Hindawi
Journal of Electrical and Computer Engineering
Volume 2019, Article ID 7070487, 12 pages
https://doi.org/10.1155/2019/7070487

Research Article

A Collaborative Filtering Recommendation Algorithm Based on User Confidence and Time Context

Guangxia Xu,[1] **Zhijing Tang**,[1] **Chuang Ma**,[1] **Yanbing Liu**,[1] **and Mahmoud Daneshmand**[2]

[1]*Department of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[2]*Department of Business Intelligence & Analytics, Stevens Institute of Technology, Hoboken 07030, USA*

Correspondence should be addressed to Zhijing Tang; s151201007@stu.cqupt.edu.cn

Received 11 June 2018; Revised 27 February 2019; Accepted 28 March 2019; Published 17 June 2019

Academic Editor: Sos S. Agaian

Complex and diverse information is flooding entire networks because of the rapid development of mobile Internet and information technology. Under this condition, it is difficult for a person to locate and access useful information for making decisions. Therefore, the personalized recommendation system which utilizes the user's behaviour information to recommend interesting items emerged. Currently, collaborative filtering has been successfully utilized in personalized recommendation systems. However, under the condition of extremely sparse rating data, the traditional method of similarity between users is relatively simple. Moreover, it does not consider that the user's interest will change over time, which results in poor performance. In this paper, a new similarity measure method which considers user confidence and time context is proposed to preferably improve the similarity calculation between users. Finally, the experimental results demonstrate that the proposed algorithm is suitable for the sparse data and effectively improves the prediction accuracy and enhances the recommendation quality at the same time.

1. Introduction

With the exponential growth of accessible information on the network, people can acquire more and more information and have more diversified choices through the Internet, which brings great convenience to people's daily life. However, people are also lost in the ocean of information and it is hard to make the proper choice [1]. Traditional search engines can alleviate the information retrieval requirements of users to some extent. Even so, search engines can only present the same sorting results to all users and could not provide personalized services according to different users' interests. Therefore, the personalized recommendation systems came into being naturally. Specifically, according to collecting various data and analysing the preferences of the user, the personalized recommendation system learns the user's interest and behaviour patterns to recommend the required information and services for the user.

Personalized recommendation systems have been widely applied to deal with information overload problems and make personalized recommendations for user's information in e-commerce sites [2, 3]. For example, Taobao shopping, Jing Dong Mall, and the recommendation of books in Amazon [4] have provided personalized recommendation for all kinds of users. Selecting the appropriate recommendation algorithm is the core of the personalized recommendation systems [5]. Currently, popular recommendation algorithms are mainly divided into content-based recommendation, collaborative filtering (CF) recommendation, hybrid recommendation, and other algorithms. Content-based recommendation utilizes a series of discrete features of items, e.g., the genres, directors, and actors in movies, to generate recommendation [6]. CF recommendation aims to calculate a list of interesting items to target users based on the preferences of their like-minded neighbourhood. These two approaches are often combined to make hybrid recommendation [7].

As is known to us all, CF is one of the most widely used and successful technologies in personalized recommendation systems. The CF algorithm was first put forward by Goldberg et al. [8]. The core idea is that the past preference behaviours of users have a significant influence on their future behaviours, and their previous behaviours are basically consistent with future behaviours [9]. Generally speaking, the similarity between users is estimated according to the user's historical behaviour. And then, according to the evaluation of the neighbours with high similarity to the target users, the target users are predicted whether to be interested in the item.

Generally, the CF algorithm is divided into the memory-based CF recommendation and the model-based CF recommendation by Breese et al. [10]. The memory-based CF recommendation makes full use of historical data to find similar items. Taking into account from a different perspective, the memory-based CF recommendation can be divided into the item-based CF recommendation and the user-based CF recommendation. The item-based CF recommendation finds an item set which is similar to the target project based on the similarity between the items. The user-based CF recommendation relies on active user neighbourhood information to make predictions and recommendations [11]. Neighbourhood selection is one crucial procedure of user-based CF approach, which selects a set of users from candidate neighbours to comprise the neighbourhood from an active user. The model-based CF recommendation usually takes advantages of data mining, machine learning, and other techniques. Besides, the model-based CF recommendation trains a new model based on user or project characteristics and rating information and uses that model to predict the target user's possible rating of a project. In addition, the previous researches have proven that the similarity measure plays an important role in the prediction accuracy of the recommendation algorithm. There are some traditional similarity measures, such as the cosine-based similarity (COS), the Pearson correlation coefficient (PCC), the Euclidean distance-based similarity (EDS), and the adjusted cosine-based similarity (ACOS), which have been widely used in CFs to evaluate similarity [12].

In this paper, aiming at improving the traditional CF algorithms to obtain a good value of accuracy, we propose a new similarity measure method called a collaborative filtering recommendation algorithm based on user confidence and time context (UCTC_User), a new approach that improves on the basis of the ACOS. UCTC_User not only considers the influence of whether the user is more reliable than others but also takes into account the dynamics of user's interest. Experimental results reveal that UCTC_User not only well improves the similarity measure, but also it can find more accurate neighbours and enhance the prediction accuracy. The main contributions of this paper are summarized as follows:

(1) Current traditional similarity measure methods are compared with each other, and we make an improvement on the basis of the ACOS by the experimental results.

(2) Two weighting factors are included in our proposed algorithm. The first factor, user confidence, is used to emphasize that users who spend more time and energy in their fields are more convincing. The second factor, time context, takes into account the important relationship between rating time.

(3) A thorough comparison between popular algorithms in collaborative filtering approaches with our proposed algorithm shows that UCTC_User is more effective.

The remainder of the paper is organized as follows. In Section 2, the related work on similarity measures are stated. The core idea of the new algorithm is proposed in Section 3. In Section 4, we validate the effectiveness of the proposed method by experimental evaluation on the dataset. Finally, our conclusions and future work are drawn in Section 5.

2. Related Work

User-based and item-based CF methods are two of the most widely used techniques in recommender systems. However, these algorithms are widely used in both industry and academia owing to their simplicity and acceptable level of accuracy. There are still some problems to solve, such as the cold problem, the data sparsity question, and the recommendation quality condition. Therefore, there are many researchers who have studied these problems. A novel effective collaborative filtering algorithm based on user preference clustering is proposed to reduce the impact of the data sparsity [13]. And, due to the open nature of collaborative recommender systems, recommender systems cannot effectively prevent malicious users from injecting fake profile data into the ratings database; Zhang F introduced the social trust of the users into the recommender system and built the trust between them [14]. In order to solve data sparse and cold start problem, Tonglong Tang presented a collaborative filtering algorithm based on double clustering and user trust, which used user-clustering matrix to measure the user's degree of similarity [15]. And, Sajad Ahmadian proposed a method which is based on three different views of reliability measures to improve the data sparse and cold start problem [16].

Recommendation systems are becoming more and more popular with the onset of the World Wide Web and big data [17]. The essence of recommendation systems is to provide different services for different users. The techniques of CF recommendation utilize the information of an active user's neighbourhood to make predictions and recommendations. Generally, the process of CF recommendation can be separated into the following four steps.

Firstly, user-item rating matrix construction: we gather user rating information after browsing or purchasing behaviour and then clean, convert, and input the data to obtain a user-item rating matrix.

Secondly, similarity computation: among the common methods of CF recommendation techniques to measure similarity, there are COS, ACOS, EDS, and so on. After calculating the similarity between the users, we sort each user's similarity with other users.

