IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 11, 2019, accepted December 22, 2019, date of publication January 10, 2020, date of current version January 28, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2965595*

# A New Similarity Computing Model of Collaborative Filtering

QIBING JIN, YUE ZHANG, WU CAI, AND YUMING ZHANG
College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China

Corresponding author: Yue Zhang (zhangyy_buct@163.com)

This study was funded by the National Natural Science Foundation of China (61673004) and Fundamental Research Funds for the Central Universities of China (XK1802-4).

**ABSTRACT** Collaborative filtering has become one of the most widely used methods for a variety of commercial recommendations. The key to collaborative filtering is use similarity calculation formula to find similar neighbors or projects. However, most similarity calculation methods only use the user common score and provide bad recommendations. This paper proposes a new similarity measure method, which effectively utilizes the user context information. The new method uses a singularity factor to adjust nonlinear equation and takes into account the user scoring habits. It can improve the accuracy of the prediction. The new method has been tested on the dataset and compared with other algorithms. The results show that the proposed method can improve the recommendation quality.

**INDEX TERMS** Recommender system, collaborative filtering, context information.

## I. INTRODUCTION

With the arrival of the fourth generation industrial revolution, we are in the age of data explosion, because a large amount of data requires people to choose, the recommendation system (RS) has been emerged. The goal of a shopping website is that buyers need to effectively find goals without wasting time, and sellers can accurately recommend products to them. The UBR recommendation system was describe by Balabanovi et.al in 1995 on The American Intelligent Artificial Association. Nowadays, the RS has evolved from a single, simple development to an efficient recommendation system that combines big data, cloud computeing and deep learning [1], [2]. The RS plays a very important role in many websites, such as Ali's product recommendation, video recommend in YouTube, Google search association, and so on [3]–[5]. When the RS is established, the user's feedback data is a vital factor influencing the user's recommendation. There are two types of feedback data in the system, one is implicit feedback and the other one is explicit feedback. When the user is viewing a specific item, implicit feedback is generated according to the timestamp, such as browsing, reading, click rate, etc., the explicit feedback refers to the specific rating for a product. Based on the user's rating information, the system tries to find a similar set of users or items to establish a similarity matrix, subsequently recommend a product list to the user based on the similarity calculation.

The RS discovers the user's preferences by mining the user's historical behavior data, and classifies the users based on different preferences and recommends similar products. It is well-known that collaborate filtering (CF) is one of the commonly used and successful techniques in RS [6], [7]. In literature, the recommendation system is divided into three categories, collaborate filtering algorithm (CF), Content-based algorithm(CB), and hybrid algorithm [8]. CB was mainly used in information retrieval system and information filtering system [9]. The hybrid system is the combination of content-based and collaborative filtering. The CF includes model-based algorithm and memory-based algorithm. The model-based algorithms are modeled to solve problems using machine learning [10]. The memory-based algorithm included user-based CF and item-based CF [11], [12]. In the user-based algorithm, in order to recommend projects to target user, recommendations are made by analyzing the preferences of neighboring users. For example, the order system can recommend him to restaurant according to the user's preference. Traditional collaborative filtering algorithms are faced with some problems. Therefore, this paper proposes a new similarity calculation method, which can effectively use of rating information and improve the accuracy of recommendation.

The associate editor coordinating the review of this manuscript and approving it for publication was Saqib Saeed.

Some of the collaborative filtering algorithms [13], [14] mentioned in recent years are very complicated in dealing with a small number of common scoring items for users. In addition, algorithms rely only on items that are commonly scored to form a neighborhood while ignoring the overall rating information of the user or item, thus have various limitations and poor performance in the recommendation process. The new algorithm proposed in this paper not only utilizes the user's common rating information, but also adds contextual information to correct it, which can provide users with accurate and good recommendations.

In this article, firstly, example is given to analyze the defects of the traditional similarity measure. The shortcomings will lead the user to get a bad recommendation list. Further, The SM algorithm proposes that adding the singularity factor in the process of calculating similarity can improve the recommendation accuracy. This paper improves the similarity calculation model based on the SM algorithm, the calculation formula uses the nonlinear model, which adds the singularity factor and the penalty factor of the common scoring project to form a new similarity measure method, and we call it the combine local and global (CLAG) similar collaborative filtering algorithm.

The next section of the paper is as follows: in section II, the related work of the CF algorithm is introduced. Further, the shortcomings of the traditional CF algorithm are described. In section III, a new collaborative filtering algorithm was proposed. Moreover, experiments and results are given in section IV. Finally, in section V, we analyze the experimental results and draw conclusions.

## II. LITERATURE REVIEW

In recent years, CF algorithm has been widely used in e-commerce sites to provide consumers with purchasing advice. The specific idea of user-based CF is as follows. (1) Calculate the similarity between the target user and the other users by using the rating for the item. (2) According to the result, neighbor users are obtained, and then predicted unrated items. The solution of neighbor users is the key point of the CF algorithm. Traditional CF algorithms such as COS, PCC, and ACOS are not reliable under certain circumstances. In order to solve the accuracy problems faced by the algorithm, many scholars have proposed new similarity calculation method. Cacheda et al. proposed the MSD algorithm [15], which can solve the problem of cold start of data. Then Bobadilla et al. proposed a new algorithm that combines Jaccard and MSD to provide users with a good recommendation list [16]. Furthermore, Bobadilla et al. introduced a singularity-based similarity measure(SM) [17], this method makes use of contextual information which is ignored in current systems, the algorithm divides the scores of the items into negative and positive, and weights the similarity using singularity factor. This method effectively enhances the accuracy of the RS. Moreover, Bobadilla et al. proposed a new method which called MJD (Mean−Jaccard−Difference) [18], in this paper, six influence factors are considered, and different weights are added to them. The weights are calculated by using neural networks. Meanwhile, the proposed algorithm effectively solves the cold start problem. Choi and Suh add project similarity weights when calculating user similarity [19]. They think that project similarity can correct the calculation bias. The correction based on the PCC similarity formula is as formula 1.

Ahn et al. proposes a new similarity calculation method called PIP [13], which is three characteristic factors of proximity, impact and popularity. The PIP method is deal with the cold-starting problem, which means the new user takes part in the system but there is no information about this user. The emergence of PIP similarity measure solves the problem of the traditional similarity, but it only considers the absolute value of the ratings on the first two factors. Furthermore, Liu et al. proposed the NHSM similarity method [20]. It doesn't limit to linear equations, he introduces nonlinear model to the similarity calculation method and considers the preferences of each user, meanwhile, NHSM model successfully overcomes the drawbacks. The traditional similarity calculation does not consider the difference between users. The influence of two users on each other should be different. In other words, the similarity should be asymmetric. Parivash Pirasteh et al. believe that the similarity should not be symmetrical, $sim(A, B) \neq sim(B, A)$, and adds asymmetric weights to the similarity calculation [21]. First, the number of ratings of non-common score items of users is considered. Second, by calculating the number of repetitions of each rating, the user's habits are indicated. Therefore, users with similar scoring habits will have a higher similarity. However, these algorithms still need to rely on co-rated items. In order to solve the problem of sparsity data, Patra et al. proposed a new similarity measure is called BCF [22], the proposed algorithm uses the Bhattacharyya coefficient to find similarity between users. It is not just a joint evaluation of the project, but uses the whole rating data, so this method has a more reliable recommendation. Yong *et al.* [23] also uses contextual information to solve the problem, and uses the KL distance to calculate the adjusted PSS similarity, which not only thinks about the user's personal preference but also adds the asymmetry factor. This algorithm effectively increases the reliability of the model.

