

Research Article

Enhancing Collaborative Filtering by User-User Covariance Matrix

Yingyuan Xiao ^{1,2}, Jingjing Shi,^{1,2} Wenguang Zheng,^{1,2}
Hongya Wang,³ and Ching-Hsien Hsu ⁴

¹Key Laboratory of Computer Vision and System, Ministry of Education, 300384 Tianjin, China

²Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, China

³Donghua University, 201620 Shanghai, China

⁴National Chung Cheng University, Chiayi 62102, Taiwan

Correspondence should be addressed to Yingyuan Xiao; xiaoyingyuan_hust@126.com

Received 11 August 2018; Revised 10 October 2018; Accepted 4 November 2018; Published 25 November 2018

Academic Editor: Emilio Insfran Pelozo

Copyright © 2018 Yingyuan Xiao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The collaborative filtering (CF) approach is one of the most successful personalized recommendation methods so far, which is employed by the majority of personalized recommender systems to predict users' preferences or interests. The basic idea of CF is that if users had the same interests in the past they will also have similar tastes in the future. In general, the traditional CF may suffer the following problems: (1) The recommendation quality of CF based system is greatly affected by the sparsity of data. (2) The traditional CF is relatively difficult to adapt the situation that users' preferences always change over time. (3) CF based approaches are used to recommend similar items to a user ignoring the user's demand for variety. In this paper, to solve the above problems we build a new user-user covariance matrix to replace the traditional CF's user-user similarity matrix. Compared with the user-user similarity matrix, the user-user covariance matrix introduces the user-user covariance to finely describe the changing trends of users' interests. Furthermore, we propose an enhancing collaborative filtering method based on the user-user covariance matrix. The experimental results show that the proposed method can significantly improve the diversity of recommendation results and ensure the good recommendation precision.

1. Introduction

Recommendation systems have been widely applied to deal with information overload problems in e-commerce sites [1]. It is well known that collaborative filtering (CF) is one of the most widely used methods in recommender systems. CF utilizes users' behaviors (e.g., ratings or clicks) to infer a target user's preference for a particular item.

CF is based on the assumption that users' previous behaviors have huge influence on their future behaviors. The basic idea of CF is that if users shared the same interests in the past they will also have similar tastes in the future. Although CF approach has been employed by the majority of traditional personalized recommender systems, CF approach usually faces the following challenges: (1) The sparse data of the user-item matrix seriously affect the recommendation quality. (2) The traditional CF is relatively difficult to adapt

the situation that users' preferences always change over time. (3) Always recommending similar items to a user will fail to meet the user's demand for variety.

In this paper, we propose an enhancing collaborative filtering method based on covariance matrix named CFCM, such that the problems described above can be solved. We first construct the user-user covariance matrix according to the traditional user-item rating matrix, which alleviates the sparsity of the rating matrix. Secondly, we introduce the covariance to effectively capture the user's interest change. Finally, we adjust the weight between positive and negative reverse recommendations to improve the diversity of CF methods. The experimental results on a large MovieLens dataset demonstrate that CFCM outperforms the state-of-the-art CF methods.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 builds a new user-user

covariance matrix and presents an enhancing collaborative filtering method. Section 4 evaluates the proposed method through extensive experiments, and Section 5 concludes this paper.

2. Related Work

CF approach is one of the most successful personalized recommendation methods. The CF approach was first proposed by Goldberg et al. [2]. Since then, many improved CF methods have been proposed for personalized recommendation. CF methods can be grouped into two categories: memory-based and model-based.

The model-based CF approaches leverage training datasets to train a predefined model [3–8]. The memory-based CF approaches are most relevant to the method proposed in this paper. At present, the memory-based CF approaches are widely adopted in commercial personalized recommender systems [9, 10]. The key to these memory-based CF approaches is how to efficiently compute the similarity between two users. Then, for a particular user, his neighborhood can be determined according to the similarity and recommendations are made based on the rating of neighborhood of the target user. Many similarity measures (i.e., similarity functions) for these memory-based CF approaches have been proposed. In the following, we use U to denote the set of users, I to represent the set of items, I_u to denote the set of those items rated by user u , and R to denote the user-item rating matrix. The user-item rating matrix R is a $|U| \times |I|$ matrix, and each entry r_{ui} of R represents the rating of user u for item i . Let \vec{r}_u denote the rating vector of user u ; i.e., \vec{r}_u is a $|I|$ -dimensional vector constructed by all entries r_{ui} , where $i \in I$. We use \bar{r}_u to denote the average rating value of user u and \bar{r}_{med} to denote the median value of all rates in the user-item rating matrix R .

Given two users u and v , a similarity function $sim(u, v)$ returns a value that indicates the similarity of the two users. The larger the value is, the more similar the two users are. The *cosine* similarity is one of the widely used similarity functions. The *cosine* similarity function is defined as follows:

$$sim_{cos}(u, v) = \frac{\vec{r}_u \cdot \vec{r}_v}{\|\vec{r}_u\| \times \|\vec{r}_v\|} = \frac{\sum_{i \in I} r_{ui} \times r_{vi}}{\sqrt{\sum_{i \in I} r_{ui}^2} \times \sqrt{\sum_{i \in I} r_{vi}^2}} \quad (1)$$

The *Pearson correlation coefficient* (PCC) [11] is another of the most popular similarity measures for CF approaches. The PCC similarity function is defined by the following equation:

$$sim_{pcc}(u, v) = \frac{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u) \times (r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)^2} \times \sqrt{\sum_{i \in I_u \cap I_v} (r_{vi} - \bar{r}_v)^2}} \quad (2)$$