Thirdly, neighbourhood selection: according to the result from the similarity ranking between users, the optimal k-nearest neighbours are selected to implement the predicted set, or set the similarity threshold, and users who exceed the threshold are selected as the neighbours of the target user.

Finally, rating prediction and item recommendation: after obtaining the nearest neighbour set of the target user, we use the similarity as a weight to get the target user's forecast of the unrated item and form a Top-N list to recommend to the user.

In the CF recommendation algorithm, the most important thing is to calculate the similarity between users. In order to calculate the similarity between two users from the user-item rating matrix, many user similarity measures are proposed.

The COS is one of the most popular similarity measures in CF recommendation algorithms [18]. It is the cosine of the angle between two vectors in vector space as a measure to the difference between two individuals. The user rating is regarded as a vector in the n-dimensional project space. If the user does not rate the project, the user's rating of the project is set as 0, and the similarity between users is measured by the cosine angle between the vectors.

Let $U = \{u_1, u_2, \ldots, u_m\}$ and $I = \{i_1, i_2, \ldots, i_n\}$ be the set of users and items, respectively. The user-item rating matrix is expressed as $R = [r_{u_i}]^{m*n}$. Here, m and n are the number of users and items, respectively. The r_{u_i} represents the rating made by the user of u on the item of i. And, $\overline{r}_u$ is represented the average rating of user u.

The COS equation is defined as follows:

$$\operatorname{sim}(u, v)^{\text{COS\_User}} = \frac{r_u \cdot r_v}{\left\| r_u \right\| \cdot \left\| r_v \right\|} = \frac{\sum_{i \in I_{uv}} r_{u_i} r_{v_i}}{\sqrt{\sum_{i \in I_{uv}} r_{u_i}^2} \sqrt{\sum_{i \in I_{uv}} r_{u_i}^2}}, \tag{1}$$

where I_{uv} stands for the common rated projects by user u and user v.The numerator is the inner product of the two user rating vectors, and the denominator is the product of the two user vector moduli.

However, the COS does not take into account the different of rating scale problem. The ACOS improves the above defects by subtracting the user's average score of the item [19], which is defined as follows:

$$\operatorname{sim}(u, v)^{\text{ACOS\_User}} = \frac{\sum_{i \in I_{uv}} \left(r_{u_i} - \overline{r}_u \right) \cdot \left(r_{v_i} - \overline{r}_v \right)}{\sqrt{\sum_i \left(r_{u_i} - \overline{r}_u \right)^2} \sqrt{\sum_j \left(r_{v_j} - \overline{r}_u \right)^2}}. \tag{2}$$

As is known to us all, the PCC is also a widely used similarity measure in CFs. The PCC is often used to compute a linear correlation between a pair of objects [20]. The PCC is expressed as follows:

$$\operatorname{sim}(u, v)^{\text{PCC\_User}} = \frac{\sum_{i \in I_u \cap I_v} \left(r_{u_i} - \overline{r}_u \right) \cdot \left(r_{v_i} - \overline{r}_v \right)}{\sqrt{\sum_{i \in I_u \cap I_v} \left(r_{u_i} - \overline{r}_u \right)^2} \sqrt{\sum_{i \in I_u \cap I_v} \left(r_{v_i} - \overline{r}_u \right)^2}}. \tag{3}$$

Moreover, the EDS is also frequently applied to measure the similarity [21]. The EDS is a good way to calculate similarity when data are dense and continuous, which is described as follows:

$$\operatorname{sim}(u, v)^{\text{EDS\_User}} = \frac{1}{1 + \sqrt{\sum_{i \in I_{uv}} \left(r_{u_i} - r_{v_i} \right)^2}}. \tag{4}$$

Generally speaking, almost all of the above similarity measures are very simple and do not take into account the users' other characteristics. Actually, we can consider more factors into the user similarity except for the co-rated items.

According to the above analysis, this paper proposes a new similarity measure to enhance the accuracy of measuring the similarity, and some important factors, such as the user confidence, time context of rating items, and hot items penalties, are considered.

3. Proposed Similarity Calculation Method

In collaborative filtering, the traditional way of searching neighbours for the active user depends on the rating information of common rated items by two users. However, some shortages exist in the traditional similarity calculation measure methods, i.e., the factor of user confidence is not taken into account, and the time context is also an important factor in the rating information.

In people's daily life, each person's concentration in each area is different. Because some people spend more time and energy on a certain field, these people make their words more authoritative. For example, in the "I am a singer" contest, singers as appraisers apparently spend more time than ordinary people, and these experts have their own famous works. Therefore, the value of each user is different for the target user in calculating the similarity between users, and the user has higher confidence if the user is an expert in this field. As a result, we introduce the user confidence to the adjusted cosine similarity calculation method to improve the accuracy of the recommendation.

The existing collaborative filtering recommendation algorithm treats each resource accessed by the user in the process of recommending the recommendation as equal, which is obviously unreasonable. In general, items which are recently visited by the user play a more important role in recommending resources that may be of interest to the user in the future, whereas early access records have a relatively small impact on generating recommendations. Time is a dynamical attribute, and as time goes by, the user's interest and the level of attention to the item are changing. At the same time, time brings cyclical effects, such as alternating seasons, changing weekends, and working days, and people have different requirements in response to changing times. Therefore, a user is most likely to be interested in resources, which are similar to the resources that the user has recently visited. We introduce the time context into the adjusted cosine similarity calculation method to increase the importance of the most recently accessed data in the recommendation generation process.

In order to overcome these drawbacks, this paper proposed a collaborative filtering algorithm based on user confidence and time context. Based on the adjusted cosine similarity, user confidence and time context are added to consider similar user's reference value and time context information. Figure 1 is the structure of our proposed algorithm.

3.1. Time Attenuation Function. In the process of memory, human beings create some new protruding connections between the neurons in the brain, and as memory increases, these connections will form a complex memory network. When we recall the corresponding information, the information needs to be read out, the entry point is not necessarily the information itself, and it can be other nodes on the network. For example, we will inadvertently contact memory from one thing to another. The disappearance of this association between synapses leads to oblivion, whether something is forgotten or not is related to the number of related things and the time of memory.

In psychology, researchers divide human memory into short-term memory and long-term memory. The information that people focus on for a period of time becomes short-term information, but these short-term memories can quickly be forgotten without reviewing the information within a certain period of time. People can translate short-term memories into long-term memories by consolidating reviews or recalling thoughts. If changes in user interest and memory are similar, you can also divide it into short-term interest in users and long-term interest. The user's short-term interest is transient, volatile, and sudden; long-term interest relative to short-term interest is relatively stable. In order to maintain a relatively long time, the user's short-term interest can be transformed into long-term interest.

In the nineteenth century, Hermann Ebbinghaus, a psychologist in Germany, revealed in his book "About Memory" in 1885 through experiments that human beings forgot their laws of memory. Ebbinghaus treats himself as an experimental object, through the memory of meaningless syllables. After a period of time, he recorded the number that can be recalled and made a comparison with the total amount. From the results, he found that as time went by, the speed of forgetting slowed down and the number of forgetting decreased. Ebbinghaus plotted the relationship between time and memory (the ratio of the number of syllables counted to the total) based on the experimental results, plotting the famous Ebbinghaus forgetting curve, which is as shown in Figure 2.

Through the Ebbinghaus forgetting curve, we can observe that the forgetting rate is very fast within 1 hour after learning the data, and the memory quantity is less than the initial one. After that, the forgetting rate gradually slows down and the curve becomes slow. When learning after the end of the week, the amount of information that can be recalled is about one quarter left. The user's interest changes over time, and for the memory forgetting rule of human beings, the user's interest in the object gradually diminishes as the memory is forgotten. Therefore, the recent behaviour of the user can reflect the current interest. We use the time attenuation function to represent the weight of the user's interest in the object. The longer the user takes the behaviour of the object, the less the influence on the user's current interest.