In recent years, many scholars have proposed many new algorithms in order to better recommend users. In [12], the paper proposed a new similarity measure which called CjacMMD (Cosine-Jaccard-Mean Measure of Divergence), it combines Cosine, Jaccard and Mean Measure of Divergence to compute overall similarity, which suitable in sparse data. Lately, Sujoy Bag et al. propose two new models, which are RJaccard and RJMSD. Since the user has few common scores, the author explores the use of all scores and constructs a simple but effective similarity calculation model [24]. Koohi and Kiani [25] proposed a new method to find neighbor users, this method does not use variable calculation, and establishes a tree by subspace classification of all scoring data, which effectively solves the problem of data sparsity. In [26], Ren et al. applies the SVM to the collaborative filtering algorithm, this method is recommended directly to the user without predicting accurate scores and can be effectively applied to sparse data. Moreover, Liu and Wu propose a new latent factor model that converts the recommendation problem into a nearest neighbor search problem [27]. Furthermore, Polatidis et al. proposes a dynamic multi-level collaborative filtering method that can obtain high quality recommendations through positive and negative adjustments [28]. Zhang *et al.* [29] use implicit feedback information to obtain similarity, which by generalizing autoencoder paradigm model into SVD++. It can deal with data sparseness and can give good recommendations. In [30], the author attach weights to latent factor models, the weights are computed by SVD model on the sparse matrix. CBE-CF algorithm [31] uses bi-clustering method to classify the rating matrix, furthermore,calculates the information entropy to update the cluster center to find the neighbor users, this method cope with the data sparsity. In [32], the paper proposed a novel method to give good recommendations and against sparse data. The main idea is to extract the user's preference pattern from the scoring matrix and create a three-level tree, the target user at the root of the tree, the direct neighbors in the second level and the indirect neighbors in the third level, finally calculated the similarity. Sahu A K et al. apply item characters and user tags to matrix factorization, which solving data sparse problems with cross-domain recommender systems [33]. KLCF method [34]uses all user ratings to calculate similarity, and uses KL to calculate item similarity for weight adjustment, breaking the rule of using only common scoring items. Experiments show that this method can be applied to sparse matrices.

### A. TRADITIONAL COLLABORATIVE FILTERING ALGORITHM MODEL

Firstly, user-based CF algorithm calculates the similarity between the target user and other users, in the second step, the neighbor users are sorted by the similarity, and in the last step, the scores of the target items are predicted according to the scores of the neighboring users on the common items. The similarity calculation plays an important role in the collaborative filtering algorithm. In order to improve the accuracy of the calculation method, many scholars have proposed a similarity calculation method. In this section, we introduce several traditional CF algorithms. We assume $M = \{m_1, m_2, \ldots, m_k\}$ and $I = \{i_1, i_2, \ldots, i_n\}$ are the set of users and items. The user-item matrix is defined as $R = [r_{mi}]^{k\times n}$, where k and n represent the number of users and items. The cosine similarity is widely used in the collaborative filtering algorithm. The user's score is used as the vector and the cosine angle is used as the similarity value. The formula is defined as follows:

$$(COS)\, sim\,(m, n) = \frac{\sum\limits_{i\in I_m \bigcap I_n} r_{m,i} r_{n,i}}{\sqrt{\sum\limits_{i\in I_m \bigcap I_n} {r_{m,i}}^2}\sqrt{\sum\limits_{i\in I_m \bigcap I_n} {r_{n,i}}^2}} \tag{2}$$

where $I$ represent the set of all items. If user not rate item $i$, the rating $r_{m,i}$ is zero. Person Correlation Coefficient (PCC) are frequently applied to measure the similarity, the formula is defined as follows:

$$(PCC)\, sim\,(m, n) = \frac{\sum\limits_{i\in I_m \bigcap I_n} \left(r_{m,i} - \overline{r_m}\right) \sum\limits_{i\in I_m \bigcap I_n} \left(r_{n,i} - \overline{r_n}\right)}{\sqrt{\sum\limits_{i\in I_m \bigcap I_n} \left(r_{m,i} - \overline{r_m}\right)^2}\sqrt{\sum\limits_{i\in I_m \bigcap I_n} \left(r_{n,i} - \overline{r_n}\right)^2}} \tag{3}$$

where $I$ are co-rates items by user $m$ and $n$, a value of -1 in the Pearson similarity indicates that the similarity between users is low, 0 mean that there is moderate similarity between users, and $+\ 1$ mean that the similarity between users is high. PCC is widely used in collaborative filtration.

In addition, Jaccard similarity is also very common in RS, this similarity considers the problem of the co-items. The principle of this method is that item with common ratings may have similar interests.

$$(Jaccard)\, sim\,(m, n) = \frac{|I_m| \bigcap |I_n|}{|I_m| \bigcup |I_n|} \tag{4}$$

Mean squared difference (MSD) similarity considers the absolute ratings, the formula is defined as follows:

$$(MSD)\, sim\,(m, n) = 1 - \frac{\sum\limits_{i\in I} \left(r_{m,i} - r_{n,i}\right)^2}{|I|} \tag{5}$$

The scholar combining Jaccard and MSD to obtain a new similarity, which called JMSD, it is defined as follows:

$$\begin{aligned}(JMSD)\, sim\,(m, n) &= (Jaccard) sim\,(m, n) * (MSD)\, sim\,(m, n)\\ &= \frac{|I_m| \bigcap |I_n|}{|I_m| \bigcup |I_n|} * \left(1 - \frac{\sum\limits_{i\in I} \left(r_{m,i} - r_{n,i}\right)^2}{|I|}\right)\end{aligned} \tag{6}$$

$$Person^i\,(m, n) = \frac{\sum\limits_{j=1}^{k} \left\{Isim\,(i, j)^2 \times \left(R_{m,j} - \overline{R}_m\right) \times \left(R_{n,j} - \overline{R}_n\right)\right\}}{\sqrt{\sum\limits_{j=1}^{k} \left\{Isim\,(i, j)^2 \times \left(R_{m,j} - \overline{R}_m\right)\right\}^2} \times \sqrt{\sum\limits_{j=1}^{k} \left\{Isim\,(i, j)^2 \times \left(R_{n,j} - \overline{R}_n\right)\right\}^2}} \tag{1}$$