A variant of PCC is the *constrained Pearson correlation coefficient* (CPCC) [12]. The difference between CPCC and PCC is that CPCC uses the median value of all users' ratings

to replace the mean value of the specified user' ratings used in PCC. The CPCC similarity function is formulated as follows:

$$sim_{cpcc}(u, v) = \frac{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_{med}) \times (r_{vi} - \bar{r}_{med})}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_{med})^2} \times \sqrt{\sum_{i \in I_u \cap I_v} (r_{vi} - \bar{r}_{med})^2}} \quad (3)$$

Jaccard and *mean squared difference* (MSD) are also commonly used as similarity measures [13]. Their similarity functions are, respectively, defined as follows:

$$sim_{jac}(u, v) = \frac{|I_u \cap I_v|}{|I_u \cup I_v|} \quad (4)$$

$$sim_{MSD}(u, v) = 1 - \frac{\sum_{i \in I} (r_{ui} - r_{vi})^2}{|I_u \cup I_v|}$$

Further, Bobadilla et al. [14] combine *Jaccard* and MSD and propose a new similarity measure called JMSD. JMSD is defined as the following equation:

$$sim_{JMSD}(u, v) = sim_{jac}(u, v) \times sim_{MSD}(u, v) \quad (5)$$

In addition, more works on similarity measures [15–19] have been investigated. Bobadilla et al. [15] present a new similarity measure, called MJM, which leverages optimization based on neural learning to combine the existing similarity measures. Ahn [16] proposes a new heuristic similarity measure called PIP (Proximity-Impact-Popularity), which is composed of three factors (proximity, impact, and popularity) of similarity and utilizes domain specific interpretation of user ratings to overcome the cold-start problem. Pirasteh et al. [17] introduce new weighting schemes and propose a new similarity measure called AC-PCC. AC-PCC allows us to consider new features in finding similarities between users. Liu et al. [18] present a new user similarity model (NHSM) to improve the recommendation performance when only few ratings are available to calculate the similarities. NHSM considers the local context information of user ratings to infer a target user's preference for a particular item. Patraet al. [19] propose BCF similarity measure, which utilizes all ratings information comprehensively for locating useful neighbors of an active user in sparse ratings dataset.

Obviously, the existing traditional similarity measures focus on how to infer a target user's local preference based on his/her similar users' previous behaviors and ignore more or less the fact that a user's preference or interest always changes dynamically. In addition, ensuring the diversity of recommendations is often a desirable feature in recommender systems. Therefore, how to truly reflect the change of user's interest is an important issue in designing a similarity function. In this paper, we introduce a novel method by using covariance to depict the changing trends of users' interests and build a new user-user covariance matrix to model user similarity.

3. Enhancing Collaborative Filtering Approach

In this section, we first introduce **covariance** to build a new user-user covariance matrix and then propose an enhancing

collaborative filtering method based on the user-user covariance matrix.

3.1. User-User Covariance Matrix. In probability theory and statistics, covariance is a measure of the joint variability of two random variables. If the trend of the two variables is consistent, the covariance between the two variables is positive. In the opposite case, if two variables change in the opposite direction, the covariance between the two variables is negative. The sign of the covariance therefore shows the tendency in the linear relationship between the variables.

Considering the above-mentioned statistical characteristics of covariance, we construct a new user-user covariance matrix to model user similarity. Let u_i and u_j denote any two users in U , $|U| = m$, I represent the set of items, I_i represent the set of items rated by u_i , I_j denote the set of items rated by u_j , r_{ik} denote the rating of user u_i for item i_k , and r_{jk} represent the rating of user u_j for item i_k . Now, we formally define the user-user covariance matrix.

Definition 1. A user-user covariance matrix is an $m \times m$ matrix, denoted by $C = \{c_{ij}\}_{m \times m}$, where

$$c_{ij} = \text{cov}(u_i, u_j) = \begin{cases} +\infty, & i = j \\ E[(u_i - E(u_i)) \cdot (u_j - E(u_j))], & i \neq j \end{cases} \quad (6)$$

In the above definition, $\text{cov}(u_i, u_j)$ is set to $+\infty$, when $i = j$. When $i \neq j$, $\text{cov}(u_i, u_j)$ denotes the covariance between u_i and u_j . Since each user is modeled as the rating sequence for all items in CF approaches, we can consider a user as a random variable, which means that it makes sense to calculate the covariance between two users. Like the covariance between two variables in probability theory and statistics, we give the calculation method for the covariance between two users at $i \neq j$ in

$$\begin{aligned} \text{cov}(u_i, u_j) &= E[(u_i - E(u_i)) \cdot (u_j - E(u_j))] \\ &= E(u_i u_j) - E(u_i) E(u_j) \\ &= \frac{1}{|I|} \sum_{k \in I} r_{ik} \cdot r_{jk} - \frac{1}{|I|^2} \sum_{k \in I} r_{ik} \cdot \sum_{k \in I} r_{jk} \\ &= \frac{1}{|I|} \sum_{k \in (I_i \cap I_j)} r_{ik} \cdot r_{jk} - \frac{1}{|I|^2} \sum_{k \in I} r_{ik} \cdot \sum_{k \in I} r_{jk} \end{aligned} \quad (7)$$

