IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 November 2022, accepted 10 December 2022, date of publication 19 December 2022, date of current version 23 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3230351

APPLIED RESEARCH

Recommendation Model Based on Probabilistic Matrix Factorization, Integrating User Trust Relationship, Interest Mining, and Item Correlation

LIFENG HAN[1], (Member, IEEE), LI CHEN[1], AND XIAOLONG SHI[2]
[1]School of Information Science and Technology, Northwest University, Xi'an 710127, China
[2]School of Computer Science and Technology, Xidian University, Xi'an 710126, China

Corresponding author: Li Chen (chenli@nwu.edu.cn)

This work was supported by the Key Research and Development Program of Shaanxi Province under Grant 2019ZDLGY10-01.

ABSTRACT Personalized recommendation has gained widespread attention in the academic and industrial fields to minimize information overload and has produced good benefits. Current research shows that social recommendations that effectively utilize user trust relationships can solve data sparsity and cold start problems common in traditional collaborative filtering algorithms. However, existing social recommendation models have focused only on direct trust relationships between users and have ignored indirect trust relationships and item correlations. To address these problems, we propose a probabilistic matrix factorization-based recommendation model based on trust relationships, interest mining, and item correlation. The proposed recommendation model considers the direct and indirect trust relationships between users, the similarities in users' preferences for item attributes, and the correlations between items. Finally, the rating of the item is predicted by the target user and provides the target user with personalized item recommendations. We evaluate the recommendation performances of the proposed recommendation model on the FilmTrust and the CiaoDVD datasets and find that it alleviates the user's cold start problem and provides higher recommendation accuracy and diversity than popular algorithms.

INDEX TERMS Correlation relationship, direct trust, indirect trust, heterogeneous network, probability matrix factorization, user interest.

I. INTRODUCTION

Major Internet companies are providing convenient information services and product information at an exponential growth rate due to the rapid development of Internet technology. The resulting information overload must be reduced [1], [2], [3], which is typically achieved using information retrieval and information filtering [4], [5]. A recommendation system is an effective information filtering method. It recommends items that users may be interested in by analyzing the users' historical behavior. For example, 80% of Netflix movies are chosen based on a recommendation system [6] and 60% of YouTube videos are selected based on recommendation results on the home page [7]. Collaborative filtering (CF) algorithms [8] are widely used to develop recommendation systems. They assume that the user is interested in items liked by neighboring users who have similar historical behavior. CF algorithms are divided into memory-based [9] and model-based [10] CF algorithms. The former is divided into user-based [11] and project-based [12] CF algorithms. This type of algorithm has been widely used for personalized e-commerce, news, music recommendations, and in many other fields [13], [14].

However, due to data sparsity and the cold start problem, CF algorithms do not accurately calculate the similarity, preventing them from making accurate, personalized recommendations for cold start users. Therefore, the auxiliary information added to the matrix decomposition model [15],

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

such as demographics, project descriptions, and information from social networks. The addition of auxiliary information improves recommendation accuracy, especially for cold-start users. Manzato [16] proposed a matrix factorization recommendation algorithm based on user preferences and movie genres and categories, improving the recommendation accuracy. Qin et al. [17] proposed a CF recommendation algorithm based on weighted item categories and improved the recommendation accuracy by adding a forget function and user attribute information. Zhang et al. [18] proposed the collaborative user network embedding (CUNE) model to mine the similarity relationship among users; however, this method did not improve the recommendation accuracy when insufficient user-rating data were used.

In addition to choosing items based on similarity, people rely on recommendations from trusted friends. So personalized recommendations based on trust relationships have attracted the attention of Chinese and international scholars [19], [20], [21]. Some model-based social recommendations, such as social recommendations using probabilistic matrix factorization (SoRec) [22], social CF based on trust (TrustMF) [23], and recommendation algorithms based on probability matrix factorization and trust (TrustPMF) [24], combine social networks and rating matrices by sharing the user's latent feature matrix. Other model-based social recommendations represent the user's latent feature matrix through social networks and perform matrix factorization, such as the matrix factorization technique with trust propagation for recommendation in social networks (SocialMF) [25] and the recommendation system with social regularization (SoReg) [26].

However, most users do not have many direct trust relationships with each other. In order to make more effective use of the link information between users, the implicit trust relationship has attracted the attention of some researchers. For example, Guo et al. [51] based on the SVD++ model, the TrustSVD model is proposed considering the direct and indirect effects of project rating and user trust. The experimental results show that the prediction accuracy of TrustSVD is better than that of trust-based and rating-based methods. However, this model only relies on the user scoring matrix and the user trust matrix and does not fully mine the information through the existing scoring matrix to infer the trust relationship between users. Obviously, this model has some limitations.

In view of this deficiency, Cui et al. [52] proposed the DMFA-SR model, and Li et al. [53] proposed the ReHI model. These models use trust propagation in social networks to fully exploit indirect trust relationships between users. Then the trust relationship is integrated into the matrix decomposition model to get the user's predicted score. The experimental results show that these models not only improve the accuracy but also alleviate the cold start problem. However, these models only focus on the trust relationship between users, ignoring the similar relationship between users themselves, and do not fully explore the correlation between items.