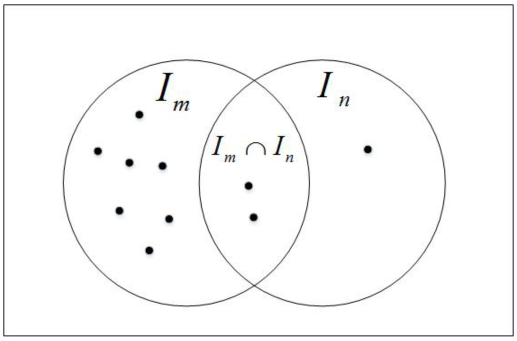


**FIGURE 1.** User rating diagram.

### B. THE PRECISION VALUE

In order to calculate the target user's score on the unrated item, firstly, we calculate the similarity between the target user and other users, and then sort the similarity values to obtain the $N$ neighbor users of the target user. We get the predicted value of the unrated item according to formula 7.

$$r_{m,t} = \overline{r_m} + \frac{\sum_{n=1}^{K} \left(r_{n,t} - \overline{r_n}\right) * sim\,(m, n)}{\sum_{n=1}^{K} sim\,(m, n)} \tag{7}$$

### C. THE DRAWBACKS IN TRADITIONAL CF

The traditional measure of similarity is simple, but there are some drawbacks that may result in relatively low prediction accuracy.

(1) The rating vectors for users are (3,4,0,1) and (2,0,5,0), there is only one common rating item between the two users. In this case, the PCC similarity cannot be calculated because the denominator is zero, and the cosine similarity calculation is most similar regardless of the rest of the score.

(2) Two users with completely different scores get high similarity values. The rating vectors for users are(1,2,0,1 )and (2,4,0,2), when the scoring vectors are multiples of each other, the use of cosine similarity to calculate similarities always yields a very high answer,it is because that they always overlap on a straight line from a geometrical perspective. This shortcoming can be corrected by ACOS.

(3) The user's common scoring project is a very important factor. Let (1,2,2,1) and (5,5,4,4) be the rating vectors of two users. Calculating only the common scoring project will result in inaccurate similarity,for example, the Jaccard similarity calculation does not take into account the user's specific rating value, so that the similarity of the two is as high as 1, If you don't consider this factor at all, you will lose some important information, such as MSD.

(4) We found that most of the similarity calculation formulas only use the user's common ratings, but the data is often sparse. The common rating is only a small department and cannot use full data when calculating. From Figure 1, the user's common rating project accounts for a different proportion of users, which will have different effects on the results.

**TABLE 1.** An example of the user-item rating matrix.

| | Item1 | Item2 | Item3 | Item4 | Item5 |
|---|---|---|---|---|---|
| User1 | 3 | **5** | **5** | 1 | 1 |
| User2 | **4** | **5** | **5** | 2 | 3 |
| User3 | **5** | **4** | 2 | **4** | **5** |
| User4 | 3 | **5** | **4** | 3 | **5** |
| User5 | 1 | **4** | **5** | 2 | **5** |
| User6 | 2 | **5** | **4** | 1 | 2 |

## III. PROPOSED SIMILARITY METHODS

### A. SINGULARITY FACTORS

The traditional similarity measure only used co-items between users, such as cosine, person correlation, MSD, Jaccard, etc. In [17], a new method is proposed that uses all contextual information which called singularity measure (SM). The main idea of the singularity measure is to calculate the similarity between two users. It should not be considered only whether the ratings of two users are similar for a certain item, the ratings of other users on this item will also affect the similarity. The value of the singularity must be modulate the value of the similarity between the two users.

The idea of SM is that if two users' ratings for the item are not similar to those of most users, it is difficult to find similarities between the two users and other users. In turn, if most users score similarly on the item, there is not much correlation between the scores of the two users and the scores of the rest of the users. In most RS systems, the scoring system (with a certain range of ratings) is used to indicate the user's preference. The level of the rating represents the user's tendency, for example, in a five-rating system (the user ratings in the range of 1, . . . 5), the rating 5 on behalf of the user is very fond of the project, that is to say, this rating is positive, and conversely, if the user A rating of 1 or 2 means that the user does not like the item and the rating is negative.

In Table 1, there are six users rated five items, assuming that they are part of the RS, rating range is 1-5, meanwhile, we want to calculate the similarity between User 2 and User 3.

For item 1, User 2 and User 3 are the only ones that score positive for the item and the remaining users are negative. In this case, we believe that the similarity between the two users is high, although ratings of the two items are inconsistent,which the item contribute a lot to the user's similarity. For item 2, the two users scored positively, the rest of the users are also positive rating, the similarity between the two users is very low, because this is a situation without singularities, the whole ratings are the same. For item 3, it shows a situation where there is a user rating that is different from the rest in which the contribution is not high of the item. If a user has a high singularity and any other user is the same for an item, the similarity between users should be very low. For item 4, user 3 ratings a positive score and user 4 is a

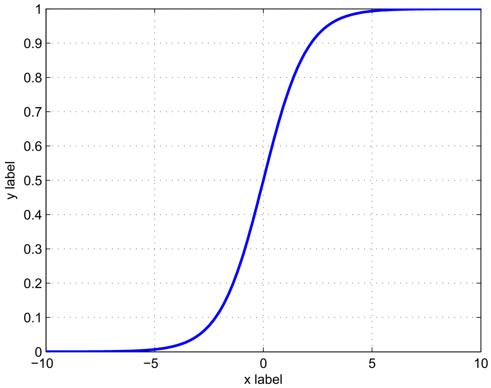


**FIGURE 2.** Sigmoid function diagram.

negative rating, regardless of the other user's rating, in which case the similarity and the singularity value are very low. The rating on item 5 represents another case where the user's rating of the item is very messy and the similarity between users should be similar.

When we calculate the similarity between two users, we additionally consider the influence of other users on them. Each time we calculated, we will add a singularity factor. The global information is added to the local common score information for correction, which can better improve the recommendation quality.

In the CF system, the key to the target user which you want to make recommendations is to find the k-neighbors. We usually need to make full use of the available information to calculate the similarity to find k-neighbors, but in reality, we can use less information (common rating information between users) than expected. Therefore, we will try to use as much global information as possible to improve the accuracy of recommendations.

### B. SIGMOID FUNCTION

Through the comparison of the second part, we find that the similarity values calculated by the model are difficult to compare with other algorithms. Therefore, we want to propose a model to solve this problem. The range of the sigmoid function is between [0, 1]. Figure 2 is sigmoid function diagram, $x > 0$, when the value of x is larger, the change of y value is smaller.

When we calculate the similarity between two users, the smaller the difference between the ratings of the two users, the higher the similarity between the users. Selecting the sigmoid function has the following advantages: a) The similarity can be in the range of [0,1], which is convenient to compare with other algorithms or with itself. b) The sigmoid function can enlarge the size of the function difference and expand the similarity value, and can also restrain the negative factors.