The literature [22] proposed an exponential time decay function to time-weighted user interest as follows:

$$w\left(t, t\left(R_{u_i}\right)\right) = e^{-\lambda\left(t-t\left(R_{u_i}\right)\right)}, \tag{5}$$

where λ is the weight coefficient, t is the current behaviour of the object time, and $t\,(R_{u_i})$ indicates the previous time that the user made an action on the project.

In this paper, we assume that the user's change of interest has a similar pattern with that of human memory, and then we use the time attenuation function as the coefficient of the similarity calculation formula to represent the user's interest in the object after a certain period of time, the degree of attenuation, this paper uses this function as a time attenuation function for our improved algorithm.

3.2. User Confidence. Generally speaking, the application of collaborative filtering based on neighbours is relatively common. To calculate the similarity between two users, the similarity formula is frequently utilized. Due to the sparsity of the rating matrix, the traditional methods are not reasonable to calculate the similarity between two users. For example, if there is only one common rating item for both users, it may lead to a high similarity by calculating the formula based on the current similarity methods. Obviously, such a high degree of similarity has great chance. Or if both the target user and the two users have the same degree of similarity, when both users recommend the target user as a neighbour user, the recommendation value of the two users in the current method is the same. However, in real life, the degree of recommendation adoption by target users to two close neighbours with similar degrees of similarity is different, because each person has different confidence in different people. Even if the same degree of similarity is calculated according to the commonly used similarities, the degree of confidence in them is not the same. In the current user similarity calculation formula, the user's confidence factor is not fully taken into account. Besides, the confidence between you and your friend in daily life is hardly different. Therefore, the user confidence is asymmetric.

Confidence, in statistics, refers to the degree to which a particular individual trusts the authenticity of a particular proposition, that is, a measure of the reasonableness of a person's belief in probability. The confidence of probability indicates that the event itself has no probability and the reason why the event is assigned belief evidence in the mind of a person whose probability is assigned probability. The confidence level refers to the probability that the overall parameter value falls within a certain range of the sample statistics value; and the confidence interval refers to the error range between the sample statistical value and the overall parameter value under a certain confidence level. The greater the confidence interval, the higher the confidence level.

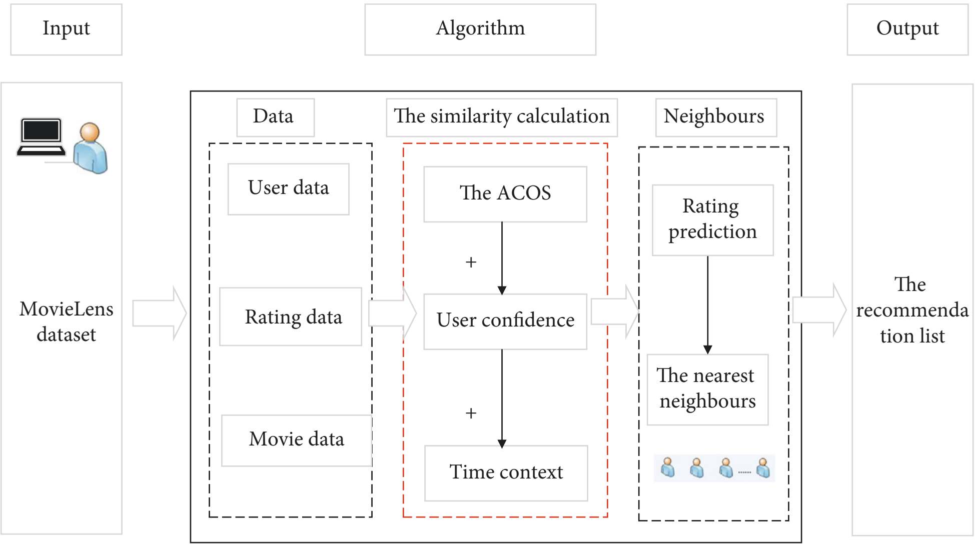


Figure 1: The structure of our proposed algorithm.

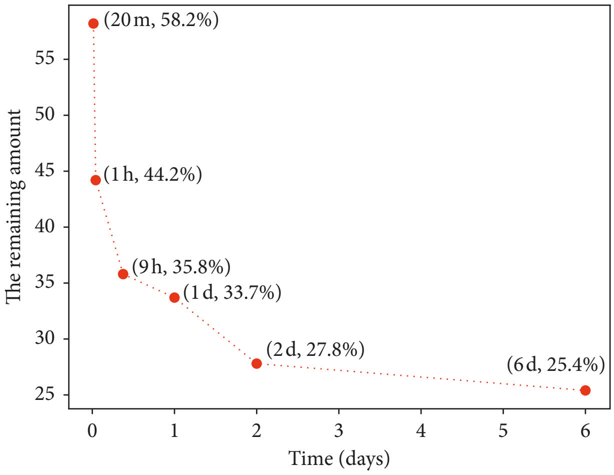


Figure 2: The Ebbinghaus forgetting curve.

Therefore, it is helpful to consider the popularity of the item in calculating the user interest similarity for improving the quality of the recommendation. Hence, we add the penalty factor of popular items into the ACOS equation. Taking books as an example, we do not think two users are similar in their interests if they have ever bought the Oxford English Dictionary, because it is a very common phenomenon to buy the Oxford English Dictionary. However, if both users bought Data Mining Introduction, then their interest might be considered more similar, as only someone who is researching data mining will buy it. Accordingly, if two users have taken the same action on the unpopular items, then the similarity of their interests can be better illustrated. We adjust the numerator of the ACOS to a new equation, and we define it as follows:

$$\sum_{i\in I_u\cap I_v}\left(\left(r_{u_i}-\bar{r}_u\right)\left(r_{v_i}-\bar{r}_v\right)+\frac{1}{\log\left(1+\left|N\left(i\right)\right|\right)}\right), \tag{6}$$

where $N(i)$ is the number of rating item i.

In people's daily life, every field has experts who spend more time and energy in their fields than the common person. As a result, these experts have more reference value for what happened in their field. From this perspective, we reach an agreement that those who spend more time and energy than others in the field are more persuasive. Therefore, we can multiply the result of the user's similarity calculation by the user confidence, and the result of this calculation is more convincing. We define the equation as follows:

$$\varphi\left(v\right)^{\text{Area\_D}}=\frac{1}{1+e^{-(\text{number}(v)-\text{ave})}}, \tag{7}$$

where number(v) is the number of rating movies by user v and ave represents the average rating items by users.

The user's interest has the characteristics of age stratification, and users of different ages have different hierarchies of interest. This paper distinguishes different user ages by dividing the user's age stage into nine levels, which is shown in Table 1.

We define following equation to calculate the age confidence between two users:

$$\varphi\left(u,v\right)^{\text{Age\_D}}=1-\frac{\left|\text{Age}_u-\text{Age}_v\right|}{8}, \tag{8}$$

where Age_u is the age segmentation of user u and Age_v is the age segmentation of user v.