Below we use an example to illustrate how to use (7) to calculate the covariance between two users. let $\vec{r}_1 = (r_{11}, r_{12}, r_{13}, r_{14}, r_{15}, r_{16}) = (0, 5, 2, 0, 3, 0)$ denote the rating vector of user u_1 for items i_1, i_2, i_3, i_4, i_5 , and i_6 , and $\vec{r}_2 = (r_{21}, r_{22}, r_{23}, r_{24}, r_{25}, r_{26}) = (4, 0, 0, 3, 0, 4)$ represent the rating vector of user u_2 for items i_1, i_2, i_3, i_4, i_5 , and i_6 . According to (7), the covariance $\text{cov}(u_1, u_2)$ between u_1 and u_2 is calculated as follows:

$$\text{cov}(u_1, u_2) = (1/|I|) \sum_{k \in (I_1 \cap I_2)} r_{1k} \cdot r_{2k} - (1/|I|^2) \sum_{k \in I} r_{1k} \cdot \sum_{k \in I} r_{2k}, \text{ where } I_1 = \{i_2, i_3, i_5\}, I_2 = \{i_1, i_4, i_6\}, I = \{i_1, i_2,$$

TABLE 1: A user-user covariance matrix.

	u_1	u_2	u_3	u_4	u_5
u_1	$+\infty$	-3.056	-0.444	-2.222	2.222
u_2	-3.056	$+\infty$	0.111	2.389	-1.972
u_3	-0.444	0.111	$+\infty$	-0.889	1.222
u_4	-2.222	2.389	-0.889	$+\infty$	-2.889
u_5	2.222	-1.972	1.222	-2.889	$+\infty$

TABLE 2: A user-item rating matrix.

	i_1	i_2	i_3	i_4	i_5	i_6
u_1	null	5	2	null	3	null
u_2	4	null	null	3	null	4
u_3	null	null	2	null	null	2
u_4	5	null	null	3	null	null
u_5	null	5	5	null	null	3

$i_3, i_4, i_5, i_6\}$, $I_1 \cap I_2 = \{i_2, i_3, i_5\} \cap \{i_1, i_4, i_6\} = \emptyset$, and $|I| = 6$. That is, $\text{cov}(u_1, u_2) = 1/6 \times 0 - (1/6^2)(r_{11} + r_{12} + r_{13} + r_{14} + r_{15} + r_{16}) \times (r_{21} + r_{22} + r_{23} + r_{24} + r_{25} + r_{26}) = 0 - (1/36)(0 + 5 + 2 + 0 + 3 + 0) \times (4 + 0 + 0 + 3 + 0 + 4) \approx -3.056$.

Further, Table 1 demonstrates a user-user covariance matrix, which is derived by (7) from a user-item rating matrix shown in Table 2. As shown in Table 1, a user-user covariance matrix is a symmetric matrix and adds the negative correlation to depict the opposite trends of users' interests. For traditional similarity measures, if two users have no common rated items, the similarity between the two users is usually equal to zero. For example, the similarity between u_1 and u_2 is equal to the similarity between u_1 and u_4 , and both values are zero according to the traditional similarity measures, because u_1 have no common rated items with u_2 and u_4 , as shown in Table 2. This means that the traditional similarity measures cannot distinguish the degree of dissimilarity between the users who have no common rated items. However, we can find from Table 1 that the covariance $\text{cov}(u_1, u_2)$ between u_1 and u_2 is different from the covariance $\text{cov}(u_1, u_4)$ between u_1 and u_4 , both values are negative, and the absolute value of $\text{cov}(u_1, u_2)$ is bigger than that of $\text{cov}(u_1, u_4)$. It means that u_2 is more disinterested in those items, which u_1 are of interest to, than u_4 . From the above example, we can learn that the user-user covariance matrix has a finer description capability for user similarity trends, and especially for the users without common rated items, it shows the more powerful description capability than the traditional similarity measures.

3.2. The Proposed Collaborative Filtering Algorithm. In the proposed enhancing collaborative filtering method based on covariance matrix (CFCM), we first build the user-user covariance matrix $C = \{c_{ij}\}_{m \times m}$ based on the user-item rating matrix. The above Definition 1 and (7) give the way of computing $C = \{c_{ij}\}_{m \times m}$. Then, for any target user u , we employ the user-user covariance matrix to calculate its K neighbors with the most similar interested trends and opposite K neighbors with the most dissimilar interested

TABLE 3: The basic attributes of MovieLens-1M.

Dataset	Number of users	Number of Movies	Number of Ratings	Sparsity level	Rating domain
MovieLens-1M	6040	3706	1M	4.2%	{1, 2, 3, 4, 5}

Input: R and u , where R is the user-item rating matrix and u denotes a target user.
Output: *Result*, i.e., the recommendation list for u .
1: $Result := \emptyset; N_{u,s} := \emptyset; N_{u,d} := \emptyset; Orderlist := \emptyset;$
2: build the user-user covariance matrix $C = \{c_{ij}\}_{m \times m}$ based on R ;
3: $N_{u,s} := \text{GetKSimilarUser}(u);$
4: $N_{u,d} := \text{GetKDSimilarUser}(u);$
5: **for** $\forall i \in I$ **do**
6: $\hat{r}_{ui} = \bar{r}_u + \alpha \times \frac{\sum_{v \in N_{u,s}} cov(u, v) \times (r_{vi} - \bar{r}_v)}{\sum_{v \in N_{u,s}} |cov(u, v)|} + (1 - \alpha) \times \frac{\sum_{v \in N_{u,d}} |cov(u, v)| \times (r_{vi} - \bar{r}_v)}{\sum_{v \in N_{u,d}} |cov(u, v)|};$
7: insert i into *Orderlist* in descending order of \hat{r}_{ui} ;
8: **end for**
9: $Result := \text{GetFirstn}(Orderlist)$
10: **return** *Result*;