### C. FORMALIZATION

In this part, we give a new similarity calculation formula. The main stages of the method proposed in this paper are shown in figure 3. This method is based on the similarity measures model. A nonlinear function is introduced to calculate the similarity and the singularity factor is used to weight the similarity. The proposed algorithm not only applies the user's co-item ratings information but also takes into account the overall rating data, effectively using context information.

In [17], it has been experimentally prove that the user's scores are discrete (positive and negative ratings) can improve the accuracy of the recommendation. The main method of this paper is to continue the idea of such discrete user rating information, mainly to observe whether user ratings belong to the same category, and users of the same category have higher similarity. We define $R^P$ as a positive rating set in the RS, and define $R^N$ as a negative rating set, the positive rating is higher than the median score, and the negative rating is less than or equal to the median. For example, we assume that the range of rating is 1-5, we can define $R^P = \{4, 5\}$, $R^N = \{1, 2, 3\}$

We define $P_i$ as a collection of users who are positively rated for item $i$, $P_i = \left\{m \in M \left| r_{m,i} \in R^P \right.\right\}$ use the rating example in Table 1, $P_1 = \{2, 3\}$, $P_2 = \{1, 2, 3, 4, 5, 6\}$, $P_3 = \{1, 2, 4, 5, 6\}$, $P_4 = \{3\}$, $P_5 = \{3, 4, 5\}$

Define $N_i$ as a collection of users who are negative rated for item $i$, $N_i = \left\{m \in M \left| r_{m,i} \in R^N \right.\right\}$ use the rating example in Table 1, $N_1 = \{1, 4, 5, 6\}$, $N_2 = \varnothing$, $N_3 = \{3\}$, $N_4 = \{1, 2, 4, 5, 6\}$, $N_5 = \{1, 2, 6\}$

We define $S_P^i$ as the singularity factor for positive rating of item $i$, the more users who rating positively, the smaller the value of the $S_P^i$, use the rating example in Table 1, $S_P^1 = 1 - {}^{2}/_{6} = {}^{2}/_{3}$, $S_P^2 = 1 - {}^{6}/_{6} = 0$, $S_P^3 = 1 - {}^{5}/_{6} = {}^{1}/_{6}$, $S_P^4 = 1 - {}^{1}/_{6} = {}^{5}/_{6}$, $S_P^5 = 1 - {}^{3}/_{6} = {}^{1}/_{2}$,

$$S_P^i = 1 - \frac{|P_i|}{k} \tag{8}$$

$|P_i|$ represents the number of $P_i$, that is, the number of users who positively rating on item $i$.

We define $S_N^i$ as the singularity factor for negative rating of item $i$, the more users who rating negatively, the smaller the value of the $S_N^i$,use the rating example in Table 1, $N_P^1 = 1 - {}^{4}/_{6} = {}^{1}/_{3}$, $N_P^2 = 1 - {}^{0}/_{6} = 1$, $N_P^3 = 1 - {}^{1}/_{6} = {}^{6}/_{6}$, $N_P^1 = 1 - {}^{5}/_{6} = {}^{1}/_{6}$, $N_P^1 = 1 - {}^{3}/_{6} = {}^{1}/_{2}$

$$S_N^i = 1 - \frac{|N_i|}{k} \tag{9}$$

$|N_i|$ represents the number of $N_i$, that is, the number of users who negatively rating on item $i$.

We can divide user ratings into three cases, as shown in the second column of table 2. In order to improve the quality of prediction, the calculation of user similarity increases the singularity factor while satisfying the principle of symmetry, as shown in the fourth column of table 2.

In each process of calculating the similarity between two users, the score for the same item is added with a singularity factor, which enables more efficient use of contextual information. We define a nonlinear model with a singularity factor

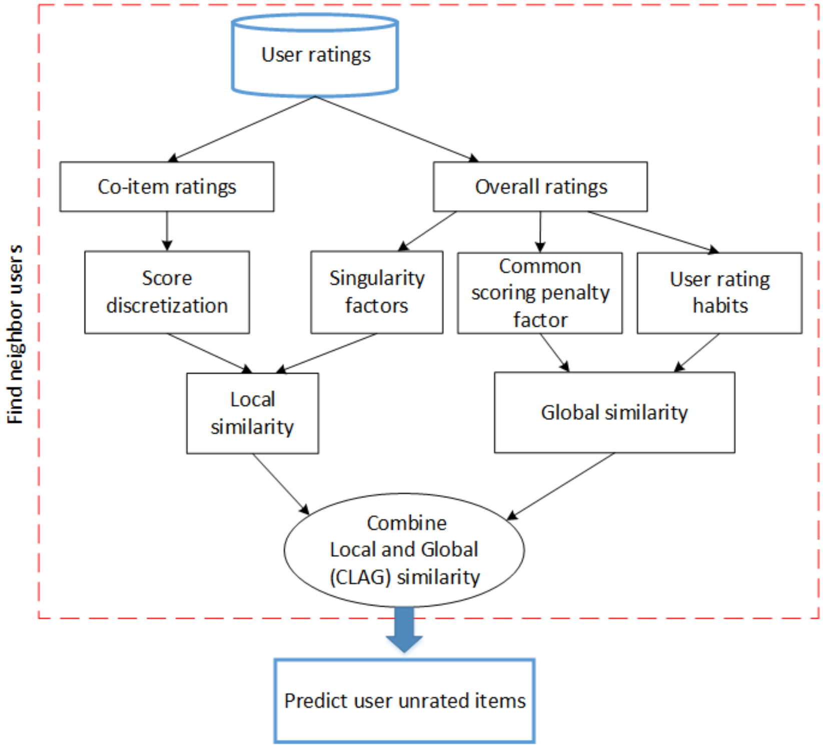


**FIGURE 3.** Stages of the proposed method.

**TABLE 2.** Possible combination of scores for users *m* and *n* a, where $p \in R^P$ and $q \in R^N$ ,set A indicates that both users score positively for an item, and set B indicates that both users score negative for an item, set C indicates that two users score an item positively and the other is a negative score.

| Case | Score combination | Distribution | Singularity factor |
|---|---|---|---|
| 1 | $(p, p)$ | $A = \{ i \in I \| r_{m,i} \in R^p \wedge r_{n,i} \in R^p \}$ | $S_P^i S_P^i$ |
| 2 | $(q, q)$ | $B = \{ i \in I \| r_{m,i} \in R^q \wedge r_{n,i} \in R^q \}$ | $S_N^i S_N^i$ |
| 3 | $(p, q), (q, p)$ | $C = \{ i \in I \| (r_{m,i} \in R^p \wedge r_{n,i} \in R^q) \vee (r_{m,i} \in R^q \wedge r_{n,i} \in R^p) \}$ | $S_P^i S_N^i$ |

weights as follows:

$$sim(m,n)^l = \frac{1}{3}\left(L(A) + L(B) + L(C)\right) \tag{10}$$

$$\begin{cases} L(A) = \dfrac{1}{|A|}\sum\limits_{\mathrm{i}\in A}\left[\left(1 - \dfrac{1}{1+e^{-|r_{m,i}-r_{n,i}|}}\right)\left(S_P^i\right)^2\right] \\ L(B) = \dfrac{1}{|B|}\sum\limits_{\mathrm{i}\in B}\left[\left(1 - \dfrac{1}{1+e^{-|r_{m,i}-r_{n,i}|}}\right)\left(S_N^i\right)^2\right] \\ L(C) = \dfrac{1}{|C|}\sum\limits_{i\in C}\left[\left(1 - \dfrac{1}{1+e^{-|r_{m,i}-\overline{r_m}||r_{n,i}-\overline{r_n}|}}\right)\left(S_P^i\right)\left(S_N^i\right)\right] \end{cases} \tag{11}$$