According to the age confidence formula between users, if the age difference between two users is larger, the age confidence between users is smaller. Obviously, the age confidence ranges from 0 to 1. If two users belong to the same age segmentation, the age confidence between them is 1. If the difference between the age segmentation of the two users is 8, the age confidence between them is zero. The comprehensive user's confidence in the field and the age confidence, and the confidence between the users are calculated as shown in the following equation:

$$\varphi\left(u,v\right)=\varphi\left(u,v\right)^{\text{Age\_D}}+\varphi\left(u,v\right)^{\text{Area\_D}}. \tag{9}$$

Table 1: User age segmentation table.

User age	Segmentation
1–10	1
11–20	2
21–30	3
31–40	4
41–50	5
51–60	6
61–70	7
71–80	8
>=81	9

Therefore, we improve the ACOS and add the user confidence to equation (6), which is described as follows:

$$\operatorname{sim}(u,v)^{\text{UD\_User}} = \frac{\sum_{i\in I_u\cap I_v}\left(\left(r_{u_i}-\overline{r}_u\right)\left(r_{v_i}-\overline{r}_v\right)+\left(1/\left(\log_2\left(1+|N(i)|\right)\right)\right)\right)}{\sqrt{\sum_i\left(r_{u_i}-\overline{r}_u\right)^2}\sqrt{\sum_j\left(r_{v_j}-\overline{r}_u\right)^2}}\cdot\varphi(v). \tag{10}$$

3.3. Time Context. In the traditional collaborative filtering recommendation algorithms, we usually make the user's historical rating data as the basis for the results and neglect the context information of data. The context information generally includes time context, user's mood, location of the user, and so on, which are often ignored. Actually, they can indirectly express the user's interest. For instance, when people are unhappy, people may listen to some sad music. We watch different types of movies when we are together with parents or friends. In winter, we are favour of soup pot, while in summer, we enjoy watermelon. Therefore, adding time context information can improve the recommendation accuracy of personalized recommendation systems.

In the context information, time context is an important information. As a whole, the impact of time context is mainly reflected in the following aspects.

Typically, people's interests change over time. For example, young people may like to dye hair, wear personalized clothing, and as time goes by, the choice of clothing gradually transfers to the mature style. New engineers are keen to buy books with introductory classics, and after working for some time, they are in favour of the choice of instructional books with depth.

Besides, items have life cycle. A new item may attract people's attention at the beginning, and as time goes by, it may be forgotten after some time. Furthermore, different items have different lifecycles. For instance, news is often short-lived, while drama may be relatively long. And, movies tend to have shorter lifecycles and generally no more attention will be paid after a while. TV shows have a relatively longer life cycle, and as the drama plays longer, and is more likely to be talked about for some time than movies.

Furthermore, alternating seasons: in summer, people prefer more refreshing clothes such as shorts, and in winter, people prefer clothing like coats and cashmere trousers. In addition to the alternate seasons, the seasonal transformation also includes changes in the festival. For example, people prefer to go out for a holiday on the National Day because they have longer periods. People will go home for sacrificial rites during the Ching Ming Festival and people buy roses and chocolates for Valentine's Day.

In a very short time interval, users who are similar to the target user give an approximate evaluation of the same item, which demonstrates that the user is more similar to the target user, indicating that the user's rating is more valuable. Moreover, if the proportion of the intersection between both users' items is large, the high similarity between the two can be reflected to some extends. According to the perspective, we define the following equation:

$$\operatorname{sim}(u,v)^{\text{TC\_User}} = \frac{\sum_{i\in I_u\cap I_v}\left(r_{v_i}-\overline{r}_v\right)\left(r_{v_i}-\overline{r}_v\right)e^{-\mu\left|t_{u_i}-t_{v_i}\right|}}{\sqrt{N(v)N(u)}}, \tag{11}$$

where t_{u_i} and t_{v_i} are the time information of user u and v when they rated the item i, respectively. And, μ is the degree of change in interest rate, which is related to the selection of experimental subjects, and we can obtain the proper value through repeated experiments.

3.4. Proposed Method. This paper proposes a collaborative filtering recommendation algorithm based on user confidence and time context. The main idea is to calculate the similarity between users by improving the ACOS, adding user confidence and time context into it. We define the equation as follows:

$$\operatorname{sim}(u,v)^{\text{UCTC\_User}} = \alpha\cdot\operatorname{sim}(u,v)^{\text{UD\_User}} + (1-\alpha)\cdot\operatorname{sim}(u,v)^{\text{TC\_User}}, \tag{12}$$

where α is the impact factor, which can be adjusted to get the best result by repeated experiments.

In order to provide a clear description, we display our proposed method in the following steps:

Input: the rating matrix R; threshold values: μ and α.

Output: the prediction P_{u_i} of active user μ.

Step 1: use equation (12) to calculate similarity between users, and the similarity matrix generated is denoted by $\operatorname{sim}(u,v)^{\text{UCTC\_User}}$.

Step 2: sort the similarity matrix $\operatorname{sim}(u,v)^{\text{UCTC\_User}}$ in descending order.

Step 3: obtain the neighbour selection range of active user u and then generate neighbour set based on the similarity between users.

Step 4: use equation (13) to predict rating P_{u_i}.

3.5. Prediction Value. To calculate the prediction value of user u on unrated item I, we should get the neighbours of set k of the user u. According to equation (12), we can compute the similarity values between user u and other users. Then, according to the user similarity values, the neighbour set is constructed by selecting the first k users close to user u. And, we can compute the predicting value according to the following equation:

$$P(u,i)^{\text{User}} = \overline{R}_u + \frac{\sum_{v\in K}\text{sim}(u,v)\cdot\left(r_{v_i} - \overline{r}_v\right)}{\sum_{v\in K}|\text{sim}(u,v)|}, \tag{13}$$

where P_{u_i} denotes the precision of active user u on the target item i and K is the neighbour set of active user u.

4. Experiments

4.1. Dataset Collection. In this paper, in order to verify the effectiveness of the proposed algorithm, we use the MovieLens (ML) dataset collected from the GroupLens Research project team at Minnesota University. We use ML 100k which is a movie rating dataset consisting of 100,000 ratings (1–5) from 943 users on 1682 movies, and each user has rated at least 20 movies. Besides, the sparsity of extracted dataset is 92.1139%. The higher the score, the more interested the user is in the movie. We performed a 5-fold cross-validation evaluation, where 80 % of the data are retained to train the recommenders, and the rest are used for the evaluation. In our proposed algorithm, the relationship information between the user and the movie is essential for establishing the experiments. Therefore, we use the user's information, movie's information, and user's rating information about movies to realize our algorithm. Figure 3 shows the extraction scheme of user interest and preference feature that our algorithm will use.

4.2. Evaluation. To date, many researchers in the study have presented many metrics to evaluate the performance of recommendation systems. Generally speaking, the metrics for evaluating the recommendation system quality mainly include two categories: mean absolute error (MAE) and root mean square error (RMSE) [13].

MAE is one of the most widely used metrics to evaluate the recommendation accuracy and is defined as the average of absolute difference between prediction values and actual ratings. Assuming that the set of predicted user ratings is $\{p_1, p_2, p_3, \ldots, p_i, \ldots, p_n\}$ and the corresponding set of actual user ratings is $\{r_1, r_2, r_3, \ldots, r_i, \ldots, r_n\}$, the MAE is calculated as follows:

$$\text{MAE} = \frac{\sum_1^n |p_i - r_i|}{n}, \tag{14}$$

where n is the number of rating movie.

RMSE is the square root of the mean square error of the algorithm. The mean square error is the expected square of the difference between the parameter estimation and the true value of the parameter. The equation of RMSE is described as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N |p_i - r_i|^2}{N}}. \tag{15}$$

The lower the MAE and RMSE reflect, the more accurate the predictions are.