ALGORITHM 1: CFCM (R, u).

trends, respectively. In this paper, we use $N_{u,s}$ to denote the set of u 's K most similar neighbors and $N_{u,d}$ to denote the set of u 's K most dissimilar neighbors. Finally, the rating prediction \hat{r}_{ui} of user u for item i is computed by the following equation:

$$\hat{r}_{ui} = \bar{r}_u + \alpha \times \frac{\sum_{v \in N_{u,s}} cov(u, v) \times (r_{vi} - \bar{r}_v)}{\sum_{v \in N_{u,s}} |cov(u, v)|} + (1 - \alpha) \times \frac{\sum_{v \in N_{u,d}} |cov(u, v)| \times (r_{vi} - \bar{r}_v)}{\sum_{v \in N_{u,d}} |cov(u, v)|} \quad (8)$$

In (8), \bar{r}_u denotes u 's average rating; $(\sum_{v \in N_{u,s}} cov(u, v) \times (r_{vi} - \bar{r}_v)) / \sum_{v \in N_{u,s}} |cov(u, v)|$ reflects the influence on \hat{r}_{ui} from the users who have the most similar interested trends with u , which ensures the recommendation precision; $(\sum_{v \in N_{u,d}} |cov(u, v)| \times (r_{vi} - \bar{r}_v)) / \sum_{v \in N_{u,d}} |cov(u, v)|$ represents the influence on \hat{r}_{ui} from the users who have the most dissimilar interested trends with u , which provides the diversity for the recommended result; α is the threshold that is employed to control the proportion of the above two aspects of influences. α takes a real value between 0 and 1.

The difference between CFCM and the traditional CF approaches is mainly reflected in the following two aspects: (1) CFCM uses the covariance $cov(u, v)$ instead of the similarity $sim(u, v)$ of the traditional CF approaches to enhance the recommendation precision, and (2) CFCM leverages the set of u 's K most dissimilar neighbors to improve the diversity of recommendation results.

Based on the above processing strategy, we formalize CFCM method in Algorithm 1.

In Algorithm 1, the function $\text{GetKSimilarUser}(u)$ is responsible for getting the top K users who have the greatest covariance with u , the function $\text{GetKDSimilarUser}(u)$ is responsible for getting the top K users who have the smallest

covariance with u , and the function $\text{GetFirstn}(Orderlist)$ retrieves the first n items from *Orderlist*.

4. Performance Evaluation

In this section, we evaluate the performance of the proposed approach (CFCM) through extensive experiments. We first describe the experimental dataset chosen for our experiments in Section 4.1. Then, in Section 4.2 we introduce the performance metrics of concern in this paper. Finally, we present and analyze the experimental results in Section 4.3.

4.1. Dataset Description. Similar to the existing literature related CF methods, we choose MovieLens (<https://grouplens.org/datasets/movielens/>) as the data source for our experiments. MovieLens includes integer ratings of users for movies. Specifically, the dataset used in our experiments is MovieLens-1M (<https://grouplens.org/datasets/movielens/1m/>) that is a stable benchmark dataset. MovieLens-1M consists of 6040 users who have rated a total of 3706 different movies and the total number of ratings is 1000209. All movies are divided into 18 basic categories by their themes. The sparsity level of the user-movie rating matrix is about 4.2% for MovieLens-1M. Here, the sparsity level is described as the percentage of all possible ratings available in a dataset. Table 3 shows the basic attributes of MovieLens-1M.

Like the literature [15–18], we randomly pick 80% of ratings from MovieLens-1M as the training set and the rest as the testing set.

4.2. Performance Metrics. In our experiments, we use four performance metrics, the *Mean Absolute Error (MAE)*, the *Normalized Mean Absolute Error (NMAE)*, *F1*, and *Diversity*

to measure the performance of the proposed approach (CFCM).

MAE is a popular evaluation metric widely used to measure the precision of a recommendation method. MAE is defined by the following equation:

$$MAE = \frac{\sum_{ij} |r_{ij} - \widehat{r}_{ij}|}{n} \quad (9)$$

where r_{ij} represents the rating of user u_i for item j , \widehat{r}_{ij} denotes the rating prediction of user u_i for item j by a recommendation method, and n represents the number of tested ratings.

$NMAE$ is the normalized MAE and is formulated as follows:

$$NMAE = \frac{1}{r_{max} - r_{min}} \frac{\sum_{ij} |r_{ij} - \widehat{r}_{ij}|}{n} \quad (10)$$

where r_{max} and r_{min} , respectively, represent the maximum and minimum values of all user's ratings.

MAE and $NMAE$ are used to describe the prediction precision of recommendation methods. The smaller MAE and $NMAE$ mean the more accurate prediction.

$F1$ is introduced as one of the most important comprehensive performance metrics for measuring the performance of the recommendation methods. $F1$ is defined by the following equation:

$$F1 = \frac{2}{1/precision + 1/recall} \quad (11)$$

where the definitions of *precision* and *recall* are formulated by the following equations, respectively:

$$precision = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \quad (12)$$

$$recall = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \quad (13)$$

where $R(u)$ represents the recommendation result set generated for user u by a recommendation method and $T(u)$ denotes the result set for user u generated by u 's actual behaviors on the testing set. Similar to some previous work, we use $N=20$ as the number of items recommended to the users while computing *precision*, *recall* and $F1$.

The higher $F1$ of a recommendation method means the better comprehensive performance of the recommendation method.