This part of the scoring project used to calculate the similarity is still the user's common score. So we define this formula as local similarity. The score is divided into three cases. These parts are calculated by equation 11, the sigmoid function is used in the calculation process. The first term in the equation 11 calculates the similarity between the user m and n, while both of them have items belonging to the set A. The second term in the equation 11 calculates the similarity between the user m and n, while both of them have items belonging to the set B. The first term in the equation 11 calculates the similarity between the user m and n, while both of them have items belonging to the set C.

In section II, we analyzed the user's common rating is a very important factor. In our model, we modify the formula 4, and also use the non-linear formula to solve. This part of the calculation uses all user's rating information, which we call global similarity. It is defined as follows:

$$sim(m,n)^{g1} = 1 - \mathrm{e}^{-\left(\frac{|I_m \cap I_n|}{|I_m|}\right)} \tag{12}$$

Further, we consider that users will have different preference, different users will have different scoring habits. Traditional methods cannot distinguish similarities between two users with different ratings. Apply the user's rating vector to the similarity calculation, it is defined as follows:

$$sim(m,n)^{g2} = \sum_{t=1}^{T}\sqrt{\overrightarrow{V_m(t)} \cdot \overrightarrow{V_n(t)}} \tag{13}$$

Therefore, we define the user $m$ ratings as a vector $\overrightarrow{V_m} = (f_1, f_2, \ldots f_T)$, the user $n$ ratings as a vector $\overrightarrow{V_n} = (l_1, l_2, \ldots l_T)$, $T$ indicates the user's rating value, $f_T$, $l_T$ indicate the number of users rated the score as $T$.

Finally, we combine the formulas 11, 12 and 13 to get the final similarity calculation method, which we called combine local and global(CLAG) similarity calculation method. The calculation formula is as follows:

$$sim\,(m, n) = sim^l\,(m, n) * sim^{g1}\,(m, n) * sim^{g2}\,(m, n) \tag{14}$$

A complete predictive scoring process are presented in Algorithm 1. We use the CLAG similarity calculate similarity between users and use equation 7 to calculate the score for unrated items.

### D. DISCUSSIONS ON THE NEW SIMILARITY MODEL

In the process of calculating the local similarity of two users, the singularity factor of the item is added. It can reflect the difference between the ratings of the two users and the rest of the users for an item, and can more effectively use the context information of the user, not just rely on the only common score.This paper effectively uses the user's scoring information.

When we calculate the user similarity, it is likely that the user's common score is only a small part of the target user, but we calculate the local similarity only by relying on this information, which leads to errors in the calculation. For this problem, this paper adds the common score information weight. If the common score information accounts for a large proportion of the target user score and the weight will be high.

Different users will have different rating hobbies, some like to score high ratings, someone likes to score low ratings, we convert user ratings to vectors, furthermore, calculate similarity between users, and then the similarity is to add the scores of both, which can improve more accurately.

We use the sigmoid function to calculate the similarity. This nonlinear model can be more suitable for the similarity calculation model, and the values are normalized, which can be more concise compared with other algorithms. The sigmoid function can enlarge the size of the function difference and expand the similarity value, and can also restrain the negative factors.

## IV. EXPERIMENTS

### A. DATA SET

The CF algorithm provides the user with a list of recommendations in which there are n unrated items. In order to estimate the effectiveness of the method, we use the MovieLens data set for verification. We choose the ML-100k dataset, it consists of 100,000 rating records by 943 users on 1682 movies. In the data set, each user is rated at least 20 movies, rating range is 1-5, each user can rate the movie as 1, 2, 3, 4, 5. The higher the user's rating of the movie, the more interested the user is in the movie. The sparseness of the dataset can be calculated as:$1 - 10000/\,(943 * 1682) = 0.936953$.

**Algorithm 1** Rating Prediction Using CLAG Model

```
Require:
    User-item rating metrix
Ensure:
    Value of predicted ratings
$sim\,(m, n) = [\,]\,,\, r\,(m, i) = [\,];$
for the user $m = 1$ to $U$ do
    for the user $n = 1$ to $U$ do
        for the item $i = 1$ to $I$ do
            if $r_{m,i} \geq 4$ and $r_{n,i} \geq 4$ then
                $L\,(A) = \frac{1}{|A|} \sum_{i \in A} \left[ \left( 1 - \frac{1}{1+e^{-|r_{m,i}-r_{n,i}|}} \right) \left( S_P^i \right)^2 \right]$
            else
                $L\,(A) = 0$
            end if
            if $r_{m,i} < 4$ and $r_{n,i} < 4$ then
                $L\,(B) = \frac{1}{|B|} \sum_{i \in B} \left[ \left( 1 - \frac{1}{1+e^{-|r_{m,i}-r_{n,i}|}} \right) \left( S_N^i \right)^2 \right]$
            else
                $L\,(B) = 0$
            end if
            if $(r_{m,i} < 4$ and $r_{n,i} \geq 4)$or$\,(r_{m,i} \geq 4$ and $r_{n,i} < 4)$
            then
                $L\,(C)$
                $= \frac{1}{|C|} \sum_{i \in C} \left[ \left( 1 - \frac{1}{1+e^{-|r_{m,i}-\overline{r_m}||r_{n,i}-\overline{r_n}|}} \right) \left( S_P^i \right) \left( S_N^i \right) \right]$
            else
                $L\,(C) = 0$
            end if
            $sim\,(m, n) = \sum_{t=1}^{T} \sqrt{\overrightarrow{V_m\,(t)} \cdot \overrightarrow{V_n\,(t)}} *$
            $\frac{1}{3}\,(L\,(A) + L\,(B) + L\,(C)) * \left( 1 - \mathrm{e}^{-\left( \frac{|I_m \cap I_n|}{|I_m|} \right)} \right)$
        end for
    end for
end for
for the user $m = 1$ to $U$ do
    for the item $i = 1$ to $I$ do
        if $r_{m,i} == 0$ and $sim\,(m, n) \neq 0$ then
            $r_{m,i} = \overline{r_m} + \frac{\sum_{n=1}^{K} (r_{n,i}-\overline{r_n}) * sim(m,n)}{\sum_{n=1}^{K} sim(m,n)}$
        else
            $r_{m,i} = 0$
        end if
    end for
end for
```

### B. EVALUATION METRICS

Predictive accuracy is the most discussed attribute in the recommendation system research so far. During the experiment, we select four evaluation indicators to estimate the recommended quality, there are MAE,RMSE, F-measure and NSP. We want to predict the user's rating of the target item, therefore, the accuracy of metric rating prediction is adopted.