In the top-N recommendation, the precision, recall, and F1-measure are often used to evaluate the accuracy of recommendations.

$R(u)$ is the recommendation items and $T(u)$ is the actual items in the test. The precision is as follows:

$$\text{precision} = \frac{\sum_u |R(u)\cap T(u)|}{\sum_u |R(u)|}. \tag{16}$$

The recall is demonstrated as follows:

$$\text{recall} = \frac{\sum_u |R(u)\cap T(u)|}{\sum_u |T(u)|}. \tag{17}$$

The F1-measure is as follows:

$$\text{F1} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \tag{18}$$

4.3. Experimental Results and Analysis. In this section, experiments are implemented on the ML 100K dataset. We compare our proposed algorithm with other algorithms by using the metrics.

As is well known to us all, the number of selected nearest neighbours has an important impact on the quality of recommendations; therefore, the number of the nearest neighbour k varies from 10 to 60 in experiments. At present, the commonly used methods to calculate the similarity between uses include COS, ACOS, PCC, and EDS. The following figures show the results of MAE and RMSE comparing with the existing algorithms.

Figures 4 and 5 show experimental results on the ML 100K dataset. It can be noted that the values of each method decrease with the increasing of the nearest neighbour k from 10 to 20. Furthermore, the values of EDS, COS, and PCC are apparently lower and lower when the nearest neighbour k varies from 20 to 60. However, it can be seen that the values of the ACOS reduce with the nearest neighbour k change from 20 to 30 and increase when the nearest neighbour k is from 30 to 60. It is obvious that the ACOS is apparently superior to other similarities with different neighbourhood values in terms of MAE and RMSE measures. The prediction accuracy of some classic similarity measures (COS, PCC, and EDS) is comparatively lower than that of the ACOS (Figures 6–8).

Next, we compare the precision, recall, and F1-measure among the classic similarity measures.

From the above experimental results, we find it is clear that the ACOS similarity method is superior to other methods. Therefore, the similarity calculation of this paper adds user confidence and time context into the ACOS.

Based on equation (11), the factor μ can be obtained by repeating the experiments; we set the nearest neighbour k to 30, as the μ changes The experimental results are presented in Figure 9 under the conditions of different μ values.

Obviously, the results show that the MAE is the lowest when μ is 3×10^{-8}.

In addition, there is an impact factor α in equation (12), and we adjust the value of variable α to get the following result.

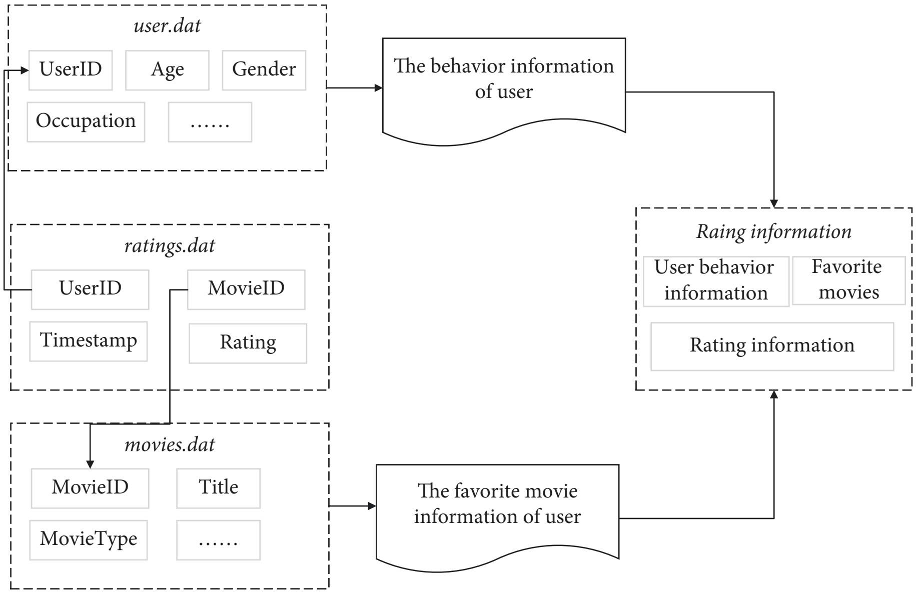


Figure 3: The extraction scheme of user interest and preference feature.

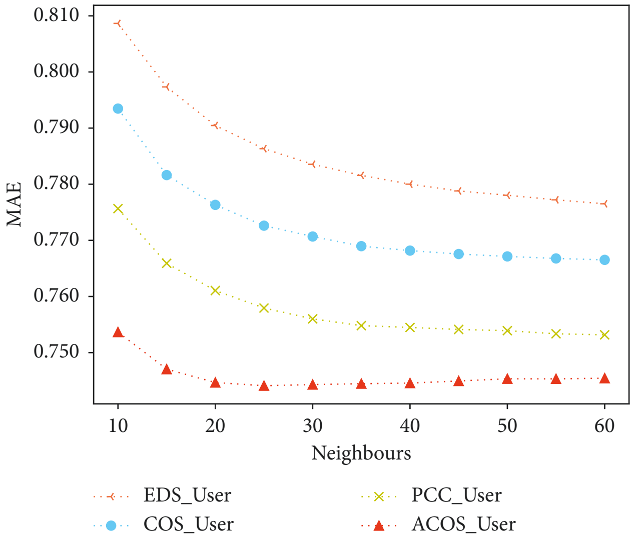


Figure 4: MAE with different neighbours on ML 100K.

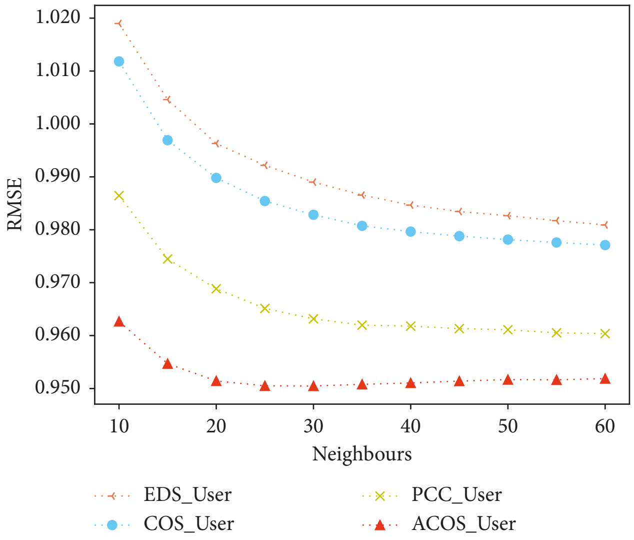


Figure 5: RMSE with different neighbours on ML 100K.

Figure 10 reveals the impact factor with different values in MAE. Figure 11 illustrates the average MAE with the variety of α in MAE. Through the experimental results, we found that when α is equal to 0.4, the best effect is obtained.

Based on the above analysis, we set α as 0.4 and μ as 3×10^{-8}. Next, we will perform the contrast experiments. Except for the ACOS, we compare the UCTC_User with PCC_WW [23] and RIT_UA [24], the results which are shown as follows.

It is obvious from Figures 12 and 13 that our scheme UCTC_User achieves almost the lowest MAE and RMSE with different neighbours on the ML 100K dataset. When the neighbour is 25, the UCTC_User can obtain the best recommendation quality. Therefore, the experimental results on the dataset show that our scheme UCTC_User has better performance than ACOS. Next, we compare the UCTC_User with the ACOS_User, PCC_WW, and RIT_UA in the precision, recall, and F1-measure (Figures 14–16).

