

Research Article

A Hybrid Recommender System for Gaussian Mixture Model and Enhanced Social Matrix Factorization Technology Based on Multiple Interests

Rui Chen ^{1,2}, Qingyi Hua ¹, Quanli Gao,³ and Ying Xing⁴

¹School of Information Science & Technology, Northwest University, Xi'an 710127, China

²School of Computer Science, Zhengzhou University of Aeronautics, Zhengzhou 450015, China

³College of Computer Science, Xi'an Polytechnic University, Xi'an 710048, China

⁴College of Software, Zhongyuan University of Technology, Zhengzhou 450007, China

Correspondence should be addressed to Qingyi Hua; huaqy@nwu.edu.cn

Received 3 April 2018; Revised 29 July 2018; Accepted 29 August 2018; Published 3 October 2018

Academic Editor: Rosa M. Benito

Copyright © 2018 Rui Chen 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.

Recommender systems are recently becoming more significant in the age of rapid development of the information technology and pervasive computing to provide e-commerce users' appropriate items. In recent years, various model-based and neighbor-based approaches have been proposed, which improve the accuracy of recommendation to some extent. However, these approaches are less accurate than expected when users' ratings on items are very sparse in comparison with the huge number of users and items in the user-item rating matrix. Data sparsity and high dimensionality in recommender systems have negatively affected the performance of recommendation. To solve these problems, we propose a hybrid recommendation approach and framework using Gaussian mixture model and matrix factorization technology. Specifically, the improved cosine similarity formula is first used to get users' neighbors, and initial ratings on unrated items are predicted. Second, users' ratings on items are converted into users' preferences on items' attributes to reduce the problem of data sparsity. Again, the obtained user-item-attribute preference data is trained through the Gaussian mixture model to classify users with the same interests into the same group. Finally, an enhanced social matrix factorization method fusing user's and item's social relationships is proposed to predict the other unseen ratings. Extensive experiments on two real-world datasets are conducted and the results are compared with the existing major recommendation models. Experimental results demonstrate that the proposed method achieves the better performance compared to other techniques in accuracy.

1. Introduction

Collaborative filtering (CF) is one of the most mature and widely used recommendation algorithms, which can be divided into model-based methods and memory-based methods [1, 2]. The former uses machine learning methods to model users' preferences by training data and predicts unknown ratings through the trained model. The latter is a neighbor-based approach, which calculates the similarities between users/items to get neighbor users with similar interests to an active user or neighbor items with similar characteristics to an active item, so as to predict and recommend the interest items for the active user.

According to recommendation strategies, recommendation algorithms can be divided into CF recommendation algorithm, content-based filtering, knowledge-based filtering, association rule-based and graph-based recommendation algorithm, etc. [3, 4]. Due to the shortcomings of a single algorithm in the recommendation performance, such as cold-start problem of CF algorithm and the lack of diversity of recommended resources in the content-based recommender system (RS) [5, 6], the researchers begin to merge together a variety of algorithms to improve the recommendation performance. Hernando et al. [7] introduce the reliability measure into prediction based on CF, to improve the prediction information provided to users. Chen et al. [8] propose a

hybrid model, which combines Gaussian mixture model with item-based CF recommendation algorithm and predicts the ratings on items from users to improve the recommendation accuracy. Luo et al. [9] propose a hierarchical Bayesian model-based CF and related inference algorithm, which reduces the prediction errors. Hofmann et al. [10] propose an algorithm of latent semantic models for CF, which is a user-centric view technique and introduces probabilistic latent semantic analysis (pLSA) into user modeling process, and it achieves higher prediction accuracies.

In recent years, some researchers introduced the context information into RS to improve the accuracy of the recommendation algorithms. Massa et al. [11] propose a trust-aware collaborative filtering algorithm, which increases the coverage of recommender systems with preserving the quality of predictions. Azadjalal et al. [12] employ trust relationships into recommendation algorithms and propose a method to identify implicit trust statements by applying a specific reliability measure. The proposed algorithm outperforms the traditional recommenders in accuracy and coverage measures. Gao et al. [4] propose a method to obtain context-aware preference based on cognitive behaviors, which elicits user preferences under multidimensional context environments by establishing the cognitive factors' mutual effect model, to achieve better prediction accuracy.

The above methods improve the accuracy of the recommendation to some extent. However, the user-item rating matrix, namely, one of the inputs to the recommendation algorithm for large amounts of data, is often highly sparse, which leads to unreliable predictions [13–17]. The main reasons are as follows: (1) excessive sparsity leads to the lack of common rating items, so neighbors cannot be selected to make predictions; (2) due to too few neighbors and common rating items, similarity is not accuracy; thus the quality of recommendation is difficult to guarantee; (3) a variety of recommendation algorithms proposed cannot predict the ratings or make the recommendations for all the items; (4) matrix factorization algorithms can predict ratings for all the items, but the accuracy did not reach the desired effect on the basis of data sparsity.

To overcome the above problems, a hybrid recommendation method that combines the Gaussian mixture model with an enhanced social matrix factorization algorithm is proposed in this paper. To reduce the sparsity and predict unknown ratings more accurately and completely, we first fill in the rating matrix from the perspectives of users, items, and users' interests so that an accurate recommendation can be made for users to achieve users' satisfaction. Specifically, CF and Gaussian mixture model are used to fill in the user-item rating matrix, and then matrix factorization technique is used to predict all unknown ratings. Firstly, we make a preliminary prediction using the proposed improved similarity method to reduce the data sparsity. Secondly, in order to classify users with the same interests into same group, ratings on items from users are converted into the users' preferences on items' attributes. Based on the assumption that each user has multiple interests, the users are divided into different groups according to different interests by using the Gaussian mixture model. Then the partial unknown ratings are predicted using

the probability that each user belongs to each different group. Finally, all unrated items are predicted by using the proposed enhanced social matrix factorization.

The remainder of this paper is organized as follows: Related studies are reviewed in Section 2. A hybrid recommendation model and framework are proposed in Section 3. In Section 4 several experiments are conducted and evaluated. In Section 5, we draw our conclusions.

2. Related Work

CF-based RS recommends items to an active user based on the opinions of his/her like-minded neighbors [24]. According to modeling methods and recommended strategies of CF recommendation methods, their methods can be classified into two categories of neighbor- and model-based methods. For neighbor-based CF, all the ratings provided by users are kept in memory and used for prediction. To calculate the similarity between users/items, all the previously rated items are considered. For model-based CF, a model based on training data is obtained and the model is used to predict unknown ratings for real data.

2.1. Neighbor-Based Collaborative Filtering. The neighbor-based CF algorithm can be divided into two subcategories: user-based CF and item-based CF [25, 26]. The execution process of the neighbor-based CF recommendation algorithm can be divided into the following three steps [27, 28]: (a) calculate the similarity between an active user (or item) and other users (or items) through ratings on items from users; (b) select nearest neighbors for the active user (or item) according to the obtained similarity; (c) predict the ratings on the candidate items from the active user according to historical preference information of nearest neighbors, so as to produce recommended results. The rationales of user-based and item-based CF algorithms are shown in Figure 1.

2.1.1. User-Based CF Recommendation Algorithm. In general, the users and items are denoted as the vectors $u = \{u_1, u_2, \dots, u_N\}$ and $i = \{i_1, i_2, \dots, i_M\}$, respectively. The ratings on items from users are usually expressed as a rating matrix, and the ratings on items from a user can be denoted as a rating vector $R_u = \{R_{u1}, R_{u2}, \dots, R_{uN}\}$. For instance, the rating on item i_3 from user u_1 is 4 in Figure 1, which is denoted as $R_{13} = 4$. The similarity between two users is obtained by comparing the rating vectors of the two users. In RS, one of the most frequently used measures to calculate the similarities between users is cosine similarity.

Cosine similarity: the ratings on items from user u and user v are described as n -dimensional vectors, respectively, and the similarity between user u and user v is obtained through comparing the angles of the two rating vectors. The smaller the angle is, the higher the similarity value is. Cosine vector similarity measure is calculated as follows [24] (see (1)):

$$\text{sim}_{uv} = \frac{\vec{R}_u \cdot \vec{R}_v}{\|\vec{R}_u\|_2 \times \|\vec{R}_v\|_2} = \frac{\sum_{i \in O_{uv}} R_{ui} \cdot R_{vi}}{\sqrt{\sum_{i \in O_u} R_{ui}^2} \sqrt{\sum_{i \in O_v} R_{vi}^2}} \quad (1)$$

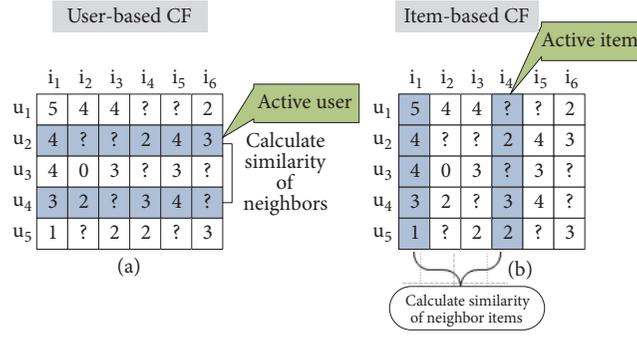


FIGURE 1: The rationales of user-based and item-based CF algorithms.

where sim_{uv} represents the similarity between user u and user v , \vec{R}_u and \vec{R}_v represent the rating vectors of u and v , respectively, and $\|\vec{R}_u\|_2$ and $\|\vec{R}_v\|_2$ represent the 2-norm of u and v , respectively, and R_{ui} and R_{vi} are the ratings on item i from u and v , respectively. O_u and O_v represent the sets of items rated by user u and user v , respectively. O_{uv} indicates the set of common items rated by both u and v .

According to [2, 24, 28], the predicted rating on item i from the active user u using the incremental weighted average is described as follows [2] (see (2)):

$$\hat{R}_{ui} = \bar{R}_u + \frac{\sum_{v \in N_u} \text{sim}_{uv} (R_{vi} - \bar{R}_v)}{\sum_{v \in N_u} |\text{sim}_{uv}|} \quad (2)$$

where \hat{R}_{ui} represents the predicted rating on item i from user u , N_u is the set of users that are most similar to active user u , \bar{R}_u and \bar{R}_v represent the average ratings on all rated items from user u and v , respectively, and sim_{uv} indicates the similarity between active user u and neighbor user v according to (1).

2.1.2. Item-Based CF Recommendation Algorithm. Similar to user-based CF recommendation algorithm, item-based CF recommendation algorithm predicts unknown ratings by using rated ratings on neighbor items from the active user. Pearson correlation coefficient (PCC) is one of the most frequently used measures to calculate the similarities between items, which is described as follows [14] (see (3)):

$$\text{sim}_{ij} = \frac{\sum_{u \in C_{ij}} (R_{ui} - \bar{R}_i) (R_{uj} - \bar{R}_j)}{\sqrt{\sum_{u \in C_{ij}} (R_{ui} - \bar{R}_i)^2} \sqrt{\sum_{u \in C_{ij}} (R_{uj} - \bar{R}_j)^2}} \quad (3)$$

where sim_{ij} denotes the similarity between item i and item j , R_{ui} and R_{uj} represent user u 's rating vectors for items i and j , respectively, and \bar{R}_i and \bar{R}_j represent the average ratings on items i and j in C_{ij} , respectively. C_{ij} denotes the set of users who rated both items i and j .

Ratings are predicted as follows [2] (see (4)):

$$\hat{R}_{ui} = \bar{R}_i + \frac{\sum_{j \in N_i} \text{sim}_{ij} \times (R_{uj} - \bar{R}_j)}{\sum_{j \in N_i} |\text{sim}_{ij}|} \quad (4)$$

where N_i is the set of items that are most similar to item i .

2.2. Model-Based CF Recommendation Technology. A recommendation algorithm should scale gracefully with the increase data. For the conventional neighbor-based CF methods, all the previously rated items are considered to compute the similarity between uses/items, it is time-consuming, and the methods are fail to achieve good scalability [18]. In order to reduce the computational complexity and improve the efficiency of recommendation without reducing the recommendation accuracy, subsequent clustering-based and classification-based techniques are introduced into RS to generate and evolve a variety of model-based recommendation algorithms, such as clustering models [29–33], singular value decomposition- (SVD-) based models [15, 18], and probabilistic matrix factorization- (PMF-) based models [34–37]. Model-based methods initially train a model based on training data to find patterns and then makes predictions for real data [38, 39].

2.2.1. Gaussian Mixture Model. Clustering analysis is an unsupervised classification approach for recognizing patterns, which is based on grouping similar objects together. In RS, similar users/items are classified into the same category by using clustering techniques, and then only the ratings of neighbor users/items in the same category are calculated to predict unrated items, which greatly reduce the computational effort. Generally, clustering is divided into hard and soft clustering. K-means is an important and well known hard clustering technique. Each object belongs to exactly one category in hard clustering, and there is no uncertainty in category membership of an object [29, 30]. In soft clustering, each object belongs to two or more categories with different degree of membership, rather than fully belonging one of the categories [31]. Gaussian mixture model (GMM) is a very well-known soft clustering technique. GMM is used in RS to more accurately discover a user's multiple interests and degree of preference on an item to better recommend items of interest to the user [8].

GMM assumes that the data obeys the mixture Gaussian distribution [40]. In other words, the data can be thought of as being generated from several Gaussian distributions. Each GMM consists of k Gaussian distributions. Among them, each distribution is called a "component", and these components are linearly added together to form the probability density function of GMM.

For a random vector x in n -dimensional sample space χ , if x obeys Gaussian distribution, the probability density function [34] is as follows (see (5)):

$$p(x) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)} \quad (5)$$

where μ is an n -dimensional mean vector and Σ is a covariance matrix of $n \times n$. It can be seen that the Gaussian distribution is completely determined by two parameters of the mean vector μ and the covariance matrix Σ .

Gaussian mixture distribution consists of k mixing components, each of which corresponds to a Gaussian distribution [37]. The mixture distribution is defined as follows (see (6)):

$$p_M(x) = \sum_{i=1}^k \alpha_i \cdot p(x | \mu_i, \Sigma_i) \quad (6)$$

where μ_i and Σ_i are the parameters of the i th Gaussian mixture and α_i is the corresponding mixing coefficient. Here, $\sum_{i=1}^k \alpha_i = 1$ and $\alpha_i > 0$.

2.2.2. Matrix Factorization. In RS, the most frequently used matrix factorization methods are singular value decomposition (SVD) [2, 15], latent factor model (LFM) [2, 39, 41], nonnegative matrix factorization (NMF) [23, 42–45], and trust-aware matrix factorization (TMF) [19, 20, 46–48], which are techniques of dimensionality reduction with implementation to RS. Reductions in dimensionality effectively preserve the information content while drastically decreasing the computation complexity and memory requirements for making recommendations [15, 39].

Latent Factor Model. The essence of matrix factorization is decomposing a very sparse rating matrix into two matrices: one represents the characteristics of the user and the other represents the features of the item [2]. The inner product of each row and each column of two matrices yields the corresponding rating.

A user-item rating matrix denoted by $R_{N \times M}$ can be decomposed into the product of two matrices $U_{N \times f}$ and $V_{f \times M}$, which is expressed as follows (see (7)):

$$R_{n \times m} \approx U_{N \times f} \times V_{f \times M} \quad (7)$$

where $U_{N \times f}$ represents the relationships between N users and f topics and $V_{f \times M}$ represents the relationships between f topics and M items. f indicates the dimensions of the latent factors.

In RS, the recommendation algorithm based on matrix factorization is implemented in two processes: (1) The original matrix is decomposed into two low-rank matrices by using known ratings in (7). The decomposition of the original matrix is the process of model training, and model parameters are usually obtained by using the stochastic gradient descent method for the established loss function. (2) Predict the unknown ratings by using the inner product of the obtained low-rank matrices U and V , i.e., $\hat{R}_{N \times M} = U_{N \times f} \times V_{f \times M}$.

Therefore, the key task of recommendation algorithm based on matrix factorization is to solve $U_{N \times f}$ and $V_{f \times M}$, which can be transformed into a regression problem. A loss function, namely, L , which represents the sum of squares of the errors between the original ratings and the predicted ratings, is defined as follows [1, 2, 21] (see (8)):

$$L = \sum_{i=1}^N \sum_{j=1}^M e_{ij}^2 = \sum_{i=1}^N \sum_{j=1}^M \left(R_{ij} - \sum_{k=1}^f U_{ik} V_{kj} \right)^2 \quad (8)$$

where $e_{ij} = R_{ij} - \hat{R}_{ij}$. U_{ik} represents the k th feature of user i and V_{kj} represents the k th feature of item j . The loss function after adding the regularization term is defined as follows (see (9)):

$$L' = \sum_{i=1}^N \sum_{j=1}^M \left(R_{ij} - \sum_{k=1}^f U_{ik} V_{kj} \right)^2 + \frac{\lambda_U}{2} \sum_{i=1}^N U_{ik}^2 + \frac{\lambda_V}{2} \sum_{j=1}^M V_{kj}^2 \quad (9)$$

In general, two regularization terms λ_U and λ_V are set to be the same for ease of calculation, i.e., $\lambda = \lambda_U = \lambda_V$. Then the stochastic gradient descent method is used to solve (9) (see (10)-(11)):

$$\begin{aligned} \frac{\partial L'}{\partial u_{ik}} &= -2 \left(R_{ij} - \sum_{k=1}^f U_{ik} V_{kj} \right) V_{kj} + \lambda U_{ik} \\ &= -2e_{ij} V_{kj} + \lambda U_{ik} \end{aligned} \quad (10)$$

$$\begin{aligned} \frac{\partial L'}{\partial v_{kj}} &= -2 \left(R_{ij} - \sum_{k=1}^f U_{ik} V_{kj} \right) U_{ik} + \lambda V_{kj} \\ &= -2e_{ij} U_{ik} + \lambda V_{kj} \end{aligned} \quad (11)$$

Update variable U'_{ik} and V'_{kj} according to the negative gradient as follows (see (12)-(13)):

$$U'_{ik} = U_{ik} - \alpha \frac{\partial L'}{\partial U_{ik}} = U_{ik} + \alpha (2e_{ij} V_{kj} - \lambda U_{ik}) \quad (12)$$

$$V'_{kj} = V_{kj} - \alpha \frac{\partial L'}{\partial V_{kj}} = V_{kj} + \alpha (2e_{ij} U_{ik} - \lambda V_{kj}) \quad (13)$$

where α denotes learning rate, λ is the regularization term parameter, and e_{ij} represents the difference between the real rating and the predicted rating.

Singular Value Decomposition. SVD is a powerful computation tool in latent semantic analysis which results in downsizing of relevant variables and finding a good approximation of R [15]. The purpose of SVD in RS is to compress a very high dimensional data into a low-dimensional space, while preserving the major information content. A matrix $R_{N \times M}$ with the rank of $r \leq \min(N, M)$ can be decomposed as follows [2, 15] (see (14)):

$$R = P \Sigma Q^T \quad (14)$$

where $P^T P = E$ and $Q^T Q = E$. E represents a unit square matrix. The rows of P and Q indicate users' features

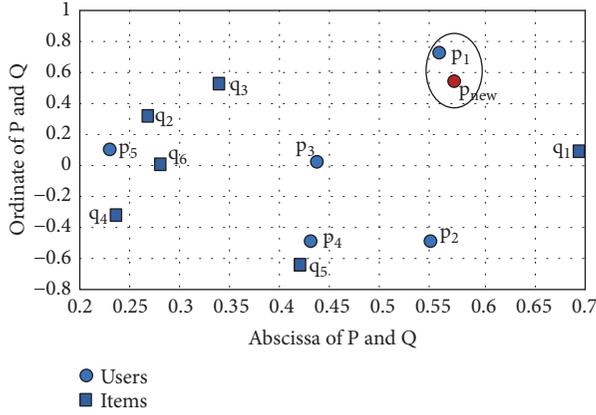


FIGURE 2: Two-dimensional space of applying SVD for users, items, and new user.

and items' features, namely, left-singular vectors and right-singular vectors, respectively, and their values correspond the eigenvectors of RR^T and R^TR . Σ is a $r \times r$ diagonal matrix, which represents the degree of association between P and Q. The diagonal values, namely, the singular values are arranged from large to small, each of which is indeed the square root of the eigenvalue of RR^T (or R^TR). Moreover, the eigenvectors of RR^T and R^TR , i.e., rows of P and Q, are arranged according to the eigenvalues.

To benefit from dimensionality reduction, we put $f < r$ and include only the f largest singular values in Σ and replace the others by zero. Therefore, R is described approximately with \hat{R} as follows (see (15)):

$$R \approx \hat{R} = P_{N \times f} \hat{\Sigma}_{f \times f} Q_{M \times f}^T \quad (15)$$

where $\hat{\Sigma}$ is the f -rank approximation of Σ . The matrices $P_{N \times f}$ and $Q_{M \times f}$, respectively, contain the f largest eigenvectors of RR^T and R^TR , whereas the diagonal matrix $\hat{\Sigma}_{f \times f}$ contains the nonnegative square roots of the f largest eigenvalues of either matrix along its diagonal [20, 46]. For finding the positions of users and items, we can map raw data to the p dimensions space as follows [17] (see (16)-(17)):

$$P_{\text{Trans}} = R_{N \times M} \times Q_{M \times f} \times \Sigma_{f \times f}^{-1} \quad (16)$$

$$Q_{\text{Trans}} = R_{N \times M}^T \times P_{N \times f} \times \Sigma_{f \times f}^{-1} \quad (17)$$

where P_{Trans} and Q_{Trans} are new positions of users and items in the f dimensions space, respectively.

For instance, the matrix in Figure 1(a) is denoted as A, which can be decomposed into P, Q, and Σ . The approximation of A can be obtained by taking the first 2-dimensional

data, i.e., $P' = \begin{pmatrix} 0.557 & 0.733 \\ 0.503 & -0.475 \\ 0.439 & 0.020 \\ 0.4356 & -0.472 \\ 0.233 & 0.118 \end{pmatrix}$, $Q' = \begin{pmatrix} 0.695 & 0.095 \\ 0.266 & 0.345 \\ 0.344 & 0.559 \\ 0.239 & -0.369 \\ 0.435 & -0.646 \\ 0.285 & 0.068 \end{pmatrix}$, $\Sigma' =$

$\begin{pmatrix} 11.65 & 0 \\ 0 & 5.767 \end{pmatrix}$. P' and Q' are projected in 2-dimensional space and plotted in Figure 2.

When a new user p_{new} with a shared rating of [5, 3, 4, 0, 1, 2] arrives, the following calculation is performed to find

the position of the new user in the 2-dimensional space as follows:

$$p_{\text{new}} = [5, 3, 4, 0, 1, 2]^T \times Q' \times \Sigma'^{-1} = [0.572, 0.561] \quad (18)$$

As can be seen in Figure 2, it can be found that the user p_1 is close to the new user. Therefore, p_1 is considered to be the nearest neighbor of the new user, and the similarity of the approximate low-dimensional data can be calculated to make a recommendation.

Classic matrix factorization [19, 41] methods can provide very accurate predictions and have the advantages of high computational efficiency and expansibility. However, classic matrix factorization algorithms would make unreliable predictions when encountering new users/items and extremely sparse data. To solve the problems mentioned above, social relationships between users are introduced into RS, and trust-based CF (TCF) [23, 37, 45, 49, 50] algorithms arise in recent years. TCF algorithms integrate trust relationships between users into recommendation algorithms to improve the quality of the recommendation [11, 13, 18, 24, 29, 30]. These methods are also divided into neighbor-based TCF (NTCF) and model-based TCF (MTCF). NTCF is not scalable, so the computational cost is extremely high when faced with large amounts of data. Therefore, NTCF is not suitable when dealing with a great number of users and items. MTCF incorporates social network information into the matrix factorization model, to optimize the parameters by using the user-item ratings and individual trust among users. Thus it is scalable and also makes an accurate recommendation [15, 18, 37, 41, 49, 50]. Therefore, MTCF has been drawing considerable attention recently [19, 51]. Many of the current recommendation methods are based on the idea of MTCF.

3. The Proposed Hybrid Recommendation Method and Framework

In this section, an overview of the proposed hybrid recommendation method is described in Section 3.1. Second, an improved CF recommendation method is described in Section 3.2 and predicting the initial ratings using EM algorithm is discussed in Section 3.3. Third, the calculation method of user's social status in trust networks is discussed and an enhanced social matrix factorization model is proposed in Section 3.4. Finally, computational complexity of the hybrid method in this paper is analyzed in Section 3.5.

3.1. Overview. In the case of sparse data, in order to improve the effect of user clustering and the accuracy of predicted ratings, we present a hybrid recommendation method based on Gaussian mixture model and enhanced social matrix factorization algorithm (GMMESMF) as Figure 3.

The prediction process is described as follows: (1) The similarities of users and items are calculated using the improved user-based and item-based measures. Here, the user's similarity calculation method takes into account factors such as the trust relationships between users, the number of users' common interactions with the same item, and the users' evaluation times for the items, to reduce the influence

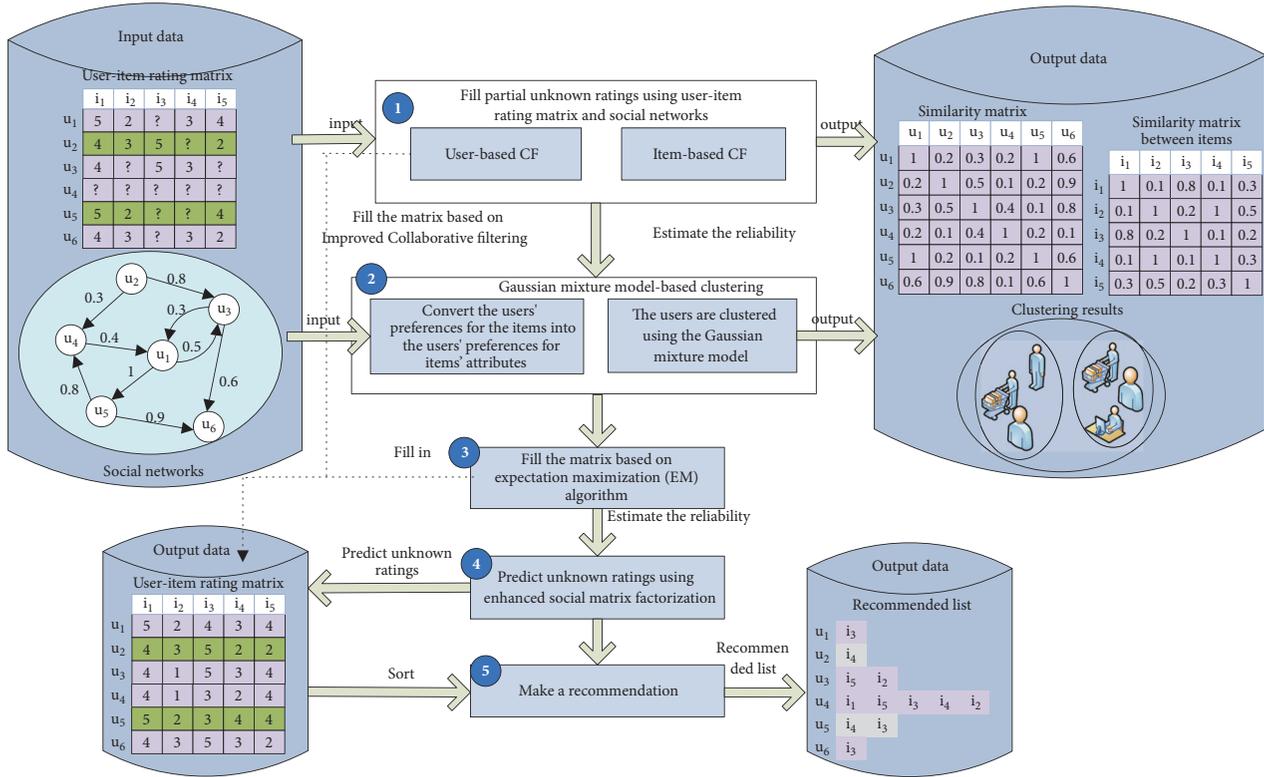


FIGURE 3: A hybrid framework of RS.

of data sparsity on the performance of CF algorithm. Then partial unrated items are predicted using user-based and item-based CF methods. (2) The users' preferences on the items are converted into the users' preferences on items' attributes. Then, the users are clustered using the Gaussian mixture model. In this way, the users in the same cluster have the same interests. This can speed up the search for nearest neighbors and increase the speed of recommendation generation. (3) We then predict some unrated items using expectation maximization (EM) algorithm to calculate the probability that each user belongs to each cluster. (4) After filling the user-item rating matrix in the first two steps, we can obtain a matrix that is not very sparse and reliable. The trust relationships between users and trust propagation are introduced into matrix factorization model, and an enhanced social matrix factorization technique is proposed to learn more accurate user characteristics and item characteristics in each cluster, so that the prediction for unrated items is completed; meanwhile, the prediction accuracy can be greatly improved. The density of the user-item rating matrix of each cluster is usually denser than the original user-item rating matrix when the number of users in each cluster is relative small. (5) According to the result of the prediction, the rating results are sorted in descending order, and the candidate items are recommended to the active user.