MAE and RMSE are the most popular evaluation indicator in the rating prediction. A smaller value means that the predicted score is more accurate. It is obtained by calculating the difference between the predicted rating value and the actual rating value. The formulated as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|r_{m,i} - \overline{r_{m,i}}\right| \tag{15}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(r_{m,i} - \overline{r_{m,i}}\right)^2} \tag{16}$$

where $N$ represent the number of the items, $r_{m,i}$ devote the predicted rating value and $\overline{r_{m,i}}$ represent the actual rating value of user $m$ rate on item $i$.

In order to better assess the accuracy of the recommendations, we calculate the Recall and Precision evaluation indicators. We definite $I_{TP}$ devote the predicted recommendation items for user, $I_{TN}$ devote the actual recommendation list in the testing set. Recall and Precision are formulated as follows:

$$Recall = \frac{1}{k}\sum_{m=1}^{k}\frac{\left|I_{TP}\bigcap I_{TN}\right|}{\left|I_{TP}\right|} \tag{17}$$

$$Precision = \frac{1}{k}\sum_{m=1}^{k}\frac{\left|I_{TP}\bigcap I_{TN}\right|}{\left|I_{TN}\right|} \tag{18}$$

Recall and precision are mutually influential. Ideally, both can reach optimal values, but in general, the precision is high, the recall rate is low, the recall rate is low, and the precision is high. F-Measure is the weighted harmonic average of Precision and Recall, so F-measure is selected as the evaluation index, which can more directly compare the recommended accuracy. The larger the F-measure, the higher the recommendation accuracy, it is defined as below:

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{19}$$

In order to verify the superiority of the proposed new similarity calculation method, we compare it with some other algorithms. During the experiment, the number of neighbors has a great influence on the accuracy of the prediction score, and the number of recommendations has a great influence on the quality of the recommendation. Therefore, different evaluation indicators are compared with different variables.

### C. COMPARED METHODS

We compare the proposed algorithm with state-of-the-art algorithms described in section II.

In addition to the SM algorithm [17], we also compare it with other newly proposed algorithms which can effectively solve the data sparsity problem. BCF algorithm [22] uses bhattacharyya coefficient to calculate similarity, this method analyzes the discrete density of two user ratings and uses all the rating information to alleviate data sparsity. The NHSM [20] algorithm uses three factors to calculate the similarity and combines the user's overall scoring situation.

CjacMMD method [12] based on Mean Measure of Divergence and considered rating hobby of users, the algorithm is calculated as follows:

$$\begin{aligned} sim\,(m,n) &= sim(m,n)^{COS} + sim(m,n)^{Jaccard} + sim(m,n)^{MMD} \\ sim(m,n)^{MMD} &= \frac{1}{1+\left[\frac{1}{I}\sum_{i=1}^{I}\left(\left(\theta_m-\theta_n\right)^2-\frac{1}{\left|I_m\right|}-\frac{1}{\left|I_n\right|}\right)\right]} \end{aligned} \tag{20}$$

where $\theta_m$ and $\theta_n$ dicates the number of ratings rated by user $m$ and $n$, $\left|\overline{I_m}\right|$ and $\left|\overline{I_n}\right|$ are the set of items by user m and n.

RJMSD [24] makes full use of all the scoring information to get the relevant neighbors of the user, and the proposed similarity calculation model can make predictions easily and efficiently. The algorithm is calculated as follows:

$$\begin{aligned} sim\,(m,n) = &\left(\frac{1}{1+\frac{1}{I_m\cap I_n}+\frac{\left|\overline{I_m}\right|}{1+\left|\overline{I_m}\right|}+\frac{1}{1+\left|\overline{I_n}\right|}}\right) \\ &* \left(1-\frac{\sum_{i\in I}\left(r_{m,i}-r_{n,i}\right)^2}{|I|}\right) \end{aligned} \tag{21}$$

where $\left|\overline{I_m}\right|$ and $\left|\overline{I_n}\right|$ are the cardinality of the set of items un-co-rated by user m and n.

### D. PERFORMANCE COMPARISON

During the experiment, randomly select 80% of the rating data as the training set, and the 20% remaining data as the testing set, and the test set data is used for determination. We use two more important factors in the collaborative filtering system as variables, that is, the number of neighbors $K$ and the recommended numbers $N$, and observe different experimental results.

NSP stands for the number of prefect prediction, and the larger the NSP, the more the number of forecasts. As seen from figure 4, the CLAG proposed in this paper increases with the number of neighbors. Experimental results show that the CjacMMD algorithm has a relatively small number of prefect prediction. CLAG has the highest forecast quantity, indicating that the CLAG algorithm proposed in this paper has good prediction.

Figure 5 is a comparison of the MAE values of different algorithms for different neighbors on the Movielens 100-k dataset. From the figure 5, we can see that the algorithm we proposed is obviously better than other algorithms. The MAE values decreases with the increases of the number of neighbors K. We observe that the CLAG algorithm can reach a steady state when the number of neighbors is small. Compared with the original SM algorithm, our proposed algorithm has been significantly improved. From the figure we can observe that the BCF, SM, NHSM, AC-COS proposed in recent years are better than the traditional algorithm,

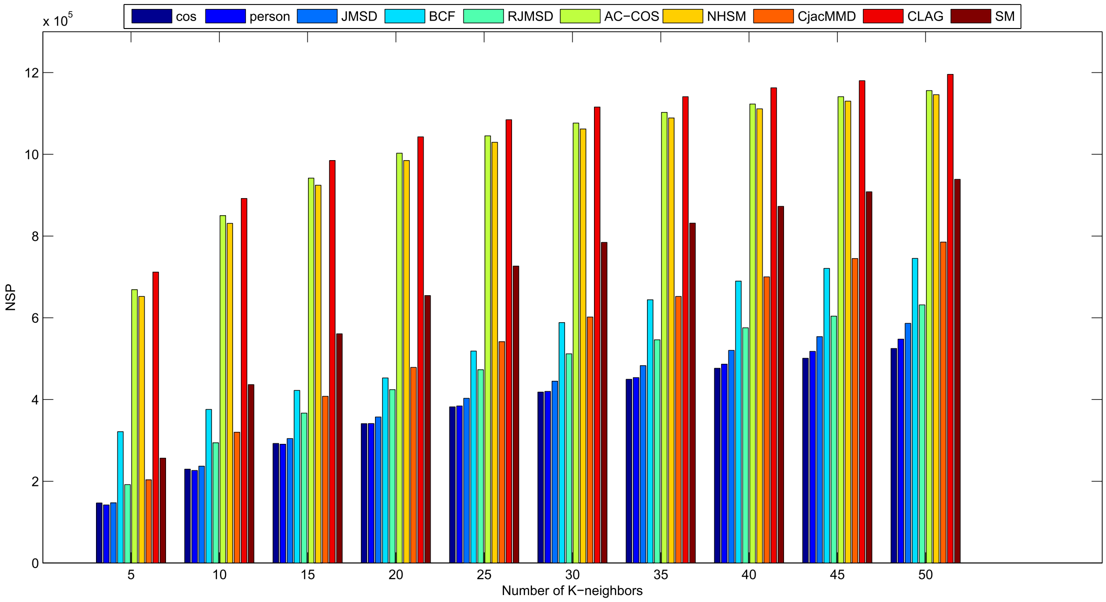


**FIGURE 4.** Comparison of the NSP values of different algorithms for different neighbors on the Movielens 100-k dataset.