From the experimental results in the precision, recall, and F1-measure, the UCTC_User has a great improvement on the basis of the user-based modified cosine similarity calculation method. When the number of neighbours is 30, the accuracy rate is improved by about 15.6%. The accuracy of the proposed algorithm is slightly higher than that of PCC_WW and RIT_UA. The experimental results show that the proposed algorithm has better recall rate and value than the other three algorithms in the process of increasing the number of neighbours from 10 to 30. However, when the number of neighbours increases from 40 to 60, the algorithm proposed in this paper has a lower recall rate and value than

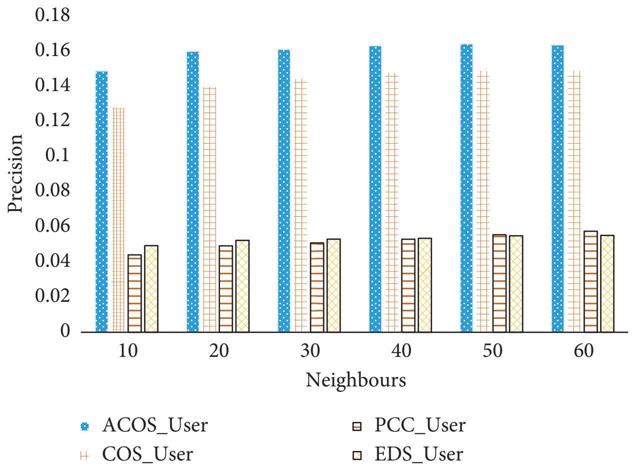


Figure 6: Precision with different neighbours on ML 100K.

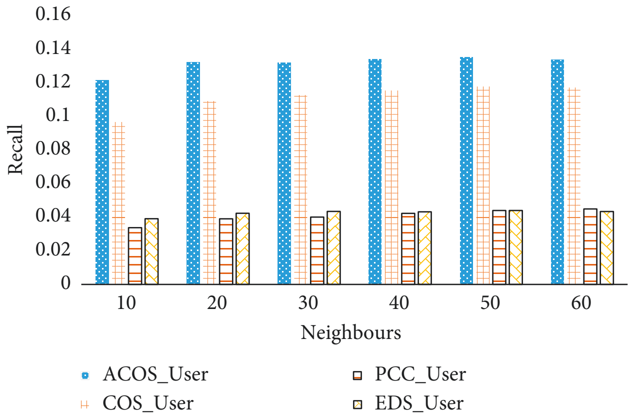


Figure 7: Recall with different neighbours on ML 100K.

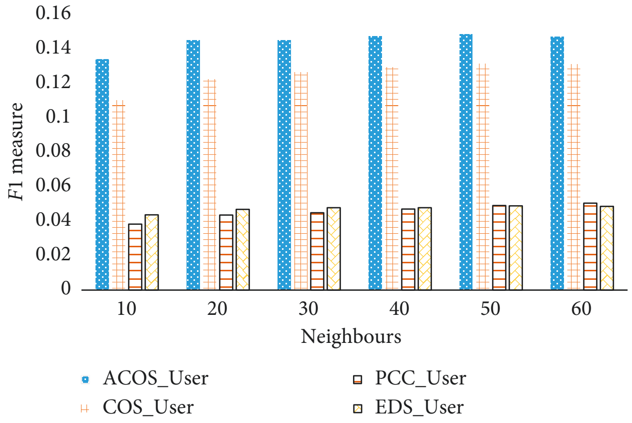


Figure 8: F1-measure with different neighbours on ML 100K.

RIT_UA, but higher than the other two algorithms. When the number of neighbours is 30, the UCTC_User has the best recommendation. Therefore, incorporating user confidence and time context based on the ACOS similarity approach is of great help in improving the accuracy, recall, and value of recommendations.

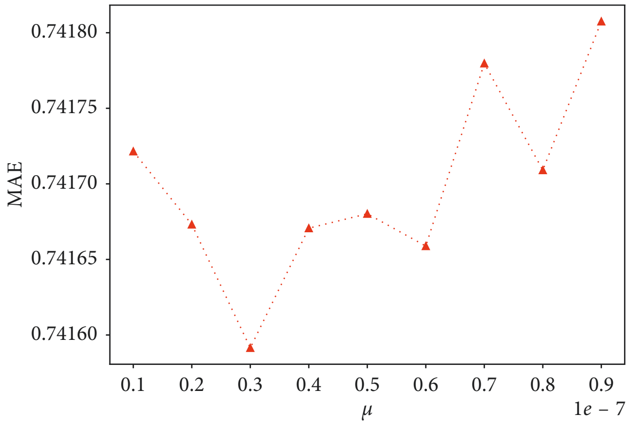


Figure 9: MAE with different μ on ML 100K.

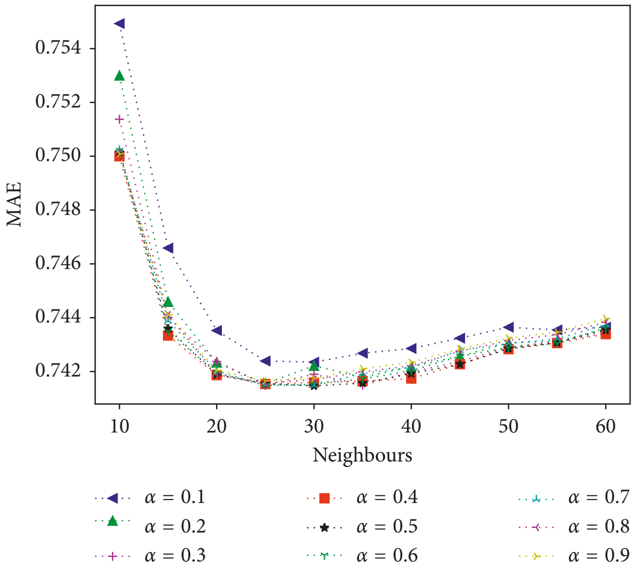


Figure 10: MAE with different α on ML 100K.

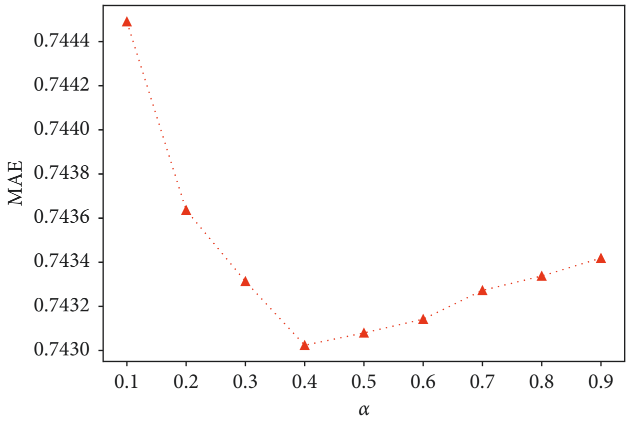


Figure 11: Average MAE with different α on ML 100K.