Diversity, as an important performance metric, is used to describe the dissimilarity between the items in the recommendation result set. The larger *Diversity* means the better recommended performance at the same *precision*. *Diversity* is formulated as follows:

$$Diversity = \frac{1}{|U|} \sum_{u \in U} \left(1 - \frac{\sum_{i, j \in R(u), i \neq j} s(i, j)}{(1/2) |R(u)| (|R(u)| - 1)} \right) \quad (14)$$

where $s(i, j)$ denotes the Jaccard similarity of i and j .

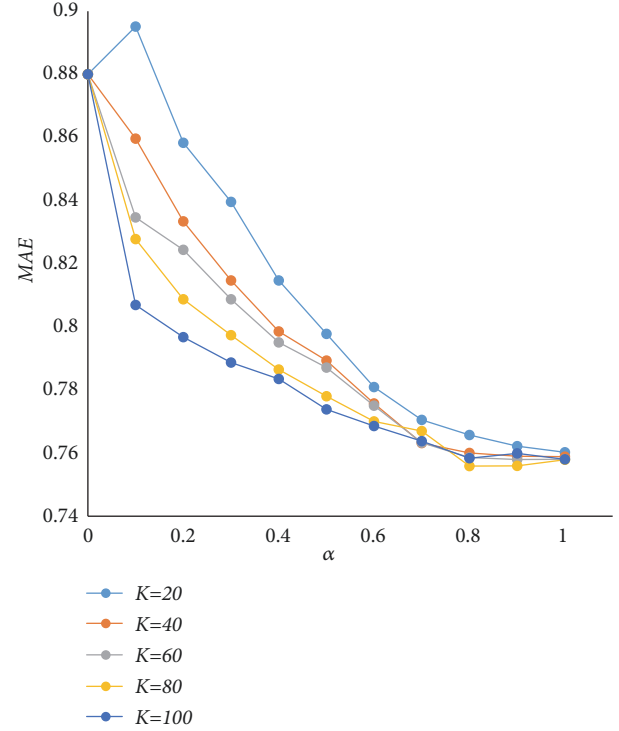
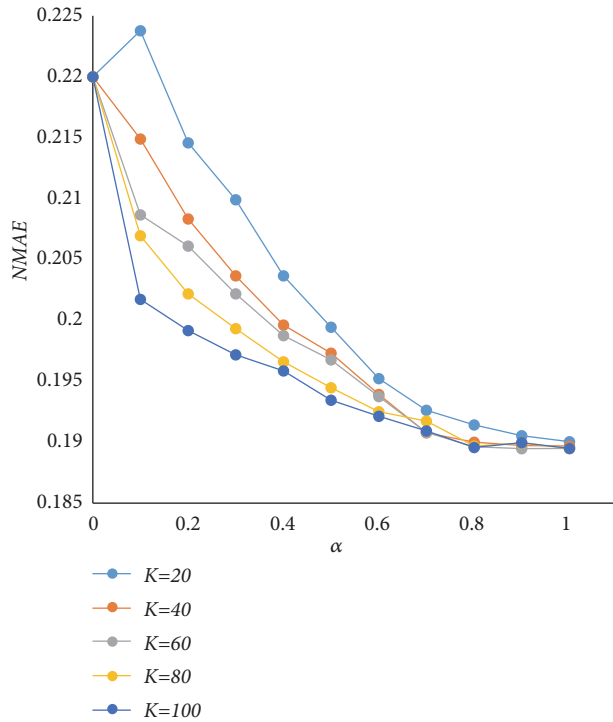
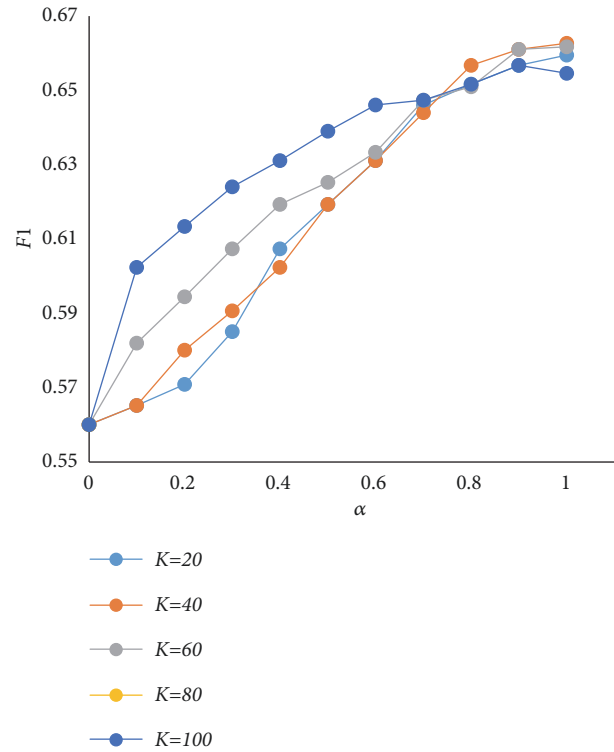


FIGURE 1: Impact of α and K on MAE .

4.3. Experimental Results. It can be seen from (8) that α and the number K of neighbors with the most similar or most dissimilar interested trend have an inevitable influence on the performance of CFCM. In this section, we first evaluate the impact of α and K on the performance of CFCM, and then we compare CFCM with the state-of-the-art approaches based on traditional CF methods. We can find that at $\alpha = 0$, (8) degenerates to $\widehat{r}_{ui} = \bar{r}_u + (\sum_{v \in U_{u,d}} |cov(u, v)| \times (r_{vi} - \bar{r}_v)) / \sum_{v \in U_{u,d}} |cov(u, v)|$, i.e., the calculation of \widehat{r}_{ui} only considers K most dissimilar neighbors of the target user. Obviously, $\alpha = 0$ is a special case for our method.