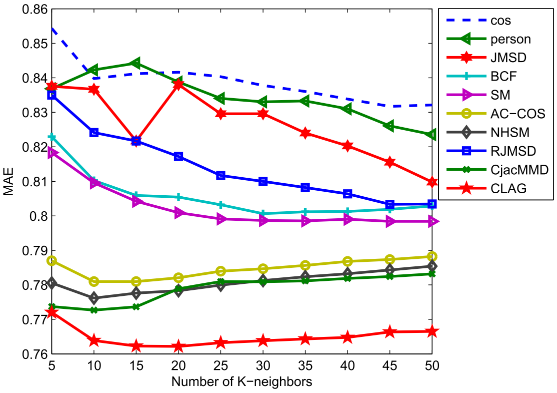


**FIGURE 5.** Comparison of the MAE values of different algorithms for different neighbors on the Movielens 100-k dataset.

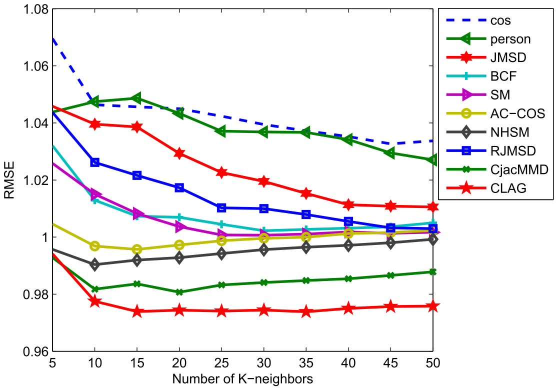


**FIGURE 6.** Comparison of the RMSE values of different algorithms for different neighbors on the Movielens 100-k dataset.

RJMSD has bad MAE values, in contrast, the MAE value also increases with the number of neighbors, CjacMMD algorithm is slightly worse than CLAG algorithm.

In figure 6, when K is more than 10, the accuracy of the improved similarity measure(CLAG) gives the best results and is very stable. we can observe that CLAG obtains the best RMSE values than other algorithms in the whole range. Compared with the SM algorithm, the stability is still reduced by 2%. SM is better than state-of-art algorithm, but it worse than NHSM, AC-COS and CjacMMD. When K is more than 15, the accuracy of the similarity between NHSM and AC-COS is deteriorated.

In figure 7, The accuracy of various similarity measures has not changed significantly in the entire Top-N range, and it is steadily increasing. We notice that the NHSM, AC-COS, CjacMMD and CLAG can surpass other methods. when the Top-N is set 20, our improved method(CLAG) has remarkable improvement. However, CLAG is worse than CjacMMD when the TOP-N is less than 20. When the recommended number more than 15, as the number

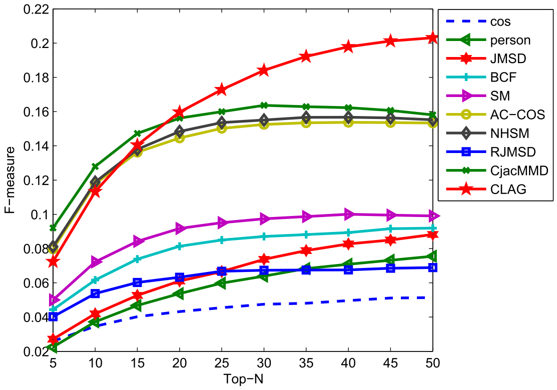


**FIGURE 7.** Comparison of the F-measure values of different algorithms for different recommendations on the Movielens 100-k dataset.

of recommendations increases, the F-measure value also increases and CLAG is superior to other algorithms.

## V. CONCLUSION

As we know, the user's scoring data is very sparse. The traditional similarity measures use the user's common score to lose a lot of useful information. The traditional collaborative filtering algorithms are difficult to accurately predict the score in the case of data sparseness. The CLAG algorithm proposed in this paper can make full use of the context information of the data, and introduce the user's scoring characteristics which still make accurate predictions under the Movielens-100k data set with only a few scores. Experiments show that CLAG algorithm can provide excellent prediction under high sparsity.

## REFERENCES

[1] M. Fu, H. Qu, Z. Yi, L. Lu, and Y. Liu, "A novel deep learning-based collaborative filtering model for recommendation system," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1084–1096, Mar. 2019.

[2] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Syst. Appl.*, vol. 69, pp. 29–39, Mar. 2017.

[3] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *CSUR ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–38, Feb. 2019.

[4] M. Reusens, W. Lemahieu, B. Baesens, and L. Sels, "Evaluating recommendation and search in the labor market," *Knowl.-Based Syst.*, vol. 152, pp. 62–69, Jul. 2018.

[5] S. Qin, R. Menezes, and M. Silaghi, "A recommender system for YouTube based on its network of reviewers," in *Proc. IEEE 2nd Int. Conf. Social Comput.*, Aug. 2010, pp. 323–328.

[6] W. Cheng, G. Yin, Y. Dong, H. Dong, and W. Zhang, "Collaborative filtering recommendation on users' interest sequences," *PLoS ONE*, vol. 11, no. 5, May 2016, Art. no. e0155739.

[7] C. B. Jiao, "Hybrid collaborative filtering recommendation algorithm based on model filling," *J. Softw.*, vol. 18, no. 7, pp. 1685–1694, 2011.

[8] C. Birtolo and D. Ronca, "Advances in clustering collaborative filtering by means of fuzzy c-means and trust," *Expert Syst. Appl.*, vol. 40, no. 17, pp. 6997–7009, Dec. 2013.

[9] P. Welter, J. Riesmeier, B. Fischer, C. Grouls, C. Kuhl, and T. M. Deserno, "Bridging the integration gap between imaging and information systems: A uniform data concept for content-based image retrieval in computer-aided diagnosis," *J. Amer. Med. Inform. Assoc.*, vol. 18, no. 4, pp. 506–510, Jul. 2011.