According to the results and detailed analysis of the above experiments, the overall analysis shows that the UCTC_User improves the quality of recommendation and

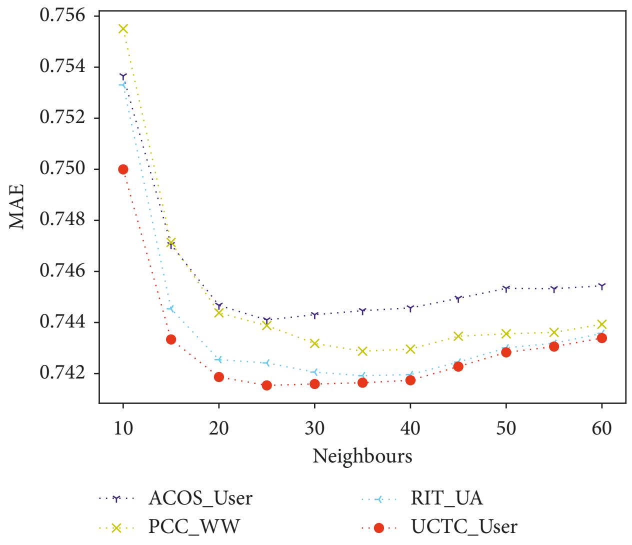


FIGURE 12: MAE with different neighbours on ML 100K.

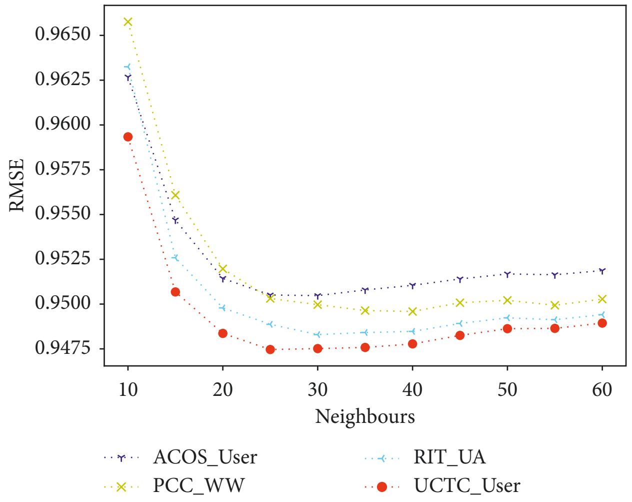


FIGURE 13: RMSE with different neighbours on ML 100K.

has better recommendation results. It verifies the correctness and feasibility of the improved algorithm proposed in this paper.

5. Conclusions

In this paper, we introduced a collaborative filtering algorithm based on user confidence and time context. Our approach is based on the perspective that experts in each field are more convincing and the interests of users change over time. For distinguishing the best typical similarity measure, the primary work of this paper is to obtain a better measure to adjust. Therefore, we improve our idea on the basis of the ACOS by comparing current popular similarity methods. As a result, we perform experiments in our paper which has proved that the UCTC_User is more effective and suitable for calculating the similarity between users. In experiments, we evaluated the effectiveness of our proposed algorithm on MAE, RMSE, precision, recall, and F1-measure. And, experimental results on the dataset demonstrated our proposed algorithm has better performance compared with current classic algorithms and some other algorithms. In conclusion, in our proposed algorithm, the UCTC_User is effective to improve the performance for recommendation systems.

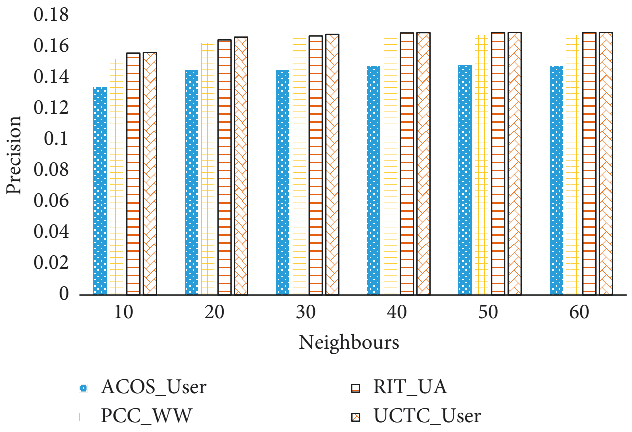


FIGURE 14: Precision with different neighbours on ML 100K.

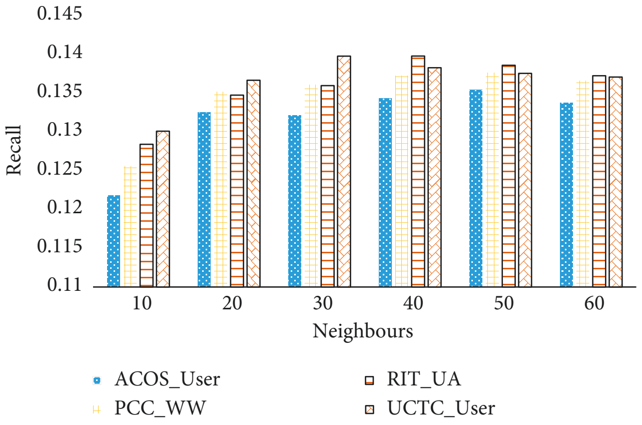


FIGURE 15: Recall with different neighbours on ML 100K.

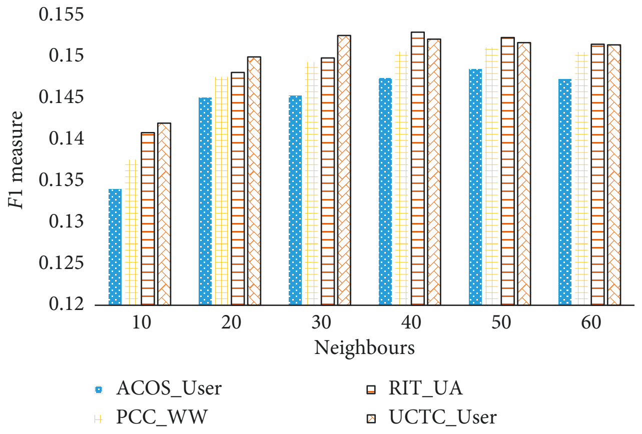


FIGURE 16: F1-measure with different neighbours on ML 100K.

In the future, we will continue to analyse the impact of the user behaviour and the information of items in recommendation systems and study the mechanism of the user rating behaviour and the impact of the item information.

Data Availability

The MovieLens 100k data used to support the findings of this study are included within the article. Everyone can download the MovieLens 100k data from the website https://grouplens.org/datasets/movielens/100k/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors acknowledge support from the National Natural Science Foundation (Grant no. 61772099); the Program for Innovation Team Building at Institutions of Higher Education in Chongqing (Grant no. CXTDG201602010); the University Outstanding Achievements Transformation Funding Project of Chongqing (Grant no. KJZH17116); the Artificial Intelligence Technology Innovation Important Subject Projects of Chongqing (cstc2017rgzn-zdyf0140); the Innovation and Entrepreneurship Demonstration Team Cultivation Plan of Chongqing (cstc2017kjrc-cxcytd0063); the China Postdoctoral Fund (Grant no. 2014M562282); the Project Postdoctoral Supported in Chongqing (Grant no. Xm2014039); the Wenfeng Leading Top Talent Project in CQUPT; the New Research Area Development Programme (Grant nos. A2015-44); the Science and Technology Research Project of Chongqing Municipal Education Committee (Grant no. KJ1400422, KJ1500441, KJ1704089, and KJ1704081); the Chongqing Research Program of Basic Research and Frontier Technology (Grant no. cstc2017jcyjAX0270 and cstc2017jcyjAX0071); the Common Key Technology Innovation of Important Industry by the Chongqing Science and Technology Commission (Grant nos. cstc2015zdcy-ztzx40001).

References

[1] K. Choi and Y. Suh, “A new similarity function for selecting neighbors for each target item in collaborative filtering,” *Knowledge-Based Systems*, vol. 37, no. 1, pp. 146–153, 2013.