Figure 1 shows the impact of the parameters α and K on MAE . As shown in Figure 1, we can learn that when $\alpha \neq 0$, MAE decreases with α for any fixed K ; i.e., the recommendation precision increases with α for any fixed K . This result is not surprising because α represents the weight of ensuring the recommended precision in (8), and a larger α means a higher recommendation precision. Furthermore, we can find from Figure 1 that MAE remains relatively stable when α is greater than 0.8 which means that there is much less benefit to improve the prediction precision of recommendation by continually increase α when it is larger the 0.8. We can also observe that MAE decreases with K when α takes a fixed value between 0 and 0.7. This is because the larger K means more neighbors are considered when calculating the rating prediction. The more neighbors naturally lead to the better recommendation precision. In addition, for a fixed α between 0.7 and 1, MAE no longer decreases with K when K is greater than 20. This demonstrates when α is greater than 0.7, the recommendation precision cannot be simply improved

FIGURE 2: Impact of α and K on $NMAE$.FIGURE 3: Impact of α and K on $F1$.

by increasing K . For $\alpha = 0$, the trend of MAE is slightly fluctuating only when K is relatively small, such as $K = 20$. At the same time, we find that K had very little influence on MAE at $\alpha = 0$.

Figure 2 demonstrates the impact of the parameters α and K on $NMAE$. As we expect, the result and the trend of the curve are very similar to Figure 1. The reason is that $NMAE$ is the normalized MAE .

Figure 3 depicts the impact of the parameters α and K on $F1$. We can see from Figure 3 that $F1$ increases with α for any fixed K , that is, the recommendation comprehensive performance (including precision and recall) increases with α for any fixed K . The reason is the same as the one for Figure 1. We can also observe from Figure 3 that $F1$ increases with K when α takes a fixed value between 0 and 0.7, and the growth rate gradually reduces as α increases. When α takes a fixed value between 0.7 and 1, $F1$ no longer increases with K . This demonstrates when α is greater than a certain threshold (i.e., 0.7), it is not feasible to improve the recommendation comprehensive performance by increasing K .

Figure 4 shows the impact of the parameters α and K on $Diversity$. We can find from Figure 4 that $Diversity$ decreases with α for any fixed K , and this decreasing trend becomes more apparent when α is greater than 0.8. This result is not surprising because the increase of α means the decrease of the weight that ensures the recommendation diversity. We can also observe from Figure 4 that $Diversity$ has a very small change with K when α takes a fixed value between 0 and 0.9. This demonstrates that $Diversity$ mainly depends on α and is less affected by K .

By analyzing Figures 1–4, we can find that when α takes 0.8, CFCM can achieve the optimal overall performance in terms of both recommendation precision and recommendation diversity.

Further, in order to demonstrate the superiority of CFCM, we compare CFCM with the existing CF methods based on the traditional similarity measures. In the state-of-the-art CF methods, we select PCC [11], CPCC [12], JMSD [14], MJD [15], PIP [16], AC-PCC [17], NHSM [18], and BCF [19] as the baseline methods of our experiments. For CFCM, the parameter α is set to 0.8 in the following experiments. To ensure fairness, we adopt the optimal experimental parameter setting for each baseline method.

Figure 5 shows the performance comparison of CFCM with the above-mentioned eight methods on MAE for different K . As we expect, our method CFCM evidently outperforms the other eight methods on MAE at all K values. We can also observe from Figure 5 that the MAE of CFCM is the most stable and is almost unaffected by K . The reason is that the user-user covariance matrix has a finer description capability for user similarity trend than the traditional similarity measures.

Figure 6 depicts the performance comparison of CFCM with the other eight methods on $NMAE$ for different K . As shown in Figure 6, CFCM evidently outperforms the other eight methods on $NMAE$ at all K values. Similar to Figure 5, the $NMAE$ of CFCM is the most stable and is almost unaffected by K .

Figure 7 demonstrates the performance comparison of CFCM with the other eight methods on $F1$ for different K .

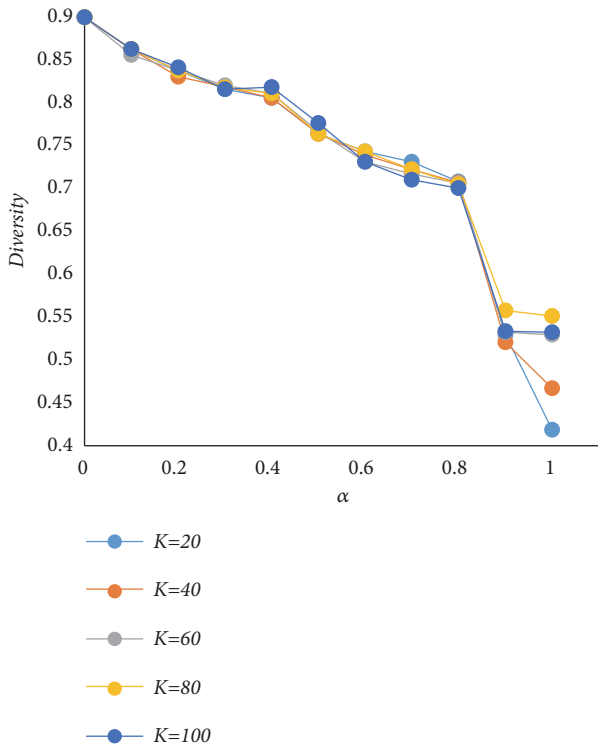


FIGURE 4: Impact of α and K on Diversity.