Through the above step-by-step filling method, partial unrated items can be filled according to the user-based and item-based collaborative filtering algorithms using their neighbor relationships. If the unknown ratings are predicted

by matrix decomposition techniques directly, the predicted ratings would be inaccurate due to too few rated items for reference. To obtain reliable ratings, we estimate the reliability of filling ratings by using the reliability measure and then predict all unknown ratings using the matrix factorization algorithm.

3.2. Fill the Rating Matrix by Improved Collaborative Filtering Recommendation Algorithm. CF algorithm collects users' preferences from the user-item rating matrix and produces the recommendations only based on the opinions that users' interests are similar to current active user. However, the rating matrix is usually very sparse, when there are few neighbor users highly related to the active user, the quality of predicted ratings would be seriously affected, resulting in poor quality recommendation [28]. However, CF is one of the most successful and effective recommendation algorithms. Therefore, we use CF algorithm to predict some unrated items to reduce the sparsity of the rating matrix.

Cosine similarity and PCC are common methods of calculating similarity between two users and between two items, respectively, which directly affect the prediction accuracy. When two rating vectors given by two users are equal ratio, such as (4, 4, 4) and (5, 5, 5), (1, 2, 1), and (2.5, 5, 2.5), as Table 1, the similarities calculated by cosine and PCC will be both 1.0, although the aforementioned rating pairs are quite different.

In order to avoid the above problem and improve the similarity between the users or the items, inspired by

TABLE 1: An example of user-item rating matrix.

User	Item 1	Item 2	Item 3	Item 4	Item 5
u1	2.5	—	5	2.5	—
u2	—	3	—	3	3
u3	1	—	2	1	—
u4	—	5	—	5	5
u5	1.5	—	3	1.5	—
u6	—	4	—	4	4

[3, 4, 14, 18, 49, 52–54], the current method of similarity calculation needs to be improved.

Definition 1. Mean User Reference Center (MURC). The mean user reference center, i.e., average rating on items by users, denotes the average rating of all items from all users.

$$\text{MURC} = \frac{\sum_{u=1}^N \left(\sum_{i=1}^M R_{ui} / \text{Card}(O) \right)}{\text{Card}(C)} \quad (19)$$

where MURC represents the center point of user preference and C and O represent the sets of all users and all items, respectively. Card(C) and Card(O) represent the numbers of C and O, respectively.

Definition 2. Average Similarity Standard Deviation (ASSD). The average similarity standard deviation refers to the distance of the similarity values of all the other users from the mean user reference center. It reflects the average distribution of the average ratings on all items from each user.

$$\text{ASSD} = \frac{\sum_{u \in C} |\bar{R}_u - \text{MURC}|}{\text{Card}(C)} \quad (20)$$

where \bar{R}_u denotes the average rating on all items from user u.

Definition 3 (Basic Similarity Region (BSR)). The basic similarity region is described as the user preference similarity distribution range whose center is the mean user reference center and whose radius is the average similarity standard deviation. If $\text{sim}_{uv} \in [\text{MURC} - \text{ASSD}, \text{MURC} + \text{ASSD}]$, then $\text{sim}_{uv} \in \text{BSR}$; otherwise, $\text{sim}_{uv} \notin \text{BSR}$. BSR denotes the basic similarity region. sim_{uv} is calculated in the coordinate of the dimensionality reduction by (16).

Inspired by [4, 14, 26, 27, 53], we incorporate salton factors and trust relationships into cosine similarity calculation between users. The improved similarity calculation method is as follows (see (21)):

$$s(u, v) = \begin{cases} w_1 \cdot w_2 \cdot \text{sim}_{uv}, & \text{sim}_{uv} \in \text{BSR} \\ \frac{\bar{R}_u}{\bar{R}_v} \cdot w_1 \cdot w_2 \cdot \text{sim}_{uv}, & \text{others and } \bar{R}_u \leq R_v \end{cases} \quad (21)$$

This will determine whether the users' preference behaviors need to be revised. Here sim_{uv} represents the similarity

between u and v, which are filtered out the noisy data. When the similarity between u and v is not in BSR, the similarity is considered to be a noisy data. Cosine similarity in (1) only considers the angle between two vectors but does not consider the length of the vector, so it needs to be revised according to (21).

Here, w_1 and w_2 represent the salton and time attenuation factors, respectively. Among them, $w_1 = (|O_{uv}| + \gamma) / (|\bar{O}_{uv}| + \gamma)$, and O_{uv} denotes the set of common items rated by both u and v. \bar{O}_{uv} denotes the set of items rated by any other two users. γ is a constant parameter, which aims at avoiding the case where the denominator in the formula is zero or w_1 is zero. The difference between u and v's rating times for the same item means that their interest changes are not synchronized. Therefore, it is necessary to introduce the time attenuation factor w_2 to weight the similarity between u and v; i.e., it is necessary to reduce the similarity when the rating times of two users are far apart. $w_2 = (\sum_{i \in O_{uv}} (1 / (1 + \exp(\omega |t_{ui} - t_{vi}|)))) / |O_{uv}|$, where ω denotes the parameter of time attenuation and t_{ui} and t_{vi} denote the times of the item i rated by users u and v, respectively.

The improved similarity between user u and user v is described as follows (see (22)):

$$\text{sim}(u, v) = \varphi T_{uv} + (1 - \varphi) s(u, v) \quad (22)$$

where T_{uv} denotes the similarity between users u and v, which can be calculated as follows [11, 30] (see (23)):

$$T_{uv} = \frac{d_{\max} - d_{uv} + 1}{d_{\max}} \quad (23)$$

where d_{\max} denotes the maximum allowable propagation distance among users and d_{uv} denotes the trust statement between users u and v. In this section, the dimensions of data first are reduced as(16) and (17) before calculating the similarity. To ensure the accuracy of the predictions, a reliability measure in [7, 24, 55] is employed to evaluate the quality by providing a feedback on the quality of the predicted rates. The reliability measure is calculated as follows [7, 24] (see (24)):

$$R_{ui} = \frac{|K_{ui}|}{3\sqrt{1 + D_{ui}}} \quad (24)$$

where R_{ui} is the reliability of a prediction \hat{R}_{ui} . K_{ui} is the number of neighbors of u who have rated the item i. D_{ui} is the variance of the ratings made by the neighbors of u over

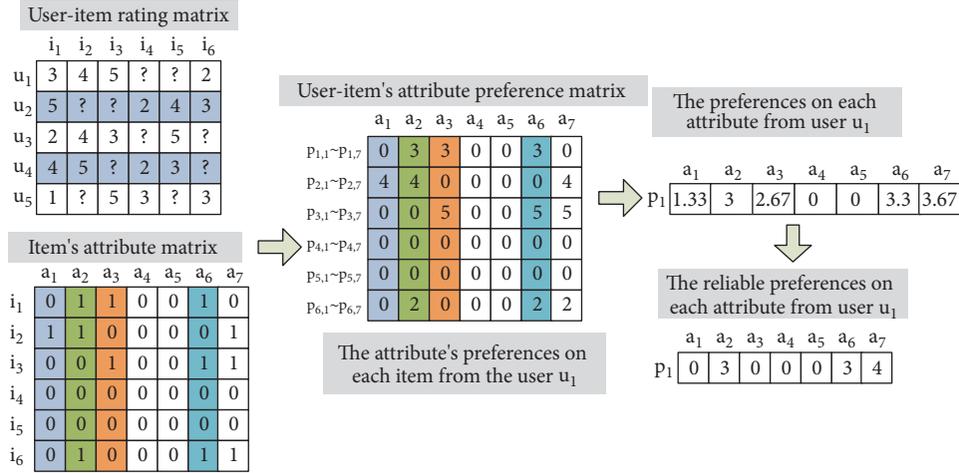


FIGURE 4: The transition from the user-item ratings to user-item' attribute preferences.

the item i . The value range of R_{ui} is $[0, 1]$. The larger the value of R_{ui} is, the higher the accuracy of the prediction is, and vice versa. If the reliability value of R_{ui} is less than a threshold value ψ ($\psi = 0.85$ in this paper), the prediction rating \hat{r}_{ui} would be set to 0.

3.3. Predict Ratings Using EM Algorithm

3.3.1. Covert Ratings on Items from Users into Preferences on Items' Attributions from Users. In the case of very sparse user-item rating matrix, in order to form a group of users with the same interest, ratings on items from users are converted into users' preferences on items' attributes, and then the attributes are clustered to get the users' interest groups by using Gaussian mixture model, and finally unknown ratings are predicted by using the neighbor-based method.

Definition 4. A user's preferences on an item's attributes reflect the user's interests on the item. The relation matrix between the users and the attributes of the items can be expressed as the product between the user-item rating matrix and the item' attribute matrix.

$$R^{(ua)} = R^{(ui)} \cdot R^{(ia)} \quad (25)$$

where $R^{(ua)}$, $R^{(ui)}$, and $R^{(ia)}$ denote the user-item's attribute matrix, the user-item rating matrix, and the item's attribute matrix, respectively. \cdot denotes the product of the two matrices, but it is not simply a multiplication of two matrices.

If a user likes an item, it is because the user is interested in the attributes of the item. Under this assumption, we can obtain the preferences on items from users by using the ratings in the user-item rating matrix and item-attribute information in item-attribute matrix. Specifically, the preference of the user u on the attribute a of the item i can be described as follows (see (26)):

$$p_{ua} = R_{ui} A_{ia} \quad (26)$$

where p_{ua} denotes the preference of user u on attribute a , R_{ui} denotes the rating of the user on the item i , and A_{ia} denotes the attribute a of the item i . Here, A_{ia} is defined as follows (see (27)):

$$A_{ia} = \begin{cases} 1, & i \text{ has the attribute of } a. \\ 0, & \text{otherwise.} \end{cases} \quad (27)$$

The transition from the user-item ratings to the user-item's attribute preferences are showed in Figure 4. For instance, the rating R_{11} on i_1 from u_1 is 3 and the item i_1 has the attributes of a_2 , a_3 , and a_6 . Thus, the preference values on a_2 , a_3 , and a_6 from u_1 are all 3.

In general, the more a user's ratings on an item's attribute is, the more reliable his/her evaluation on the item's attributes is. The lower the preference value is, the less reliable the user's attribute rating on the item is. Therefore, these ratings can be removed. Considering the reliability of the user's preferences on the item's attributes, the preference values against each item are weighted average as follows (see (28)):

$$p'_{ua} = \frac{\sum_{i \in O_u} A_{ia}}{\sum_{i \in O_u} \sum_{j=1}^s A_{ij}} \sum_{k=1}^n p_{ua}^{(k)} \quad (28)$$

where p'_{ua} denotes the reliable preference of the user u on the attribute a and O_u denotes the set of items rated by user u . $p_{ua}^{(k)}$ denotes the k th preference value on attribute a from user u according to rating on the k th item from the user u , and s represents the number of the item's attribute. When the preference values of items' attributes from u_1 are obtained, a very sparse matrix is mapped to a low-dimensional space; thus the sparsity is also alleviated.

For instance, according to the rating R_{11} on item i_1 from the user u_1 and the attributes of the item i_1 , the preference values on a_2 , a_3 , and a_6 from user u_1 can be obtained using (26), which are 3, 3, and 3, respectively, and the other preference values on a_1 , a_4 , a_5 , and a_7 are all zero. That is, $p_{12}^{(1)} = R_{11} A_{12} = 3 * 1 = 3$, $p_{13}^{(1)} = R_{11} A_{13} = 3 * 1 = 3$, $p_{16}^{(1)} = R_{11} A_{16} = 3 * 1 = 3$, and the other preference values are all zero.

$1 = 3$, and $p_{16}^{(1)} = R_{11}A_{16} = 3 * 1 = 3$. The obtained attribute preference values are denoted by $p_1^{(1)} = (p_{11}^{(1)}, p_{12}^{(1)}, \dots, p_{17}^{(1)}) = (0, 3, 3, 0, 0, 3, 0)$. Likewise, according to the ratings R_{12}, R_{13}, \dots , and R_{16} on items i_2, i_3, \dots , and i_6 from u_1 , we can obtain the preference values on a_1, a_2, \dots , and a_7 from the user u_1 , which are denoted by $p_1^{(2)}, p_1^{(3)}, \dots, p_1^{(6)}$. For example, $p_1^{(2)} = (p_{11}^{(2)}, p_{12}^{(2)}, \dots, p_{17}^{(2)}) = (4, 4, 0, 0, 0, 4)$, and $p_1^{(6)} = (p_{11}^{(6)}, p_{12}^{(6)}, \dots, p_{17}^{(6)}) = (0, 2, 0, 0, 0, 2, 2)$. Therefore, we can obtain the weighted attribute's preference values from user u_1 according to (28); i.e., $p_1 = (p_{11}, p_{12}, \dots, p_{17}) = (1.33, 3, 2.67, 0, 0, 3.3, 3.67)$. For example, $p_{11} = (4/12) \sum_{k=1}^6 p_{11}^{(k)} = 1.33$, and $p_{12} = (4/12) \sum_{k=1}^6 p_{12}^{(k)} = 3$. The unreliable attribute's preference values are removed, and then the reliable attribute's preference from u_1 is obtained as shown in Figure 4.

3.3.2. Predict the Partial Ratings Using Gaussian Mixture Model. According to [8, 10, 56], whether a user is interested in an item depends on the nature of the item. For example, whether a user likes a movie may depends on three factors: whether it is an entertainment or a literary film, a foreign language, or a Chinese language film and whether the actor is famous. This article uses the Gaussian mixture model proposed by Hofmann as the basis of clustering and supposes that the conditional probability of the rating on an item's attribute a by the group z obeys the Gaussian distribution [2, 40] (see (29)).

$$p(r | u, a) = \sum_{z_k \in Z} p(z_k | u) p(r | \mu_{a,z_k}, \Sigma_{a,z_k}) \quad (29)$$

where $p(r | u, a)$ denotes the joint probability of user u and item's attribute a . z_k is a potential group that has not been observed. $p(z_k | u)$ denotes the probability that the user u belongs to the group z_k . For each k , $p(z_k | u)$ represents the probability that user u belongs to the group z , and $\sum_{k=1}^K z_k = 1$. $p(r | \mu_{a,z_k}, \Sigma_{a,z_k})$ indicates that the rating of each user in group z obeys a Gaussian distribution, which represents the conditional probability of the ratings on the attribute a from the group z .

According to the Bayesian formula, the joint probability density formula can be obtained as follows (see (30)):

$$\begin{aligned} p(u, a, r) &= p(u) p(a) p(r | u, a) \\ &= p(u) p(a) \sum_{z_k \in Z} p(z_k | u) p(r | \mu_{a,z_k}, \Sigma_{a,z_k}) \end{aligned} \quad (30)$$

where $p(u)$ and $p(a)$ are both constants. Therefore, the joint probability density function can be written in the following form (see (31)):

$$p(u, a, r, z | \theta) \propto \sum_{z_k \in Z} p(z_k | u) p(r | \mu_{a,z_k}, \Sigma_{a,z_k}) \quad (31)$$

where θ represents the general name of all parameters. Log-likelihood function is described as follows (see (32)):

$$L(\theta) = \log \sum_{z_k \in Z} p(z_k | u) p(r | \mu_{a,z_k}, \Sigma_{a,z_k}) \quad (32)$$

Then we use the parameter θ to optimize the function.

To obtain the values of each parameter ($k, p(z_k | u), \mu_{a,z}, \Sigma_{a,z}$), we use EM algorithm to alternately execute step E and step M to solve the parameters as follows [10, 56] (see (33)-(36)).

E-Step:

$$p(z_k | u, a, r) = \frac{p(z_k | u) p(r | \mu_{a,z}, \Sigma_{a,z})}{\sum_{z_k \in Z} p(r | \mu_{a,z}, \Sigma_{a,z}) p(z_k | u)} \quad (33)$$

M-Step:

$$p(z_k | u) = \frac{\sum_{\{u', a, r\}: u'=u} p(z_k | u, a, r)}{\sum_{z_k \in Z} \sum_{\{u', a, r\}: u' \neq u} p(z' | u, a, r)} \quad (34)$$

$$\mu_{a,z} = \frac{\sum_{\{r', a, r\}: a'=a} r p(z_k | u, a, r)}{\sum_{\{u', a, r\}: a'=a} p(z_k | u, a, r)} \quad (35)$$

$$\Sigma_{a,z} = \frac{\sum_{\{u', a, r\}: a'=a} (r - \mu_{a,z})^2 p(z | u, a, r)}{\sum_{\{u', a, r\}: a'=a} p(z | u, a, r)} \quad (36)$$

Execute E and M steps alternately, until $L(\theta) - L(\theta)'$ which is less than a given value of ϵ converges, the iteration ends, and the model parameters are obtained.

Finally, the predicted rating on attribute a from user u is as follows [10] (see (37)):

$$\begin{aligned} R_{ua} &= \int_r r p(r | u, a) dr \\ &= \sum_{k=1}^K p(z_k | u) \int_r r p(r | a, z_k) dr \\ &= \sum_{k=1}^K p(z_k | u) \mu_{ka} \end{aligned} \quad (37)$$

where μ_{ka} denotes group-attribute average value. In a similar way, we can predict the unrated items by using the mixture model.

Next, an example is used to illustrate how to predict user ratings using a Gaussian mixture model as Figure 5. A user-item rating matrix contains 80 ratings on a scale from 1 to 5 of 6 movies by 20 users, and each movie has several different genres, e.g., comedy, crime, action, and romance. The ratings on movies from users are converted to users' preferences on movies' attributes according to (28). Then three attributes are selected as features to train the data; suppose that each user has multiple different interests and three groups are divided on the basis of Gaussian mixture model. In Figure 5, different colors represent different groups, respectively. If we will predict the rating $R_{11,2}$, i.e., rating on item i_2 from user u_{11} in the user-item rating matrix. According to (33), we can obtain the probabilities that user u_{11} belongs to group one, group two, and group three and is denoted as z_1, z_2 , and z_3 . The probabilities are 1.169e-72, 0.133, and 0.867. The first probability is close to zero, and it is ignored for ease of calculation. Therefore, $R_{11,2} = p_1 \mu_{1,2} + p_2 \mu_{2,2} + p_3 \mu_{3,2} = 0 + 0.133 \times 3.67 + 0.867 \times 4.75 = 4.67$.

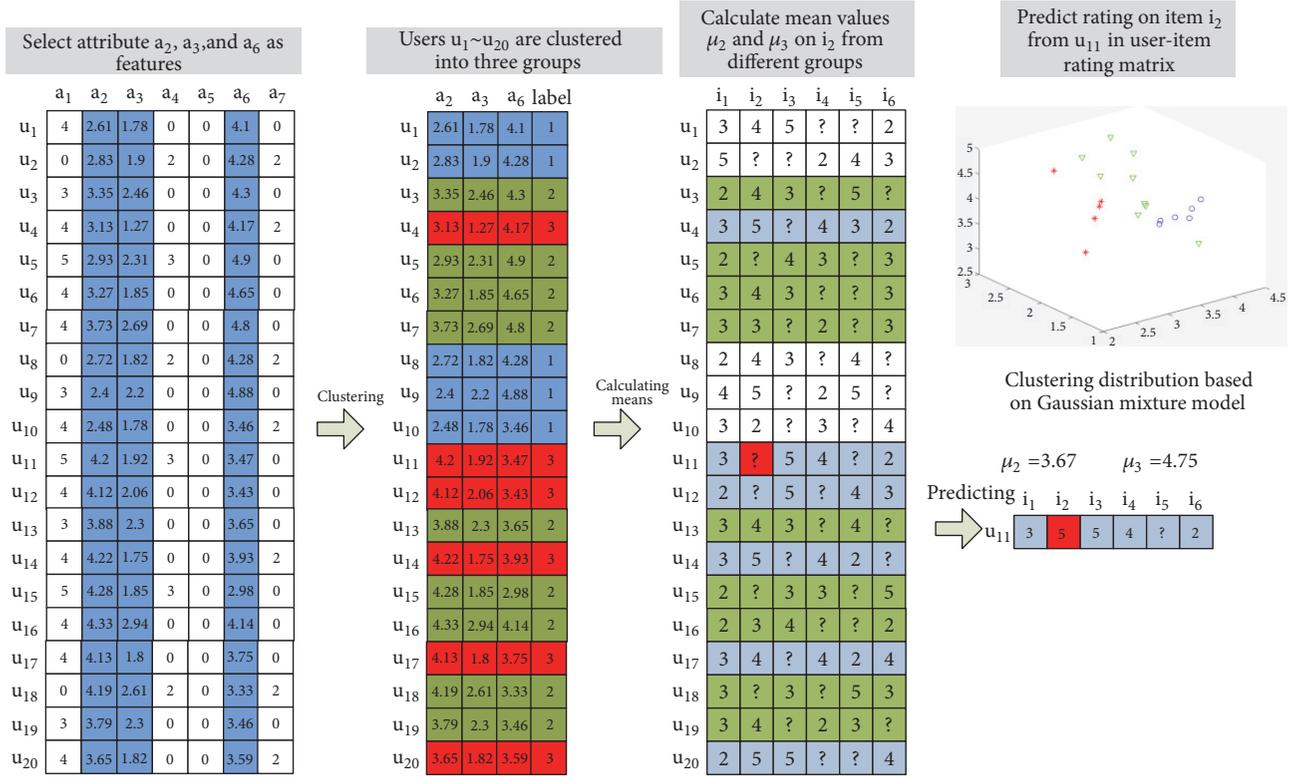


FIGURE 5: The transfer from user-item rating matrix to preference for user-item attributes.

3.3.3. Reliability of Ratings Prediction. Until now, initial unknown ratings are predicted in the mixture Gaussian model. However, sometimes the predicted ratings as the above method are not very accurate, especially when an item is rarely rated by users. Therefore, these predicted rating data is unreliable, and we will remove the reliable data from the user-item rating matrix.

Definition 5 (reliability of rating prediction (RRP)). In the each component of the mixture Gaussian model, when the proportion of the number of rated item in the number of the current component is less than ε ($\varepsilon \leq 0.1$ in this paper), the predicted ratings on these items are unreliable. The definition is as follows (see. (38)):

$$R_{ui} \notin \text{RRP}, \quad \frac{\sum_{u,v \in z_k} |R_{vi}|}{|z_k|} < \varepsilon$$

$$R_{ui} \in \text{RRP}, \quad \frac{\sum_{u,v \in z_k} |R_{vi}|}{|z_k|} \geq \varepsilon \quad (38)$$

For $R_{ui} \notin \text{RRP}$, we will set the ratings to zero; in other words, they are considered as unknown ratings, so that they are predicted by the matrix factorization method in Section 3.4.

The EM algorithm of initial ratings prediction is described in Algorithm 1.

3.4. Complete Matrix with Matrix Factorization Algorithm. In Section 3.3.2, Gaussian mixture model is used to predict

unknown ratings from user's perspective. However, when some users have few or no ratings on items, it is difficult to predict unknown ratings according to nearest neighbors and the ratings obtained are inaccurate. In this section, inspired by [13, 18, 19, 21, 22, 46, 51], an enhanced social matrix factorization model fusing user's social status and item's similarity is proposed, which is called ESME, to predict all unrated items. The graphical model is shown in Figure 6. The enhanced social matrix factorization model uses the rated items and trust relationships between users to train model parameters and predicts unknown ratings by using the trained model parameters. It avoids the problem of inaccurate prediction because some users have few ratings on items and some items are only rated by few users.

3.4.1. Motivation. To precisely capture user latent feature and item latent feature, a novel enhanced social matrix factorization method is proposed. The method uses the individual trust with social status and item's social relationship to optimize the solution in both user latent feature space and user-item rating space, which is our main contribution in this section. Firstly, to accurately reflect the impact of users with different social status on user decisions, a regularization term is added in (9) to minimize the differences among the latent feature of trusted users. In addition, inspired by the user's social relationship, an item social relationship matrix S is constructed, which is used to improve the item's latent feature vector V . This is because there is also a relationship between the items. In real life, when people buy products, they

Input: Ratings in training set R , Gaussian mixture distribution parameter posterior probability p, μ, Σ , and k .
Output: The probabilities of cluster partition $G = \{G_1, G_2, \dots, G_k\}$ and estimated matrix \hat{R} .

- 1 Convert ratings into preferences of item attributes based on Eq. (28).
- 2 Initialize model parameters: α, μ, Σ , and k .
- 3 **repeat**
- 4 **for** $j = 1, 2, \dots, m$ do
- 5 Calculate the posterior probability that user u_j belongs to group z_k based on Eq. (32).
- 6 **end for**
- 7 **for** $i = 1, 2, \dots, k$ do
- 8 Calculate new posterior probabilities based on Eq. (33).
- 9 Calculate new mean vectors based on Eq. (34).
- 10 Calculate new covariance matrices based on Eq. (35).
- 11 **end for**
- 12 update model parameters p, μ, Σ .
- 13 **until** convergence
- 14 Mark cluster label probability according to $\lambda_j = \operatorname{argmax} p(z_k = i | u_j)$
- 15 Predict unknown ratings based on $R_{ui} = \sum_{k=1}^K p(z_k | u) \mu_{ki}$.
- 16 Reset the unreliable ratings to zeroes based on Eq. (38).
- 17 **return** \hat{R} .

ALGORITHM 1: The EM algorithm of initial ratings prediction.

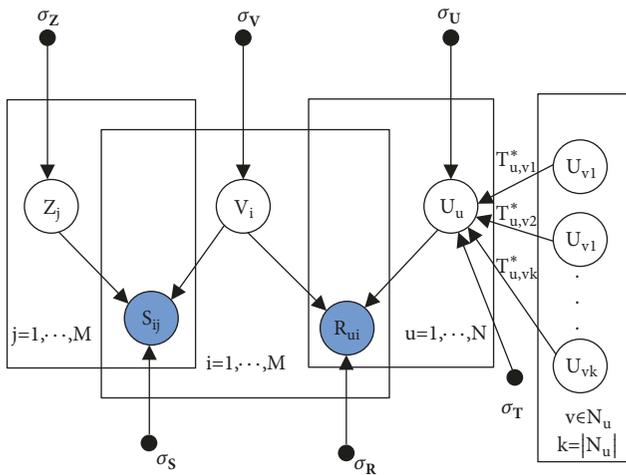


FIGURE 6: Graphical model for enhanced social matrix factorization model (ESMF).

will consider similar or substitute products in many cases. Based on this consideration, the item's social relationship is introduced into the matrix factorization model. Therefore, the proposed approach accurately and realistically models real-world recommendation processes.

3.4.2. Calculate the Social Status of Users in Trust Networks.

The user's social status is a very important concept, which reflects the importance of a user in social networks and the degree of an individual attachment to other individuals in social networks. The social status theory is used to explain how the user's social level influences the establishment of trust relationships between users. Usually, in the social network, high-level users usually belong to the authority users, and low-level users are more likely to establish trust

relationships with their higher-level users. For instance, in a social network, user v is an authoritative scholar in terms of historical research but is a beginner in terms of computer technology. Therefore, the user u will accept suggestion of the user v when selecting historical books but will not accept suggestion of the user v when choosing computer books.

In social networks, users with higher social status usually provide valuable information to users with lower social status; therefore, they have a lot of in-degrees and out-degrees. In fact, users with lower social status usually refer to the suggestions of users with higher social status, and thus they have more out-degrees [56]. In this study, we employ the PageRank algorithm to calculate the social status of each user in social networks as follows [17, 56, 57] (see (39)):

$$PR_u = \eta \sum_{v \in T(u)} \frac{PR_v}{|T(u)|} + \frac{1}{N} (1 - \eta) \quad (39)$$

where PR_u denotes the value of the user u 's PageRank and $T(u)$ denotes the set of the user u who trusts friends. N represents the number of users, and η is the probability of jumping out of the currently trust network, which is range of $[0, 1]$.