[2] C. Martinez-Cruz, C. Porcel, J. Bernabé-Moreno, and E. Herrera-Viedma, “A model to represent users trust in recommender systems using ontologies and fuzzy linguistic modeling,” *Information Sciences*, vol. 311, pp. 102–118, 2015.

[3] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[4] E. Brynjolfsson, Y. Hu, and M. D. Smith, “Consumer surplus in the digital economy: estimating the value of increased product variety at online booksellers,” *Management Science*, vol. 49, no. 11, pp. 1580–1596, 2003.

[5] S. A. Mohammadi and A. Andalib, “Using the opinion leaders in social networks to improve the cold start challenge in recommender systems,” in *Proceedings of 2017 3rd International Conference on Web Research (ICWR)*, pp. 62–66, Tehran, Iran, April 2017.

[6] F. Zhang, T. Gong, V. E. Lee, G. Zhao, C. Rong, and G. Qu, “Fast algorithms to evaluate collaborative filtering recommender systems,” *Knowledge-Based Systems*, vol. 96, pp. 96–103, 2016.

[7] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” *Knowledge-Based Systems*, vol. 46, no. 1, pp. 109–132, 2013.

[8] 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.

[9] M. Jahrer and R. Legenstein, “Improved neighborhood-based algorithms for large-scale recommender systems,” in *Proceedings of KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, Las Vegas, NV, USA, August 2008.

[10] J. S. Breese, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” *New Page*, vol. 7, no. 7, pp. 43–52, 2013.

[11] J. Herlocker, J. A. Konstan, and J. Riedl, “An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms,” *Information Retrieval*, vol. 5, no. 4, pp. 287–310, 2002.

[12] C. Desrosiers and G. Karypis, “A comprehensive survey of neighborhood-based recommendation methods,” in *Recommender Systems Handbook*, pp. 107–144, Springer, Berlin, Germany, 2011.

[13] J. Zhang, Y. Lin, M. Lin, and J. Liu, “An effective collaborative filtering algorithm based on user preference clustering,” *Applied Intelligence*, vol. 45, no. 2, pp. 230–240, 2016.

[14] F. Zhang, L. Bai, and F. Gao, “A user trust-based collaborative filtering recommendation algorithm,” in *Information and Communications Security*, pp. 411–424, Springer, Berlin, Germany, 2009.

[15] T. Tang and X. Li, “A collaborative filtering algorithm based on double clustering and user trust,” in *Proceedings of International Conference on Sensor Network and Computer Engineering*, Xi’an, China, July 2016.

[16] S. Ahmadian, M. Afsharchi, and M. Meghdadi, “A novel approach based on multi-view reliability measures to alleviate data sparsity in recommender systems,” *Multimedia Tools and Applications*, vol. 78, pp. 1–36, 2019.

[17] P. Moradi and S. Ahmadian, “A reliability-based recommendation method to improve trust-aware recommender systems,” *Expert Systems with Applications*, vol. 42, no. 21, pp. 7386–7398, 2015.

[18] Y. Park, S. Park, W. Jung, and S.-G. Lee, “Reversed CF: a fast collaborative filtering algorithm using a k-nearest neighbor graph,” *Expert Systems with Applications*, vol. 42, no. 8, pp. 4022–4028, 2015.

[19] H. J. Ahn, “A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem,” *Information Sciences*, vol. 178, no. 1, pp. 37–51, 2018.

[20] Y. Wang, J. Deng, J. Gao, and P. Zhang, “A hybrid user similarity model for collaborative filtering,” *Information Sciences*, vol. 418-419, pp. 102–118, 2017.

[21] H. Sun, Y. Peng, J. Chen et al., “A new similarity measure based on adjusted euclidean distance for memory-based collaborative filtering,” *Journal of Software*, vol. 6, no. 6, pp. 993–1000, 2011.

[22] Y. Ding and L. Xue, “Time weight collaborative filtering,” in *Proceedings of the 14th ACM International Conference on*

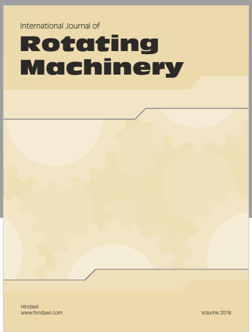
Information and Knowledge Management, pp. 485–492, Bremen, Germany, November 2005.

[23] W. Wang and J. Zheng, “Improvement of collaborative filtering algorithm based on user similarity,” *Journal of East China Normal University*, vol. 2016, no. 3, pp. 60–66, 2016.

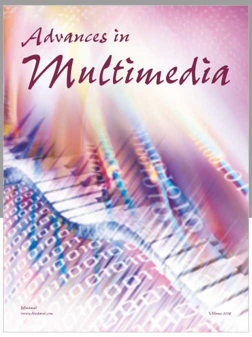
[24] X. Fan, Z. Chen, L. Zhu, Z. Liao, and B. Fu, “A novel hybrid similarity calculation model,” *Scientific Programming*, vol. 2017, Article ID 4379141, 9 pages, 2017.

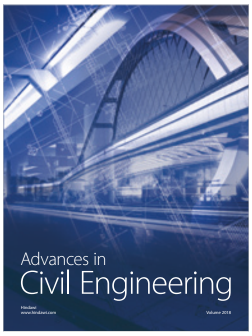
Journal of
Engineering

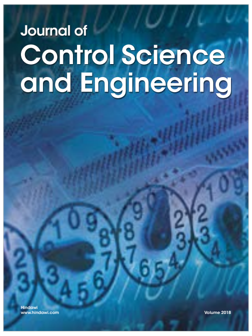
The Scientific World Journal


International Journal of
Rotating Machinery

Journal of
Sensors

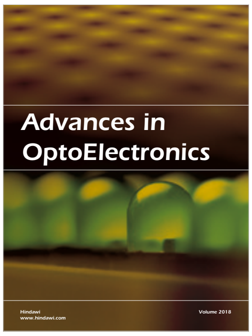

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science and Engineering

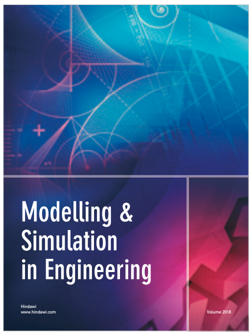
Journal of
Robotics

Journal of
Electrical and Computer Engineering

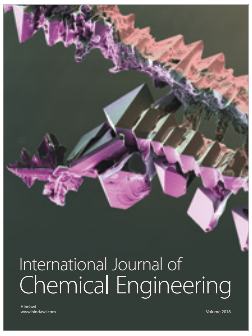

Advances in
OptoElectronics

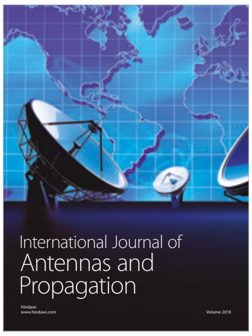
VLSI Design

International Journal of
Navigation and Observation

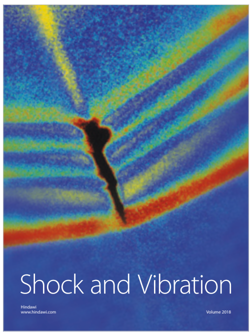

Modelling & Simulation in Engineering

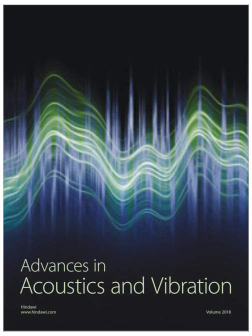
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration