27] J. Davidson, B. Liebald, J. Liu et al., "The youtube video recommendation system," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, pp. 293–296, New York, NY, USA, September 2010.
- [28] P. Cui, Z. Wang, and Su Zhou, "What videos are similar with you? learning a common attributed representation for video recommendation," in *Proceedings of the 22nd ACM International Conference on Multimedia*, pp. 597–606, Orlando, FL, USA, October 2014.