Definition 6. The trust relationships with social status between users u and v . The higher a person's social status in a field is, the greater the influence is, and the more likelihood that others will accept his suggestion. In addition, the number of common ratings is considered as interaction relationship factor between two users. The more the number of common rating items is, the more similar the rating is, and the closer their interests get. A trust network can be constructed based on the combination of the trust statements and the similarities between users. Therefore, the trust statement with social status between users and common interests of users is defined as follows (see (40)):

$$T_{uv}^* = \begin{cases} \frac{|I_{uv}|}{|I_u| + |I_v|} * \text{sim}_{uv} * PR_v * t_{uv} & PR_v > PR_u \\ \frac{|I_{uv}|}{|I_u| + |I_v|} * \text{sim}_{uv} * \frac{PR_v}{PR_u} * t_{uv} & \text{otherwise} \end{cases} \quad (40)$$

where I_u and I_v denote the sets of items rated by users u , v , respectively, and I_{uv} denotes the set of common items rated by users u and v . sim_{uv} denotes the similarity between users u and v that is calculated using (21).

The higher the status of the user v in social networks is, the higher the credibility of the user v is. For instance, if the values of PR_u , PR_v , and PR_w are 0.8, 1, and 0.6, respectively, the user u is likely to accept the recommendation of v rather than w . Otherwise, the possibility that user v accepts the suggestion of u is less than the possibility that user u accepts the suggestion of v . It is noticed that PR_u and PR_v are normalized values of social status.

3.4.3. The Proposed Enhanced Recommendation Model Based on Social Networks. Similar to other matrix factorization methods [18, 21], zero-mean spherical Gaussian priors are placed on user and item feature vectors as follows (see (41)-(42)):

$$p(U | \sigma_U^2) = \prod_{u=1}^N N(U_u | 0, \sigma_U^2 I) \quad (41)$$

$$p(V | \sigma_V^2) = \prod_{i=1}^M N(V_i | 0, \sigma_V^2 I) \quad (42)$$

Hence, through a Bayesian inference, the posterior probability of latent feature vectors of U and V can be obtained as follows (see (43)):

$$\begin{aligned} p(U, V | R, \sigma_R^2, \sigma_U^2, \sigma_V^2) & \propto p(R | U, V, \sigma_R^2) p(U | \sigma_U^2) p(V | \sigma_V^2) \\ & = \prod_{u=1}^N \prod_{i=1}^M N[r_{ui} | g(U_u^T V_i), \sigma_R^2]^{I_{ui}^R} \\ & \quad \times \prod_{u=1}^N N(U_u | 0, \sigma_U^2 I) \times \prod_{i=1}^M N(V_i | 0, \sigma_V^2 I) \end{aligned} \quad (43)$$

where I_{ui}^R is the indicator function that is equal to 1 if u has rated i and equal to 0 otherwise. For the user latent features, there are two influence factors: the zero-mean Gaussian prior to avoid overfitting and the conditional distribution of user latent features given the latent features of his trusted neighbors (see (44)).

$$\begin{aligned} p(U | T, \sigma_U^2, \sigma_R^2) & \propto p(U | \sigma_U^2) \times p(U | T, \sigma_R^2) \\ & = \prod_{u=1}^N N(U_u | 0, \sigma_U^2 I) \\ & \quad \times \prod_{u=1}^N N\left(U_u | \sum_{v \in N_u} T_{uv}^* U_v, \sigma_U^2 I\right) \end{aligned} \quad (44)$$

Similar to (42), through a Bayesian inference, the posterior probability of latent feature vectors given the rating and social trust matrices can be obtained as follows (see (45)):

$$\begin{aligned} p(U, V | R, T, \sigma_R^2, \sigma_T^2, \sigma_U^2, \sigma_V^2) & \propto p(R | U, V, \sigma_R^2) p(U | T, \sigma_T^2, \sigma_U^2) p(V | \sigma_V^2) \\ & = \prod_{u=1}^N \prod_{i=1}^M [N(R_{ui} | g(U_u^T V_i), \sigma_R^2)]^{I_{ui}^R} \\ & \quad \times \prod_{u=1}^N N\left(U_u | g\left(\sum_{k \in N_u} T_{uk}^* U_k^T\right), \sigma_T^2 I\right) \\ & \quad \times \prod_{i=1}^M N(V_i | 0, \sigma_V^2 I) \end{aligned} \quad (45)$$

Similarly, item's similarity is introduced into matrix factorization model. The idea of social matrix factorization is to obtain a high quality f -dimensional feature vector V of items based on analyzing the item's similarity matrix S . Let $V \in R^{f \times M}$ and $Z \in R^{f \times M}$ be the latent item and auxiliary feature matrices. The conditional distribution over the observed item social network relationships is described as follows (see (46)):

$$p(S | V, Z, \sigma_S^2) = \prod_{i=1}^M \prod_{j=1}^M [N(S_{ij} | g(V_i^T Z_j), \sigma_S^2)]^{I_{ij}^S} \quad (46)$$

where S_{ij} denotes the similarity between items i and j . According to Figure 6, zero-mean spherical Gaussian prior on auxiliary feature vector is as follows (see (47)):

$$p(Z | \sigma_Z^2) = \prod_{j=1}^M N(Z_j | 0, \sigma_Z^2 I) \quad (47)$$

Hence, through a Bayesian inference, the posterior probability of latent feature vectors given the item's similarity matrix can be obtained as follows (see (48)):

$$\begin{aligned} p(V, Z | S, \sigma_V^2, \sigma_S^2, \sigma_Z^2) & \propto p(S | V, Z, \sigma_S^2) p(V | \sigma_V^2) p(Z | \sigma_Z^2) \\ & = \prod_{i=1}^M \prod_{j=1}^M [N(S_{ij} | g(V_i^T Z_j), \sigma_S^2)]^{I_{ij}^S} \\ & \quad \times \prod_{i=1}^M N(V_i | 0, \sigma_V^2 I) \times \prod_{j=1}^M N(Z_j | 0, \sigma_Z^2 I) \end{aligned} \quad (48)$$

Based on Figure 6, using a Bayesian inference, the posterior probability of latent feature vectors of ESMF given

rating, item similarity, and social trust matrices is described as follows (see (49)):

$$\begin{aligned}
 & p(U, V, Z | R, T, S, \sigma_R^2, \sigma_T^2, \sigma_U^2, \sigma_V^2, \sigma_Z^2, \sigma_S^2) \\
 & \propto p(R | U, V, \sigma_R^2) p((S | V, Z, \sigma_S^2)) \\
 & \cdot p(U | T, \sigma_T^2, \sigma_U^2) p(V | \sigma_V^2) p(Z | \sigma_Z^2) \\
 & = \prod_{u=1}^N \prod_{i=1}^M [N(R_{ui} | g(U_u^T V_i), \sigma_R^2)]^{I_{ui}^R} \\
 & = \prod_{i=1}^M \prod_{j=1}^M [N(S_{ij} | g(V_i^T Z_j), \sigma_S^2)]^{I_{ij}^S} \\
 & \times \prod_{u=1}^N N\left(U_u | g\left(\sum_{k \in N_u} T_{uk}^* U_k^T\right), \sigma_U^2 I\right) \\
 & \times \prod_{u=1}^N N(U_u | 0, \sigma_U^2 I) \times \prod_{i=1}^M N(V_i | 0, \sigma_V^2 I) \\
 & \times \prod_{j=1}^M N(Z_j | 0, \sigma_Z^2 I)
 \end{aligned} \quad (49)$$

The log of the above posterior probability can be computed, and the parameters (observation noise variance and prior variance) are kept fixed, then maximizing the log-posterior over latent features of users and items is equivalent to minimizing the following objective function (see (50)):

$$\begin{aligned}
 L(R, U, V, T, Z, S) & = \frac{1}{2} \sum_{u=1}^N \sum_{i=1}^M (R_{ui} - g(U_u^T V_i))^2 + \frac{\lambda_T}{2} \\
 & \cdot \sum_{u=1}^N \left(\left(U_u - \sum_{v \in N_u} T_{uv}^* U_v \right)^T - \left(U_u \sum_{v \in N_u} T_{uv}^* U_v \right) \right) \\
 & + \frac{\lambda_S}{2} \sum_{j=1}^M \sum_{i=1}^M (S_{ij} - g(Z_j^T V_i))^2 + \frac{\lambda_U}{2} \|U\|_F^2 \\
 & + \frac{\lambda_V}{2} \|V\|_F^2 + \frac{\lambda_Z}{2} \|Z\|_F^2
 \end{aligned} \quad (50)$$

Among them, $\lambda_T = \sigma_R^2 / \sigma_T^2$, $\lambda_U = \sigma_R^2 / \sigma_U^2$, $\lambda_V = \sigma_R^2 / \sigma_V^2$, and $\lambda_Z = \sigma_R^2 / \sigma_Z^2$. T_{uv}^* represents the degree of users u trust in v , whose formula is shown in (39). N and M are the number of users and items, respectively. $g(x)$ is the logistic function $g(x) = 1 / (1 + e^{-x})$, which bounds the range of $U^T V$ within $[0, 1]$. The value of R_{ui} is also normalized to the range of $[0, 1]$ using the function $f(x) = (x-1) / (\max-1)$, and \max is the maximum of ratings in the user-item rating matrix. U_u and U_v are the latent features of users u and v , respectively, and $\|U\|_F^2$, $\|V\|_F^2$, and $\|Z\|_F^2$ are the Frobenius 2-norm of matrices U , V , and Z respectively.

Then the stochastic gradient descent method is used to optimize the aforementioned objective function as follows (see (51)-(53)):

$$\begin{aligned}
 \frac{\partial L}{\partial U_u} & = - \sum_{i=1}^M I_{ui}^R V_i g'(U_u^T V_i) (g(U_u^T V_i) - R_{ui}) + \lambda_U U_u \\
 & + \lambda_T \left(U_u - \sum_{v \in N_u} T_{uv} U_v \right) \\
 & - \lambda_T \sum_{v | u \in N_u} T_{vu} \left(U_v - \sum_{w \in N_u} T_{vw} U_w \right)
 \end{aligned} \quad (51)$$

$$\begin{aligned}
 \frac{\partial L}{\partial V_i} & = \sum_{u=1}^N I_{ui}^R U_u g'(U_u^T V_i) (g(U_u^T V_i) - R_{ui}) \\
 & + \lambda_S \sum_{j=1}^M I_{ij}^S g'(Z_j^T V_i) (g(Z_j^T V_i) - S_{ij}) Z_j \\
 & + \lambda_V V_i
 \end{aligned} \quad (52)$$

$$\frac{\partial L}{\partial Z_j} = \lambda_S \sum_{i=1}^M I_{ij}^S V_i g'(Z_j^T V_i) (g(Z_j^T V_i) - S_{ij}) + \lambda_Z Z_j \quad (53)$$

where $g'(x)$ is the derivative of logistic function; $g'(x) = e^{-x} / (1 + e^{-x})^2$.

The proposed hybrid recommendation algorithm is summarized in Algorithm 2.

3.5. Complexity Analysis. The amount of data available in many practical applications of RS can be enormous and the scalability of recommendation algorithms is a crucial factor for a successful system deployment. Therefore, considering the efficiency of execution of the system, one has to distinguish between the offline and online computational complexity of an algorithm. In this paper, filling the rating matrix using improved CF recommendation algorithm, calculation of the E-step and M-step, and model training of matrix factorization are all executed in the offline stage, and rating predictions of EM and matrix factorization algorithms are executed in online stage.

It is very sparse for the original user-item rating matrix, and it needs to use the CF algorithm to calculate the user's similarity and item's similarity offline to fill the user-item rating matrix. The computational complexity is $O(\bar{r} * \bar{r})$, where \bar{r} is the average number of ratings per user, and it is a very small value. Analyzing the offline complexity of the EM algorithm requires first of all to calculate the E-step and M-step, respectively. For a single E-step, the computational complexity is $O(k * N)$. In the M-step, the posterior probabilities for each rating are accumulated to form the new estimates for $p(z_k | u)$, and the M-step also requires $O(k * N)$ operations. Here, k is the number of clusters, and N is the number of users. For the enhanced social recommendation model, the main computation is to estimate the objective function L and its gradients against variables. Because of the sparsity of R and T , the computational complexity of

Input: ratings in training set R , user social matrix T , k , λ_T , λ_S , learning rate α .
Output: user latent feature matrix U and item latent feature matrix V

- 1 Initialize α , λ_T , λ_S , λ_Z , k , U , and V .
- 2 Fill the rating matrix based on Eq. (2) and Eq. (22)
- 3 Obtain the preferences on item attribute based on Eq. (28)
- 4 Predict partial ratings based on Eq. (37)
- 5 Reset the unreliable ratings to zeroes using Eq. (38)
- 6 Repeat
 - 7 For $r_{ij} \in R$ do
 - 8 update $U_u \leftarrow U_u - \alpha(\partial L / \partial U_u)$ based on Eq. (51)
 - 9 update $V_i \leftarrow V_i - \alpha(\partial L / \partial V_i)$ based on Eq. (52)
 - 10 update $Z_j \leftarrow Z_j - \alpha(\partial L / \partial Z_j)$ based on Eq. (53)
 - 11 End for
- 12 Until convergence
- 13 Return U , V and C
- 14 Predict final ratings based on Eq. (7)

ALGORITHM 2: A hybrid recommendation algorithm based on Gaussian mixture model and enhanced social matrix factorization technology.

evaluating the object function L is $O(\rho_R f + \rho_T f + \rho_S f)$, where ρ_R , ρ_T , and ρ_S are the numbers of nonzero entries in matrices R , T , and S , respectively. f is the dimensions of the latent feature vectors. The computational complexities for gradients $\partial L / \partial U_u$, $\partial L / \partial V_i$, and $\partial L / \partial C_j$ in ((50)-(52)) are $O(\rho_R f + \rho_T f)$, $O(\rho_R f + \rho_T f)$, and $O(\rho_R f)$, respectively. Therefore, the total computational complexity in one iteration is $O(\rho_R f + \rho_T f + \rho_S f)$, which is linear with respect to the number of observations in the two sparse matrices.

More important for a lot of applications is the online complexity of computing predictions in a dynamic environment. For a prediction r_{ua} in (36), $p(z_k | u)$ and μ_{ka} are assumed to be explicitly available as part of the statistical model, this requires $2k$ arithmetic operations, so the computational complexity is $O(k)$. For the enhanced social matrix factorization algorithm, the computational complexity is $O(f * M)$ for one iteration, where M is the number of items. Thus, the total online computational complexity of the proposed hybrid system in this paper is $O(k + f * M)$.