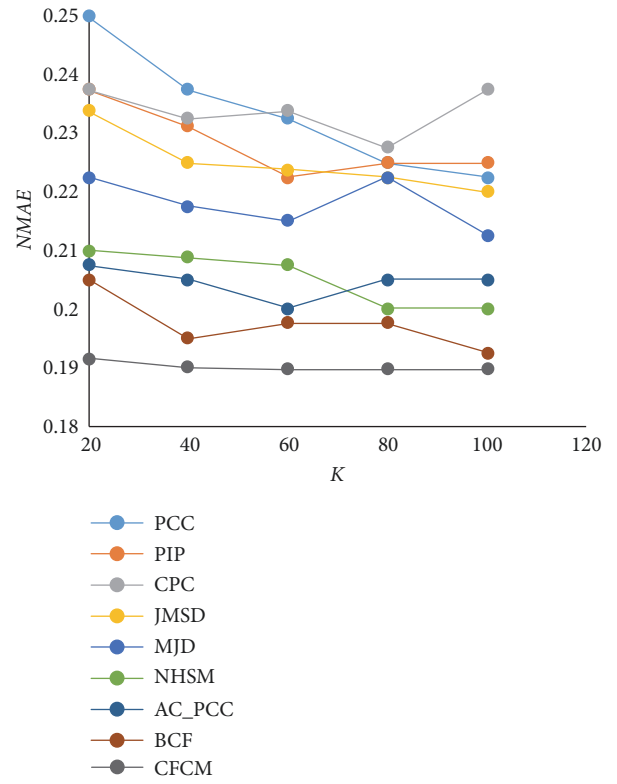


FIGURE 6: Performance comparison on NMAE.

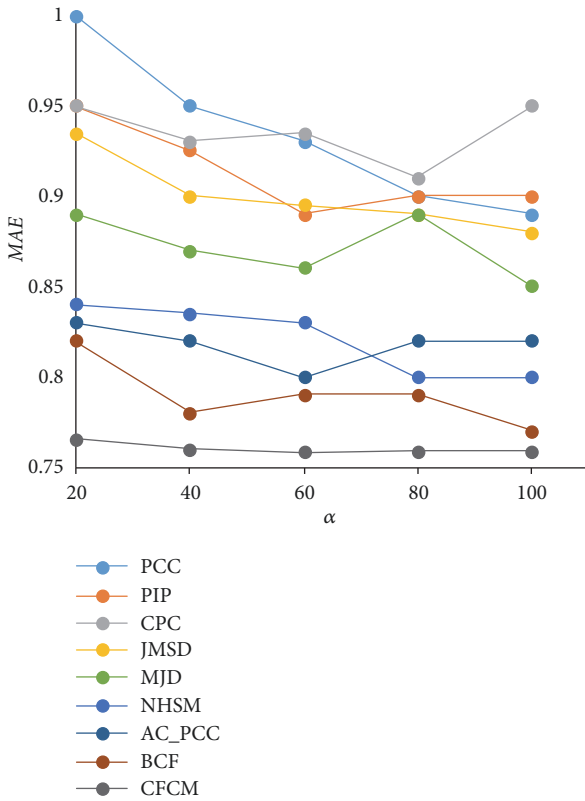


FIGURE 5: Performance comparison on MAE.

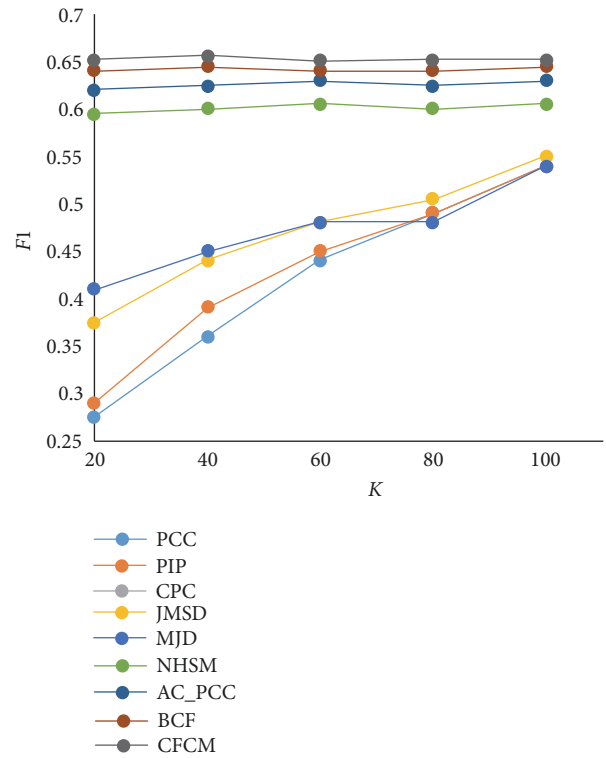


FIGURE 7: Performance comparison on F1.