[10] Y. Lu Murphey, M. Masrur, Z. Chen, and B. Zhang, "Model-based fault diagnosis in electric drives using machine learning," *IEEE/ASME Trans. Mechatronics*, vol. 11, no. 3, pp. 290–303, Jun. 2006.

[11] C.-F. Hsiao, Y. Chen, and C.-Y. Lee, "A generalized mixed-radix algorithm for memory-based FFT processors," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 1, pp. 26–30, Jan. 2010.

[12] Suryakant and T. Mahara, "A new similarity measure based on mean measure of divergence for collaborative filtering in sparse environment," *Procedia Comput. Sci.*, vol. 89, pp. 450–456, 2016.

[13] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Inf. Sci.*, vol. 178, no. 1, pp. 37–51, Jan. 2008.

[14] C. Lin, R. Xie, L. Li, Z. Huang, T. Li, "PRemiSE: Personalized news recommendation via implicit social experts," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 1607–1611.

[15] F. Cacheda, V. Carneiro, D. Fernández, and V. Formoso, "Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems," *ACM Trans. Web*, vol. 5, no. 1, pp. 1–33, Feb. 2011.

[16] J. Bobadilla, F. Serradilla, and J. Bernal, "A new collaborative filtering metric that improves the behavior of recommender systems," *Knowl.-Based Syst.*, vol. 23, no. 6, pp. 520–528, Aug. 2010.

[17] J. Bobadilla, F. Ortega, and A. Hernando, "A collaborative filtering similarity measure based on singularities," *Inf. Process. Manage.*, vol. 48, no. 2, pp. 204–217, Mar. 2012.

[18] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," *Knowl.-Based Syst.*, vol. 26, pp. 225–238, Feb. 2012.

[19] K. Choi and Y. Suh, "A new similarity function for selecting neighbors for each target item in collaborative filtering," *Knowl.-Based Syst.*, vol. 37, no. 1, pp. 146–153, Jan. 2013.

[20] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering," *Knowl.-Based Syst.*, vol. 56, pp. 156–166, Jan. 2014.

[21] P. Pirasteh, D. Hwang, and J. E. Jung, "Weighted similarity schemes for high scalability in user-based collaborative filtering," *Mobile Netw. Appl.*, vol. 20, no. 4, pp. 497–507, Aug. 2015.

[22] B. K. Patra, R. Launonen, V. Ollikainen, and S. Nandi, "A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data," *Knowl.-Based Syst.*, vol. 82, pp. 163–177, Jul. 2015.

[23] Y. Wang, J. Deng, J. Gao, and P. Zhang, "A hybrid user similarity model for collaborative filtering," *Inf. Sci.*, vols. 418–419, pp. 102–118, Dec. 2017.

[24] S. Bag, S. K. Kumar, and M. K. Tiwari, "An efficient recommendation generation using relevant Jaccard similarity," *Inf. Sci.*, vol. 483, pp. 53–64, May 2019.

[25] H. Koohi and K. Kiani, "A new method to find neighbor users that improves the performance of Collaborative Filtering," *Expert Syst. Appl.*, vol. 83, pp. 30–39, 2017.

[26] J. Chen, Uliji, H. Wang, and Z. Yan, "Evolutionary heterogeneous clustering for rating prediction based on user collaborative filtering," *Swarm Evol. Comput.*, vol. 38, pp. 35–41, Feb. 2018.

[27] C.-L. Liu and X.-W. Wu, "Fast recommendation on latent collaborative relations," *Knowl.-Based Syst.*, vol. 109, pp. 25–34, Oct. 2016.

[28] N. Polatidis and C. K. Georgiadis, "A dynamic multi-level collaborative filtering method for improved recommendations," *Comput. Standards Inter.*, vol. 51, pp. 14–21, Mar. 2017.

[29] S. Zhang, L. Yao, and X. Xu, "AutoSVD++: An efficient hybrid collaborative filtering model via contractive auto-encoders," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2017, pp. 957–960.

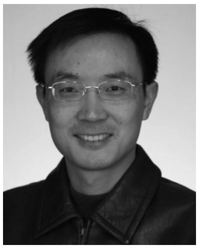
[30] Y. Gu, X. Yang, M. Peng, and G. Lin, "Robust weighted SVD-type latent factor models for rating prediction," *Expert Syst. Appl.*, vol. 141, Mar. 2020, Art. no. 112885.

[31] M. Jiang, Z. Zhang, J. Jiang, Q. Wang, and Z. Pei, "A collaborative filtering recommendation algorithm based on information theory and bi-clustering," *Neural Comput. Appl.*, vol. 31, no. 12, pp. 8279–8287, Dec. 2019.

[32] V. Maihami, D. Zandi, and K. Naderi, "Proposing a novel method for improving the performance of collaborative filtering systems regarding the priority of similar users," *Phys. A, Stat. Mech. Appl.*, vol. 536, Dec. 2019, Art. no. 121021.

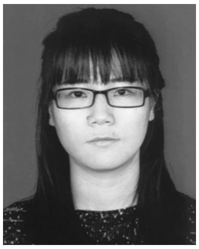
[33] A. K. Sahu, P. Dwivedi, and V. Kant, "Tags and item features as a bridge for cross-domain recommender systems," *Procedia Comput. Sci.*, vol. 125, pp. 624–631, 2018.

[34] J. Deng, Y. Wang, J. Guo, Y. Deng, J. Gao, and Y. Park, "A similarity measure based on Kullback–Leibler divergence for collaborative filtering in sparse data," *J. Inf. Sci.*, pp. 1–20, 2018.



**QIBING JIN** received the Ph.D. degree in control theory and engineering from Northeastern University, Shenyang, China, in 1999. He joined the Beijing University of Chemical Technology, Beijing, China, in 2002, where he is currently a Full Professor with the College of Information Science and Technology and also the Director of the Institute of Automation. His main research interests include advanced control, intelligent instrument, system identification, and control theory. He has rich experience in control engineering, and his many research results have been applied in petroleum and chemical industry. In recent years, he received several prizes for science and technology progress.

**YUE ZHANG** received the B.S. degree in communication engineering from Heilongjiang University, Harbin, China, in 2013. She is currently pursuing the M.S. degree in control science and engineering from the Beijing University of Chemical Technology, Beijing, China, in 2019. Her main research direction is engaged in machine learning.



**WU CAI** received the B.S. degree from the Beijing University of Chemical Technology, Beijing, China, in 2014, where she is currently pursuing the Ph.D. degree with the Department of Information Science and Engineering. Her research interest is in decoupling and disturbance rejection of multivariable systems.

**YUMING ZHANG** received the B.Sc. degree in automation from the Tianjin University of Commerce, Tianjin, China, in 2011. He is currently pursuing the Ph.D. degree with the Institute of Automation, Beijing University of Chemical Technology, Beijing, China. Since 2013, he has been at the Beijing University of Chemical Technology. His main research interests are modern control and system identification, including active disturbance rejection control, disturbance observer, model predictive control, and intelligent modeling.