Compared with the existing recommendation algorithms, online computational complexity in this paper is superior to that of algorithms in [18, 19, 41, 46], and they are on the same order of magnitude. Offline computational complexity of the proposed hybrid system exceeds that of most traditional methods, such as user-based CF, item-based CF, SVD, and k -means CF; this is because our method only fills very small amounts of data using CF and EM. Compared with the most social matrix factorization algorithms, offline computational complexity of these algorithms is all on the same order of magnitude. The results of complexity analysis show that the hybrid algorithm proposed in this paper is efficient and can be extended to larger datasets.

4. Experiment

In order to evaluate the performance of the proposed method in this paper, several experiments are performed to show the effectiveness of our proposed method. In particular, the proposed method is compared to other major existing

recommendation approaches in terms of their recommendation performances on Epinions and Tencent datasets.

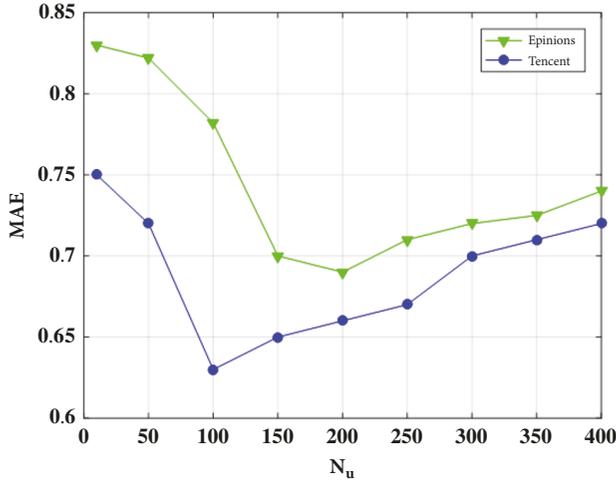
4.1. Datasets. In this paper, two real-world availability datasets, namely, Epinions and Tencent datasets, are used for conducting the experiments. The two real-world datasets contain the trust statements among users, so that the information can be integrated into improved similarity calculation and the enhanced social matrix factorization model to improve the accuracy of recommendations.

Epinions.com is a consumer opinion site that was established to facilitate knowledge sharing about products. Users on Epinions can write reviews about items (e.g., foods, books, and electronics) and assign numeric ratings, which range from 1 to 5, to these items [30]. Moreover, these users can also express their trust statements with the other users. The values of the trust statements in this dataset are 0 or 1. The extracted dataset from the Epinions website consists of 1,261,218 ratings rated by 12,630 users on 3,620 different items. The sparsity is 97.24%.

Tencent dataset is from the track 1 task of the 2012 KDD, provided by Tencent microblog. Tencent microblog, an online social networking site similar to Facebook and Twitter, has become an important communication platform for making friends and sharing information [17]. This dataset is sampled from 50 days of behavioral data of about 200 million registered users, including about 2 million active users, 6,000 items, and 300 million records of historical activity. A smaller dataset is extracted, which contains 326,560 ratings data from 9,650 users on 1,650 items. The sparsity is 97.95%.

4.2. Evaluation Measures. In this paper, Mean Absolute Error (MAE) is used for evaluating the performance of the proposed methods. The MAE measure for the user u is calculated as follows [19] (see (54)):

$$MAE = \frac{\sum_{i=1}^N |R_{ui} - \hat{R}_{ui}|}{N} \quad (54)$$


 FIGURE 7: The effect of parameter N_u on the MAE.

where R_{ui} and \hat{R}_{ui} are real and predicted ratings of the item i for the user u , respectively. N is the total number of ratings that are predicted by the recommendation method.

In addition, precision and recall are also widely used metrics in recommender systems. The evaluation metrics are averaged over all users. The items are sorted according to their ratings from the largest to the smallest, the top N items are recommended for the current active user, and the N items are compared with the most relevant items to in the test set. The larger the value of precision@ N ($P@N$) is, the higher the accuracy is. Recall@ N ($R@N$) describes how many percentage of related items are included in the list of the recommendation. The calculations of accuracy and recall metrics are depicted as follows [54] (see (55)-(56)):

$$P@N = \frac{|\text{relevant_items} \cap \text{top@N_items}|}{N} \quad (55)$$

$$R@N = \frac{|\text{relevant_items} \cap \text{top@N_items}|}{|\text{relevant_items}|} \quad (56)$$

where top@N_items and relevant_items are the recommended and actually liked list, respectively.

4.3. Results of the Experiments

4.3.1. Parameter Settings. Table 2 shows the details about the parameters used in all methods, including their meanings and the default values.

4.3.2. The Influence of the Parameters. Firstly, we analyze the impact of the parameter changes in the data processing process of each step on the proposed algorithm and find out their optimal configuration by setting up multiple sets of different parameter configurations.

The parameter N_u is an important parameter which defines the number of nearest neighbors for the active user. The performed experiments test the effect of different values

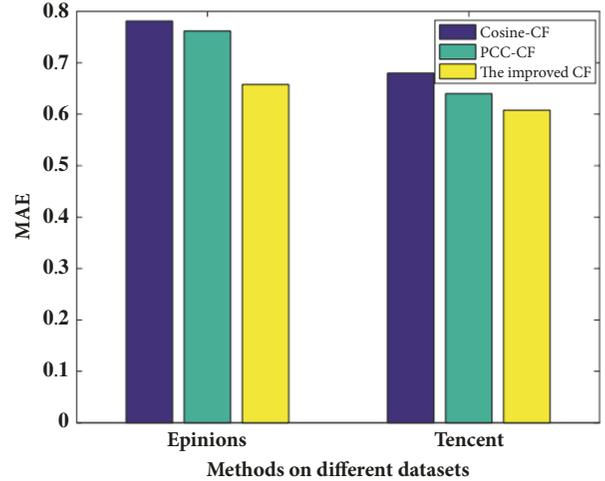


FIGURE 8: Impact of similarity measures.

of N_u (i.e., $N_u = 10, 50, 100, 150, 200, 250, 300, 350,$ and 400) on the mentioned measures for the Epinions and Tencent datasets. Figure 7 shows the results of the MAE measure based on different values of the parameter N_u for both of the mentioned datasets. As the number of the nearest neighbors grows, the MAE measure gradually increases; however, when the numbers of the nearest neighbors exceed 200 for the Epinions dataset and 100 for Tencent dataset, respectively, the MAE measures begin to decrease, indicating that some weak similarity neighbors are introduced to increase noisy data, and their performances have begun seriously degraded.

Figure 8 shows the performance of the CF method on different datasets through the use of different similarities as (1), (3), and (19), respectively, i.e., cosine, PCC, and the improved cosine similarity. The numbers of nearest neighbors N_u are set to 200 and 100 on Epinions and Tencent datasets, respectively. It can be seen that the accuracy of the recommendations has been significantly improved. It indicates that the proposed improved cosine similarity method is superior to the traditional cosine similarity and PCC method on the Epinions and Tencent datasets.

Moreover, the parameters γ , ω , and φ are important parameters which control the influences of the common item set rated by users, time attenuation factor, and the tradeoff between the similarities and the trust relationships among users, respectively. We set γ from 0.5 to 5, ω from 0.01 to 1, and φ from 0 to 1 to evaluate the performance of the experiment. Figures 9, 10, and 11 show the MAE measures based on the different values of the parameters γ , ω , and φ on improved CF recommendation algorithm, while N_u is set to 250 and 50 for the Epinions and Tencent datasets, respectively. As shown in Figures 9, 10, and 11, the accuracies reach the highest level when $\gamma = 2.5$, $\omega = 0.05$, $\varphi = 0.4$ and $\gamma = 3.5$, $\omega = 0.08$, $\varphi = 0.4$ on the Epinions and Tencent datasets, respectively.

As shown in Figure 11, it can be concluded that the recommendation accuracy can achieve its better performance when $\varphi = 0.4$; i.e., the trust relationship and the similarity reach a balance on the Epinions and Tencent datasets, respectively.

TABLE 2: Description and default values of parameters.

Symbol	Description	Default value
λ_U	The regularization constant of user feature in [18–22].	$\lambda_U = 0.1$
λ_V	The regularization constant of item feature in [18–22].	$\lambda_V = 0.1$
λ_T	The regularization constant of user’s trust in [18, 19]	$\lambda_T = 5$
λ_Z	The regularization constant of social relationships between items.	$\lambda_Z = 0.1$
λ_C	The tradeoff parameter plays the role of adjusting the effects of interpersonal trust between users in [21, 22].	$\lambda_C = 10$
λ_S	The tradeoff parameter plays the role of adjusting the effect of social relationships between items.	$\lambda_S = 5$
α	The tradeoff parameter plays the role of adjusting the effects from recommendations of neighbors and trusted friends in [18].	$\alpha = 0.5$
k	The number of clusters.	$k = 15$
f	The dimension of latent feature space in [18, 19, 21].	$f = 10$
N_u	The number of nearest neighbors for the active user.	100
γ	The tradeoff parameter controls the influence of the common item set rated by users.	γ
ω	The tradeoff parameter controls the influence of time attenuation factor.	ω
φ	The tradeoff parameter controls the influences from the similarities and trust relationships among users,	φ
μ	The tradeoff parameter controls the effects of similarity between users and the time-weighted similarity between users.	$\mu = 0.5$
δ	The tradeoff parameter controls the effects of the error caused by the estimated values in [23].	$\delta = 0.5$

In order to obtain a more optimized combination of parameters, we choose the appropriate value for parameters combining from a single parameter optimal interval. Then, the experimental results of the parameters combination performance are presented in Table 3.

Secondly, in order to evaluate the influence of number of clusters on recommendation accuracy, several experiments are conducted on the Epinions and Tencent datasets, and the value of k is set from 1 to 35. As shown in Figure 12, it can be seen that the predicted rating has a lower predicted error when k is set to relatively small values, and the value of MAE grows quickly when the value of k exceeds 8. It can be seen that the predicted error reaches the minimum when the value of k gets closer to 15.

Thirdly, in order to evaluate the influence of the parameters λ_T and λ_S on recommendation accuracy, experiments are conducted through setting different values of λ_T and λ_S . Among them, λ_T balances the information from the

user-item rating matrix and the user social trust network. If $\lambda_T = 0$, the user-item rating matrix is only mined for matrix factorization, and if $\lambda_T = \infty$, the social network information is only extracted to predict user’s preference. λ_S controls the influence of item’s similarity on the item latent feature space V . Figure 13 shows the impact of λ_T when $\lambda_S = 10$ and $\lambda_S = 20$ on the MAE measure for the Epinions and Tencent datasets, respectively. As shown in Figure 13, our model obtain the lowest MAE at 0.616 when $\lambda_T = 10$ on the Epinions dataset and at 0.585 when $\lambda_T = 7$ on the Tencent dataset, respectively.

Figure 14 shows the impact of λ_S when $\lambda_T = 10$ and $\lambda_T = 7$ on the MAE measure for the Epinions and Tencent datasets, respectively. It can be observed from Figures 13 and 14 that the values of λ_T and λ_S affect the recommendation results significantly, which demonstrate that, fusing the item’s social relationship, user’s trust relationship with social status, with the user-item rating matrix greatly, improves the recommendation accuracy. As λ_S increases, the prediction accuracy also

TABLE 3: Parameters combination performance on Epinions and Tencent datasets.

Epinions dataset		Tencent dataset	
Parameters combination	MAE	Parameters combination	MAE
$\gamma = 0.5, \omega = 0.01, \varphi = 0$	0.781	$\gamma = 0.5, \omega = 0.01, \varphi = 0$	0.721
$\gamma = 0.5, \omega = 0.05, \varphi = 0.4$	0.716	$\gamma = 0.5, \omega = 0.08, \varphi = 0.4$	0.658
$\gamma = 0.5, \omega = 0.07, \varphi = 0.5$	0.726	$\gamma = 0.5, \omega = 0.09, \varphi = 0.5$	0.672
$\gamma = 2, \omega = 0.01, \varphi = 0$	0.697	$\gamma = 2, \omega = 0.01, \varphi = 0$	0.691
$\gamma = 2, \omega = 0.05, \varphi = 0.4$	0.665	$\gamma = 2, \omega = 0.08, \varphi = 0.4$	0.628
$\gamma = 2, \omega = 0.05, \varphi = 0.5$	0.671	$\gamma = 2, \omega = 0.09, \varphi = 0.5$	0.641
$\gamma = 2.5, \omega = 0.01, \varphi = 0$	0.722	$\gamma = 2.5, \omega = 0.01, \varphi = 0$	0.679
$\gamma = 2.5, \omega = 0.05, \varphi = 0.4$	0.656	$\gamma = 2.5, \omega = 0.08, \varphi = 0.4$	0.602
$\gamma = 2.5, \omega = 0.07, \varphi = 0.5$	0.668	$\gamma = 2.5, \omega = 0.09, \varphi = 0.5$	0.627
$\gamma = 3, \omega = 0.01, \varphi = 0$	0.721	$\gamma = 3.5, \omega = 0.01, \varphi = 0$	0.681
$\gamma = 3, \omega = 0.05, \varphi = 0.4$	0.662	$\gamma = 3.5, \omega = 0.08, \varphi = 0.4$	0.616
$\gamma = 3, \omega = 0.07, \varphi = 0.5$	0.672	$\gamma = 3.5, \omega = 0.09, \varphi = 0.5$	0.629
$\gamma = 3.5, \omega = 0.01, \varphi = 0$	0.699	$\gamma = 4, \omega = 0.01, \varphi = 0$	0.691
$\gamma = 3.5, \omega = 0.05, \varphi = 0.4$	0.670	$\gamma = 4, \omega = 0.08, \varphi = 0.4$	0.618
$\gamma = 3.5, \omega = 0.07, \varphi = 0.5$	0.675	$\gamma = 4, \omega = 0.09, \varphi = 0.5$	0.633

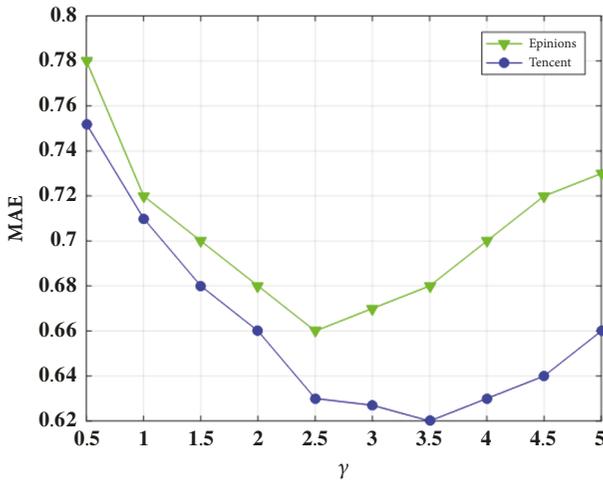


FIGURE 9: Impact of parameter γ on the MAE.