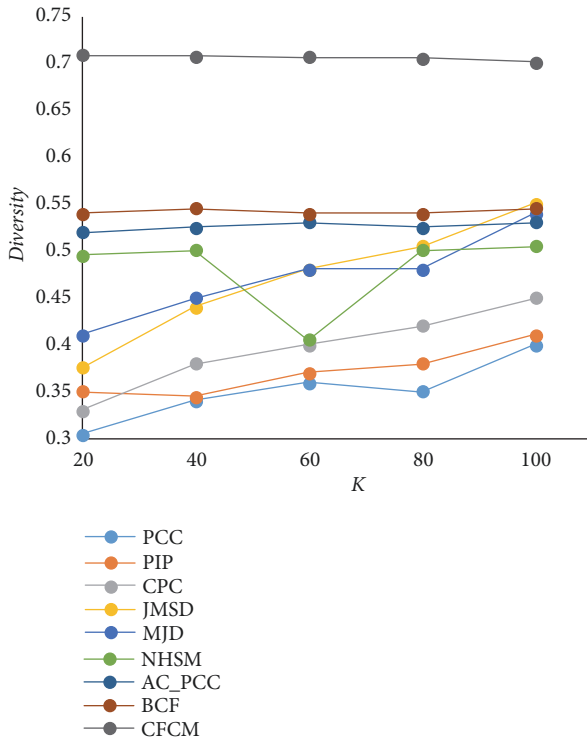


FIGURE 8: Performance comparison on *Diversity*.

We can see from Figure 7 that CFCM evidently outperforms the other eight methods on *F1* at all *K* values. We can also observe from Figure 7 that the *NMAEs* of CFCM, NHSM and AC_PCC are almost unaffected by *K* and have a good stability.

Figure 8 shows the performance comparison of CFCM with the other eight methods on *Diversity* for different *K*. As we expect, CFCM evidently outperforms the other eight methods on *Diversity* at all *K* values. This is because CFCM adds the influencing factor from the neighbors with the most dissimilar interested trend into the rating prediction, which enhances the diversity of recommendation results.

5. Conclusion

Although the traditional CF methods have been widely studied and deployed, most of these methods suffer several inherent weaknesses. In this paper, to solve the inherent weaknesses we build a new user-user covariance matrix to replace the traditional CF's user-user similarity matrix. Compared to the user-user similarity matrix, the user-user covariance matrix has a finer description capability for user similarity trends, and especially for the users without common rated items, it shows the more powerful description capability. Further, we propose an enhancing collaborative filtering method based on the user-user covariance matrix. The experimental results show that the proposed method can significantly improve the diversity of recommendation results in ensuring a good recommendation precision.

In the future work, we will consider introducing new quality measure metrics, such as coverage quality measure

[20], to measure the proposed approach more comprehensively.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declared no conflicts of interest regarding the submission of this manuscript.

Acknowledgments

This work is supported by the National Nature Science Foundation of China (61702368 and 61170174), Major Research Project of National Nature Science Foundation of China (91646117), Natural Science Foundation of Tianjin (17JCY-BJC15200, 15JCYBJC46500, and 18JQJNJ00700), and the Opening Foundation of Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology (TJUT-KLICNST-K20170001).

References

- [1] Y. Xiao, G. Wang, C. Hsu, and H. Wang, "A time-sensitive personalized recommendation method based on probabilistic matrix factorization technique," *Soft Computing*, vol. 22, no. 20, pp. 6785–6796, 2018.
- [2] 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.
- [3] T. Hofmann, "Collaborative filtering via gaussian probabilistic latent semantic analysis," in *Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 259–266, Toronto, Canada, 2003.
- [4] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Transactions on Information and System Security*, vol. 22, no. 1, pp. 89–115, 2004.
- [5] A. Kohrs and B. Merialdo, "Clustering for collaborative filtering applications," in *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation*, 1999.
- [6] J. D. M. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, pp. 713–719, August 2005.
- [7] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using markov chain Monte Carlo," in *Proceedings of the 25th International Conference on Machine Learning (ICML '08)*, pp. 880–887, ACM, Helsinki, Finland, July 2008.
- [8] Y. Xiao, P. Ai, C.-H. Hsu, H. Wang, and X. Jiao, "Time-ordered collaborative filtering for news recommendation," *China Communications*, vol. 12, no. 12, pp. 53–62, 2015.
- [9] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.
- [10] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering

- of netnews,” in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW 1994*, pp. 175–186, USA, October 1994.
- [11] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, “Collaborative filtering recommender systems,” *ACM Transactions on Information Systems*, vol. 4, no. 2, pp. 81–173, 2011.
- [12] U. Shardanand and P. Maes, “Social information filtering: algorithms for automating ‘word of mouth,’” in *Proceedings of the Conference on Human Factors in Computing Systems*, pp. 210–217, May 1995.
- [13] 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 Transactions on the Web (TWEB)*, vol. 5, no. 1, pp. 1–33, 2011.
- [14] J. Bobadilla, F. Serradilla, and J. Bernal, “A new collaborative filtering metric that improves the behavior of recommender systems,” *Knowledge-Based Systems*, vol. 23, no. 6, pp. 520–528, 2010.
- [15] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, “A collaborative filtering approach to mitigate the new user cold start problem,” *Knowledge-Based Systems*, vol. 26, pp. 225–238, 2012.
- [16] 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, 2008.
- [17] P. Pirasteh, D. Hwang, and J. E. Jung, “Weighted Similarity Schemes for High Scalability in User-Based Collaborative Filtering,” *Mobile Networks and Applications*, vol. 20, no. 4, pp. 1–11, 2014.
- [18] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, “A new user similarity model to improve the accuracy of collaborative filtering,” *Knowledge-Based Systems*, vol. 56, pp. 156–166, 2014.
- [19] B. K. Patra, R. Launonen, V. Ollikainen, and S. Nandi, “A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data,” *Knowledge-Based Systems*, vol. 82, pp. 163–177, 2015.
- [20] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” *Knowledge-Based Systems*, vol. 46, pp. 156–166, 2013.