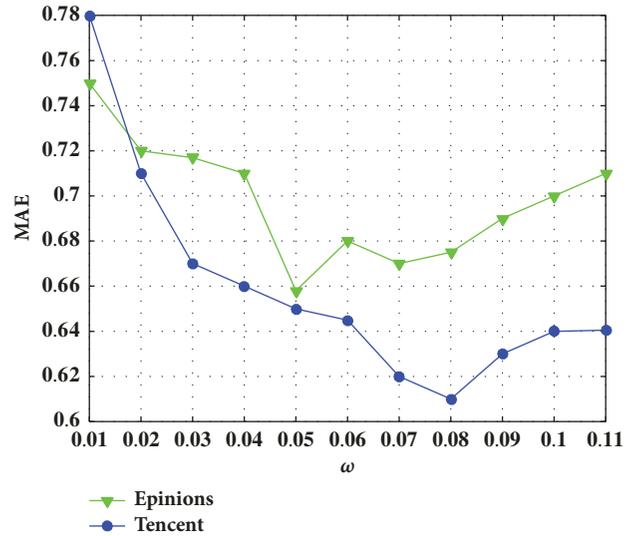


FIGURE 10: Impact of parameter ω on the MAE.

increases at first, but when λ_s surpasses a certain threshold, the prediction accuracy decrease with further increase of the value of λ_s . The trend of λ_T is similar to λ_s . Our model obtain the lowest MAE at 0.591 when $\lambda_s = 15$ on the Epinions dataset and at 0.553 when $\lambda_s = 20$ on the Tencent dataset, respectively.

In addition, the number of hidden features, i.e., f , is another important parameter affecting the recommendation performance of the proposed algorithm. f varied the range from 5 to 50 with a step value of 5 and other parameters $\lambda_T = 10, \lambda_s = 15$ and $\lambda_T = 7, \lambda_s = 20$ on the Epinions and Tencent datasets, respectively. Figure 15 shows the MAE result based on different values of the parameter f for both of the mentioned datasets. It can be observed that the value of MAE decreases at first and then gradually increases and finally tends to be stable. The overall recommendation performance decreases with the increase of f . This observation shows that although the increase of f can make the matrix

factorization model show more hidden features, some noise will be introduced at the same time to reduce the accuracy of recommendation algorithm. It verifies the basic assumptions of the matrix factorization model: only a small amount of hidden factors affects the user's preferences and characterizes the item.

4.3.3. *Performance Comparison and Analysis.* In the experiments, the proposed method (GMMESMF) is compared to the probabilistic matrix factorization (PMF) [34], reliability-based trust-aware collaborative filtering (RTCF) [24], matrix factorization-based model for recommendation in social rating networks (SocialMF) [18], time- and community-aware RS (TCARS) [49], context-aware recommender system via individual trust among users (CSIT) [19], imputation-based

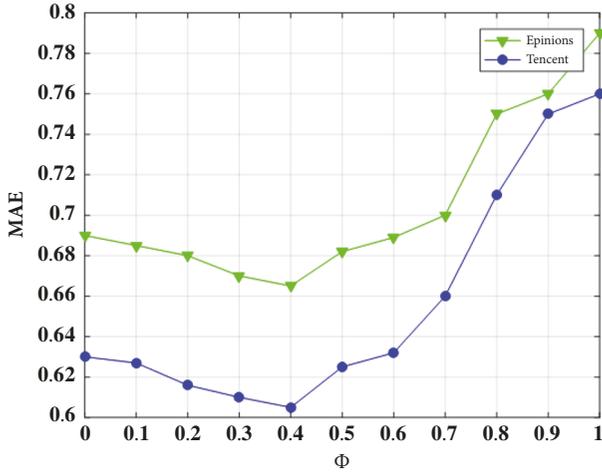


FIGURE 11: Impact of parameter ϕ on the MAE.

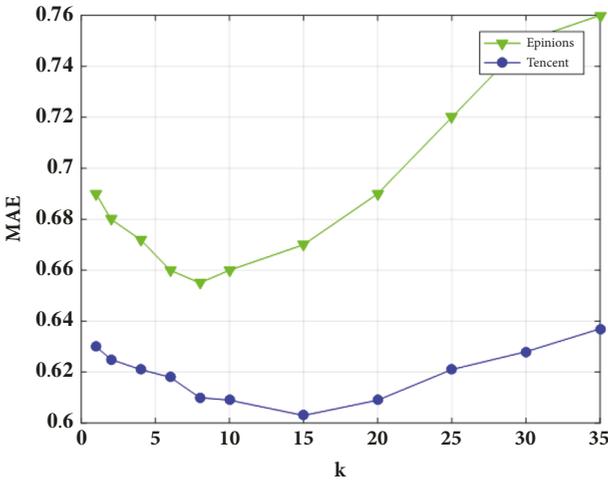


FIGURE 12: Impact of parameter k on the MAE.

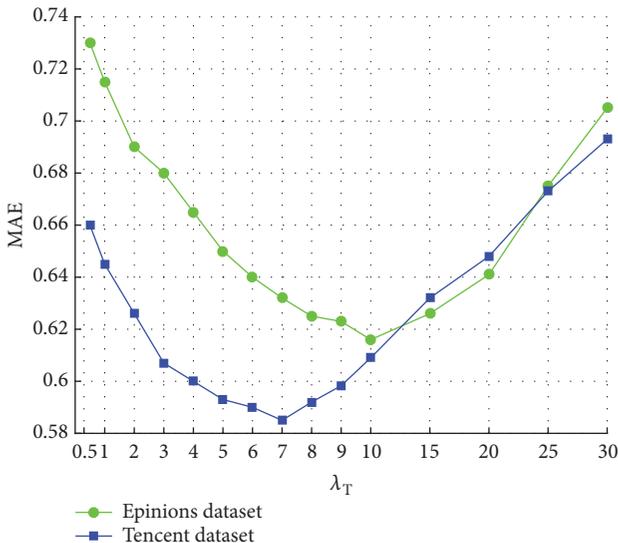


FIGURE 13: Impact of parameter λ_T on the MAE (dimensionality = 5).

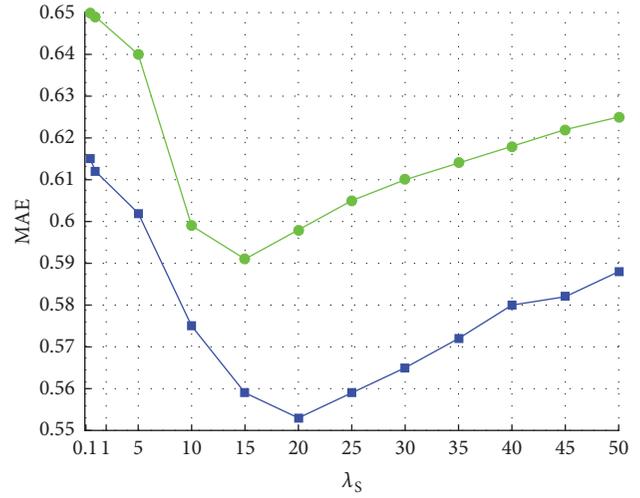


FIGURE 14: Impact of parameter λ_S on the MAE (dimensionality = 10).

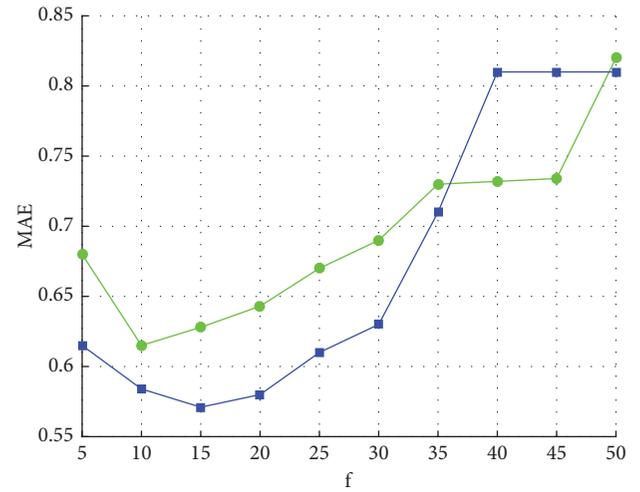


FIGURE 15: Impact of parameter f on the MAE.

matrix factorization (IMF) [23], and implicit social recommendation (ISRec) [22] in terms of the MAE, P@N, and R@N measures on the Epinions and Tencent datasets.

PMF is proposed by Salakhutdinov and only uses the user-item rating matrix for recommendations based on probabilistic matrix factorization. RTCF is proposed by Moradi to improve the accuracy of the trust-aware RS, and a novel trust-based reliability measure is used to evaluate the quality of the predicted ratings. SocialMF is a recommendation algorithm based on social networks proposed by Jamali, which adds a trust propagation mechanism to PMF to improve the accuracy of recommendations. TCARS is a novel recommendation method that efficiently uses the time of ratings and an improved overlapping community detection

TABLE 4: MAE and P@N results on Epinions and Tencent datasets.

Datasets	Measures	PMF	RTCF	SocialMF	TCARS	CSIT	IMF	ISRec	GMMESMF
Epinions	MAE	0.735	0.672	0.683	0.648	0.668	0.705	0.636	0.591
	P@10	0.365	0.425	0.423	0.438	0.435	0.401	0.441	0.452
	R@10	0.218	0.246	0.244	0.252	0.248	0.223	0.258	0.265
	P@20	0.372	0.437	0.426	0.445	0.439	0.389	0.449	0.478
	R@20	0.223	0.254	0.252	0.257	0.259	0.236	0.263	0.279
Tencent	MAE	0.615	0.592	0.596	0.581	0.587	0.608	0.572	0.553
	P@10	0.388	0.451	0.448	0.472	0.455	0.396	0.476	0.487
	R@10	0.236	0.249	0.246	0.253	0.248	0.238	0.258	0.268
	P@20	0.391	0.470	0.465	0.478	0.472	0.412	0.481	0.496
	R@20	0.241	0.253	0.251	0.259	0.255	0.239	0.262	0.275

method to complete the recommendation lists for users. ISRec is an implicit recommendation model, which integrates user's and item's social relationships with matrix factorization through constructing implicit social relationship. CSIT is proposed by Li to optimize the prediction solution in both user latent feature space and user-item rating space using the individual trust among users. IMF is proposed to improve the performance of matrix factorization-based methods using imputed ratings of unknown entries, which utilizes imputed ratings to overcome the sparsity problem. ISRec is proposed to find user's implicit social relationship to reduce the sparsity and improve the recommendation quality.

Each of the datasets has been categorized into fivefold cross validation subsets with the assessment goal of the proposed method. In the process of recommendation prediction, 80% of dataset is used for training and remaining of 20% for testing. As a result, we will have five different results on the basis of the five different testing subsets, and the average of these results will be considered as the final result.

For fair comparison, we set the parameters of different algorithms by referring to the corresponding literatures and experimental results of the comparison algorithms. Under the settings of these parameters, each comparison algorithm achieves optimal performance.

Table 4 shows the MAE, P@N, and R@N measures on the Epinions and Tencent datasets. It compares the proposed method (i.e., GMMESMF) with the PMF, RTCF, SocialMF, PRM, CSIT, DSTNMF, and ISRec in terms of the MAE, P@N, and R@N measures on the Epinions and Tencent datasets. On the Epinions dataset, for fair comparison, the common parameter settings are the same. The number of hidden features, i.e., f , is set to 10, N_u is set to 200, and regularization term parameters $\lambda_U = \lambda_V = 0.1$, $\lambda_S = 15$, $\lambda_C = 20$, and $\lambda_T = 10$ for all algorithms. α is a tradeoff parameter that plays the role in adjusting the degree of influence of neighbors and trusted friends on the recommendation, since it has different degrees of influence in each algorithm, so the settings are different. For TCARS, $\mu = 0.5$, $\omega = 5$. For CSIT, $\alpha = 0.9$. For IMF, $\alpha = 0.5$, $\delta = 0.2$. For GMMESMF, the parameters $\alpha = 0.1$, $\lambda_Z = 0.1$, and the results achieve the MAE value of 0.591, while the P@10 and R@10 obtain 0.452 and 0.265, respectively. The results show that the GMMESMF method

obtains the lowest accuracy and the highest recall on the Epinions dataset.

Likewise, the experiments are repeated on the Tencent dataset, and as shown in Table 4, the performance of the GMMESMF outperforms the other algorithms on the Tencent dataset.

Among all the methods, the traditional method, i.e., PMF, behaves the worst, followed by IMF. This is because they only use the known user-item rating information to predict the unknown ratings without using additional social network information. The performance of IMF is better than PMF because the former improves the performance of matrix factorization-based methods through employing filled ratings of unknown entries to overcome the sparsity problem. IMF increases recommendation accuracy by 4.08% on Epinions dataset and by 1.13% on Tencent dataset in terms of MAE over PMF, respectively.

The other recommendation algorithms, i.e., RTCF, SocialMF, TCARS, CSIT, ISRec, and GMMESMF, outperform PMF and IMF. Among them, RTCF, SocialMF, TCARS, ISRec, CSIT, and GMMESMF increase recommendation accuracy by 8.6%, 7.1%, 11.8%, 13.5%, 9.1%, and 19.6% in terms of MAE over PMF for Epinions dataset and by 3.7%, 3.1%, 5.5%, 7%, 4.6%, and 10.1% in terms of MAE over PMF for Tencent dataset, respectively. This is because these algorithms integrate several social relationships, including user's trust, interest, and item's attributes, into model training to obtain more accurate users' preferences. The experimental results indicate that social relationships are useful to alleviate the problems of data sparsity and cold start.

Our proposed GMMESMF model achieves the best performance compared to other models. GMMESMF increases recommendation accuracy by 12.1%, 13.5%, 8.8%, 11.5%, and 7.1% in terms of MAE over RTCF, SocialMF, TCARS, CSIT, and ISRec for Epinions dataset, respectively. Similarly, there are significant improvements in terms of P@N and R@N over the other recommendation algorithms. The difference with ISRec model is that our proposed GMMESMF model not only combines the individual trust between users, but also considers user's social status and item's social relationship to improve the recommendation performance. This is because GMMESMF simultaneously optimizes the solution in user latent feature space, item latent feature space, and user-item

TABLE 5: MAE on Epinions and Tencent datasets for cold-start users.

Datasets	Measures	PMF	RTCF	SocialMF	TCARS	CSIT	IMF	ISRec	GMMESMF
Epinions	MAE	1.215	0.984	1.021	0.925	0.946	1.078	0.913	0.832
Tencent	MAE	1.086	0.842	0.872	0.817	0.831	0.982	0.805	0.733

rating space using the above three social factors. In addition, our model alleviates data sparsity and ensures the reliability of the predictions through filling unknown ratings based on improved CF method. The experimental results demonstrate that GMMESMF is effective.

Similarly, to verify the effectiveness of our model under cold-start users, an experiment is conducted on cold-start users by using the Epinions and Tencent datasets, and comparative results for different models are shown in Table 5. Here, we define the users who have rated no more than 3 items as cold-start users [11, 13].

It can be observed that GMMESMF improves the MAE performance of PMF by more than 31% and by more than 15%, 18%, 10%, 12%, 22%, and 8% against RTCF, SocialMF, TCARS, CSIT, IMF, and ISRec on the Epinions dataset, respectively. Similarly, GMMESMF improves the MAE performance of PMF by more than 32%, by more than 12%, 15%, 10%, 11%, 25%, and 8% against RTCF, SocialMF, TCARS, CSIT, IMF, and ISRec respectively on the Tencent dataset.

It is because that the GMMESMF model is able to decrease the recommendation error by using the combination of improved CF filling method based on trust relationships and matrix factorization.

5. Conclusions

In this paper, a novel hybrid method is proposed to improve the accuracy of RS. A constrained similarity measure is proposed which is based on cosine similarity, salton factors, and trust relationship. In addition, a novel multiple steps filling method is also proposed to improve the prediction based on the assumptions that a user has multiple interests, similar users have the same preference, and similar items are liked by users with the same interests. The proposed method first uses the user-based CF and the item-based CF to fill in the user-item rating matrix and then uses the Gaussian mixture model to predict ratings to reduce the sparsity of rating matrix. Finally, an enhanced social matrix factorization method is proposed to predict the ratings of unrated items, which fuses user's trust relationship with social status and item's social relationship into matrix factorization algorithm, aiming at improve the accuracy of recommendation through mining the intrinsic connections from the user-item rating matrix and users interaction with items. Extensive experiments are also conducted on two real-world datasets, and the experimental results show that the proposed method achieves higher accuracy compared to the existing major methods in this paper. Although it has some advantages in its recommendation effectiveness, our algorithm still has some limitations and there is room for further improvement in some aspects. The limitations of our approach are twofold: first, GMMESMF needs to be

prepopulated, and we have to fill in some unrated items in the user-item rating matrix before predicting the ratings for all of the unrated items. Second, our GMMESMF model faces an increased computational complexity when the similarity of too many users and items are calculated.

There will be several interesting directions to explore for our future work. We would like to develop a novel k-NN graph construction algorithm that reduces computational complexity and extend the model to make recommendation based on social networks integrating multiple context information. Furthermore, our future study focuses on constructing recommendation models from the perspective of users, such as social relationships between users, social tags, and item's attributes which will be considered and further investigated.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61272286, in part by Joint Funded Projects of the Special Scientific Research Fund for Doctoral Program of Higher Education under Grant 20126101110006, and in part by the Industrial Science and Technology Research Project of Shaanxi Province under Grant 2016GY-123.

References

- [1] F. Ricci, L. Rokach, and B. Shapira, *Recommender system handbook*, Springer, New York, NY, USA, 2015.
- [2] L. Lü, M. Medo, C. H. Yeung, Y. Zhang, Z. Zhang, and T. Zhou, "Recommender systems," *Physics Reports*, vol. 519, no. 1, pp. 1–49, 2012.
- [3] L. Wang, X. Meng, and Y. Zhang, "Context-Aware Recommender Systems," *Journal of Software*, vol. 23, no. 1, pp. 1–20, 2012.
- [4] Q. L. Gao, L. Gao, J. F. Yang, and H. Wang, "A preference elicitation method based on users' cognitive behavior for context-aware recommender system," *Chinese Journal of Computers*, *Jisuanji Xuebao*, vol. 38, no. 9, pp. 1767–1776, 2015.
- [5] R. Saia, L. Boratto, and S. Carta, "Semantics-aware content-based recommender systems: design and architecture guidelines," *Neurocomputing*, 2017.

- [6] Z. Yang, B. Wu, K. Zheng, X. Wang, and L. Lei, "A survey of collaborative filtering-based recommender systems for mobile internet applications," *IEEE Access*, vol. 4, pp. 3273–3287, 2016.
- [7] A. Hernando, J. Bobadilla, F. Ortega, and J. Tejedor, "Incorporating reliability measurements into the predictions of a recommender system," *Information Sciences*, vol. 218, pp. 1–16, 2013.
- [8] D. Chen and F. Kong, "Hybrid Gaussian pLSA model and item based collaborative filtering recommendation," *Computer Engineering and Applications*, vol. 46, no. 23, pp. 209–211, 2010.
- [9] C. Luo, B. Zhang, and Y. Xiang, "Gaussian-Gamma collaborative filtering: A hierarchical Bayesian model for recommender systems," *Journal of Computer & System Sciences*, 2017.
- [10] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Transactions on Information and System Security*, vol. 22, no. 1, pp. 89–115, 2004.
- [11] P. Massa and P. Avesani, "Trust-aware collaborative filtering for recommender systems," in *On the Move to Meaningful Internet Systems: CoopIS, DOA, and ODBASE*, vol. 3290 of *Lecture Notes in Computer Science*, pp. 492–508, Springer, Berlin, Germany, 2004.
- [12] M. M. Azadjalal, P. Moradi, A. Abdollahpouri, and M. Jalili, "A trust-aware recommendation method based on Pareto dominance and confidence concepts," *Knowledge-Based Systems*, vol. 116, pp. 130–143, 2017.
- [13] X. Zheng, Y. Luo, L. Sun, X. Ding, and J. Zhang, "A novel social network hybrid recommender system based on hypergraph topologic structure," *World Wide Web-internet & Web Information Systems*, pp. 1–29, 2017.
- [14] Q. Gao, L. Gao, and J. Yang, "Weighted collaborative filtering algorithm with fused impact factor," *Computer Engineering*, vol. 40, no. 8, pp. 38–42, 2014.
- [15] M. Nilashi, O. Ibrahim, and K. Bagherifard, "A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques," *Expert Systems with Applications*, vol. 92, pp. 507–520, 2018.
- [16] R. Duan, *Study on collaborative filtering recommendation based on textual content and contextual information*, Hefei University of Technology, 2017.
- [17] C. Aggarwal, "Recommender systems," pp. 268–282, Springer Cham Heidelberg, New York, NY, USA, 2016.
- [18] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the 4th ACM Recommender Systems Conference (RecSys '10)*, pp. 135–142, Barcelona, Spain, September 2010.
- [19] J. Li, C. Chen, H. Chen, and C. Tong, "Towards Context-aware Social Recommendation via Individual Trust," *Knowledge-Based Systems*, vol. 127, pp. 58–66, 2017.
- [20] S. Sun, C. Zhao, and Z. Zhang, "A matrix factorization based trust factors model," in *Proceedings of the IEEE International Conference on Information and Automation, ICIA '15 - In conjunction with IEEE International Conference on Automation and Logistics*, pp. 803–808, China, August 2015.
- [21] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM '11)*, pp. 287–296, ACM, Hong Kong, February 2011.
- [22] Y. Li, M. Song, and H. E, "Recommender system using implicit social information," *IEICE Transaction on Information and Systems*, vol. E98D, no. 2, pp. 346–354, 2015.
- [23] M. Ranjbar, P. Moradi, M. Azami, and M. Jalili, "An imputation-based matrix factorization method for improving accuracy of collaborative filtering systems," *Engineering Applications of Artificial Intelligence*, vol. 46, pp. 58–66, 2015.
- [24] 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.
- [25] H. Koohi and K. Kiani, "A new method to find neighbor users that improves the performance of Collaborative Filtering," *Expert Systems with Applications*, vol. 83, pp. 30–39, 2017.
- [26] H. Liu, X. Kong, X. Bai, W. Wang, T. M. Bekele, and F. Xia, "Context-Based Collaborative Filtering for Citation Recommendation," *IEEE Access*, vol. 3, pp. 1695–1703, 2015.
- [27] Y. Li, *Research for collaborative filtering algorithm regarding sparsity*, Northwest University, 2017.
- [28] W. Kong, *Research on the key problems of collaborative filtering recommender system*, National Engineering research center for e-learning Central China Normal University, 2013.
- [29] C. Birtolo, D. Ronca, and G. Aurilio, "Trust-aware clustering collaborative filtering: Identification of relevant items," *IFIP Advances in Information and Communication Technology*, vol. 381, no. 1, pp. 374–384, 2012.
- [30] P. Moradi, S. Ahmadian, and F. Akhlaghian, "An effective trust-based recommendation method using a novel graph clustering algorithm," *Physica A: Statistical Mechanics and its Applications*, vol. 436, pp. 462–481, 2015.
- [31] R. Chen, Q. Hua, and W. Jia, "A hybrid recommendation method and development framework of user interface patterns based on hypergraph theory," *International Journal of Innovative Computing, Information and Control*, vol. 13, no. 4, 2017.
- [32] D. Wu, X. Luo, G. Wang, M. Shang, Y. Yuan, and H. Yan, "A Highly Accurate Framework for Self-Labeled Semisupervised Classification in Industrial Applications," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 909–920, 2018.
- [33] D. Wu, M. Shang, X. Luo et al., "Self-training semi-supervised classification based on density peaks of data," *Neurocomputing*, vol. 275, pp. 180–191, 2018.
- [34] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS '07)*, pp. 252–260, Vancouver, Canada, December 2007.
- [35] T. Ma, J. Zhou, and M. Tang, "Social network and tag sources based augmenting collaborative recommender system," *IEICE Transactions on Information Systems*, vol. 98, no. 4, pp. 902–910, 2015.
- [36] 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.
- [37] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1273–1284, 2014.
- [38] Z. Shajarian, S. A. Seyedi, and P. Moradi, "A clustering-based matrix factorization method to improve the accuracy of recommendation systems," in *Proceedings of the 2017 Iranian Conference on Electrical Engineering (ICEE)*, pp. 2241–2246, Tehran, Iran, May 2017.

- [39] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proceedings of the 7th ACM Conference on Recommender Systems, RecSys 2013*, pp. 165–172, China, October 2013.
- [40] Z. Zhou, *Machine Learning*, Tsinghua University Press, 2016.
- [41] X. Luo, M. Shang, and S. Li, "Efficient Extraction of Non-negative Latent Factors from High-Dimensional and Sparse Matrices in Industrial Applications," in *Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 311–319, Barcelona, Spain, December 2016.
- [42] Y. Li, D. Wang, H. He, L. Jiao, and Y. Xue, "Mining intrinsic information by matrix factorization-based approaches for collaborative filtering in recommender systems," *Neurocomputing*, vol. 249, pp. 48–63, 2017.
- [43] K. Ji and H. Shen, "Making recommendations from top-N user-item subgroups," *Neurocomputing*, vol. 165, pp. 228–237, 2015.
- [44] A. Hernando, J. Bobadilla, and F. Ortega, "A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model," *Knowledge-Based Systems*, vol. 97, pp. 188–202, 2016.
- [45] E. Bojnordi and P. Moradi, "A novel collaborative filtering model based on combination of Correlation method with matrix completion technique," *Iranian Journal of Science and Technology - Transactions of Electrical Engineering*, vol. 37, no. 1, pp. 93–100, 2013.
- [46] B. Yang, Y. Lei, J. Liu, and W. Li, "Social Collaborative Filtering by Trust," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1633–1647, 2017.
- [47] G. Yin, Y. Zhang, and Y. Dong, "A constrained trust recommendation using probabilistic matrix factorization," *Acta Electronica Sinica*, vol. 42, no. 2, pp. 904–911, 2014 (Chinese).
- [48] H. Li, X. Diao, J. Cao, Y. Shang, and Y. Feng, "Integrated collaborative filtering for implicit feedback incorporating covisitation," *IEICE Transaction on Information and Systems*, vol. E100D, no. 7, pp. 1530–1533, 2017.
- [49] F. Rezaeimehr, P. Moradi, S. Ahmadian, N. N. Qader, and M. Jalili, "TCARS: Time- and Community-Aware Recommendation System," *Future Generation Computer Systems*, vol. 78, pp. 419–429, 2018.
- [50] D. Z. Navgaran, P. Moradi, and F. Akhlaghian, "Evolutionary based matrix factorization method for collaborative filtering systems," in *Proceedings of the 21st Iranian Conference on Electrical Engineering, ICEE '13*, Iran, May 2013.
- [51] X. Qian, H. Feng, G. Zhao, and T. Mei, "Personalized recommendation combining user interest and social circle," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1763–1777, 2014.
- [52] Z. Xian, Q. Li, X. Huang, and L. Li, "New SVD-based collaborative filtering algorithms with differential privacy," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 33, no. 4, pp. 2133–2144, 2017.
- [53] Q. Li and S. Li, "Collaborative filtering recommendation algorithm based on spectral clustering and fusion of multiple factors," *Application Research of Computers*, vol. 34, no. 10, pp. 2905–2908, 2017.
- [54] Q. Gao, L. Gao, J. Fan, and J. Ren, "A preference elicitation method based on bipartite graphical correlation and implicit trust," *Neurocomputing*, vol. 237, pp. 92–100, 2017.
- [55] Z. Tan and L. He, "An Efficient Similarity Measure for User-Based Collaborative Filtering Recommender Systems Inspired by the Physical Resonance Principle," *IEEE Access*, vol. 5, pp. 27211–27228, 2017.
- [56] Z. Zhao, "Machine learning algorithm," *Publishing House of Electronics Industry*, pp. 201–225, 2017.
- [57] Y. Wang, X. Wang, and W. L. Zuo, "Trust prediction modeling based on social theories," *Journal of Software. Ruanjian Xuebao*, vol. 25, no. 12, pp. 2893–2904, 2014.