Therefore, we propose a probability matrix decomposition model that considers the user trust relationship, user similarity, and item correlation. The main contributions of this paper are as follows:

1) We employ the fusion method, which combines the direct and indirect trust relationships, at the time of calculating the user trust relationship.
2) We propose a method to calculate the indirect trust relationship through the heterogeneous network, which obtains the node path through the improved random walk algorithm, and finally calculates the indirect trust degree between users.
3) We propose a method to calculate the correlation degree between items only from the user-object interaction matrix.
4) We propose a method to calculate user interest similarity based on users' preference for item attributes.
5) We integrate the user trust relationship, interest similarity and project relevance, and finally integrate them into the probability matrix decomposition model to get the user's prediction score of the item.

II. RELATED WORK

This section reviews the related work, including network representation learning, CF recommendation algorithms based on trust relationships, and probabilistic matrix factorization (PMF).

A. RECOMMENDATION MODEL BASED ON NETWORK REPRESENTATION LEARNING

Large amounts of data and networks have accumulated due to the development of Internet technology. The relationships between objects and their nodes are reflected by nodes and edges, such as recommendation systems, knowledge graphs, and social networks. Nodes contain rich attribute information, and edges have connection information, providing great application value for the research and analysis of complex networks. A network is typically described using a simple and intuitive adjacency matrix. However, the expansion of the network results in data sparsity, significantly reducing the calculation efficiency. The emergence of network representation learning aims to solve these problems. Network representation learning is a machine learning method that uses a vector form to represent the network structure and node attributes. The goal is to represent each node in the network as a low-dimensional dense vector containing the topology information of the network.

Existing network representation learning methods include shallow neural networks and deep learning methods. Representative shallow neural networks are online learning of social representations (DeepWalk) [27], scalable feature learning for networks (node2vec) [28], and large-scale information network embedding (LINE) [29]. DeepWalk is based on natural language processing (NLP). It regards the fixed-length node sequences generated by the random walk as sentences and the nodes in the sequences as words. It performs representation using the low-dimensional vectors of the learning nodes based on the Word2Vec model. The node2vec model is based on the concept of bias parameters. It uses a breadth-first search (BFS) and depth-first search (DFS) in random sequence generation. The bias parameters determine the search mode. However, DeepWalk and node2vec consider only the first-order structure of the nodes, i.e., two connected nodes. Since the first-order structure is relatively rare in networks, the LINE model also considers the second-order structure. It is assumed that two nodes have more similarities when they share more neighbor nodes. As a typical representative of the network representation method based on deep learning, structural deep network embedding (SDNE) [30] employs deep learning models to capture non-linear relationships between nodes. The method consists of a supervised Laplacian matrix module and an unsupervised deep self-encoding module. The former models the first-order similarity of nodes, and the latter models the second-order similarity.

These algorithms can effectively analyze network structures; however, many networks are heterogeneous. They contain more information and richer semantic information than homogeneous networks. Some scholars proposed scalable representation learning for heterogeneous networks (metapath2vec), bipartite network embedding (BiNE), and other algorithms [31], [32], [33], [34].

Network representation learning substantially enhances the feature representation capabilities of personalized recommendation systems [35], [36], [37]. The recommender system consists of a large network containing user rating information and item tags. Therefore, network representation learning can substantially enrich the information used by algorithms in the recommender system. This topic has become a research hotspot in personalized recommendation systems. For example, a user-item bipartite graph based on user rating information was extended to a user-user graph, which learned the low-dimensional vector representation of the user nodes, obtained the users' implicit friends, and integrated the implicit social relationships into a matrix factorization model for item recommendation [18]. This type of algorithm has achieved good recommendation performance in rating prediction and *Top-N* list recommendations.

B. CF RECOMMENDATION MODELS

Memory-based CF models often encounter recommendation bottlenecks because they only rely on the user's explicit rating information. Large amounts of complex data are produced on major platforms daily, and the interactions between users and the system are not limited to rating information. Since the number of users and items has increased sharply, user-user relationships, item-item relationships, and factors affecting users' preferences for items have become more complex. Therefore, traditional methods face significant challenges in solving these problems.

Model-based CF methods map user ratings of items to a lower dimensional user feature space, and they have better interpretability and scalability than memory-based CF methods. Therefore, model-based CF methods have become a research hotspot. Matrix factorization has drawn the attention of researchers owing to its simplicity and efficiency.

Many matrix factorization algorithms can mine user and item information, improving the recommendation performance [15], [38]. However, the user and item feature vectors learned by the matrix factorization model cannot fully describe the user's preferences due to data sparsity, reducing the recommendation accuracy. Moreover, traditional matrix factorization models cannot learn the feature vectors of cold-start users or determine their preferences. Thus, these algorithms are not good solutions to the cold-start problem.

Normalized matrix factorization (NMF) has been used to improve the recommendation accuracy of matrix factorization models. Matrix factorization multiplies the dimension elements corresponding to the user and item feature vectors and uses the sum of the equal weights of the product as the user's score of the item.

C. CF RECOMMENDATION BASED ON TRUST RELATIONSHIP

Trust relationships are considered reliable external information that increases the sample size and alleviates the cold-start problem. This information is valuable because people rely on recommendations from people they trust [39], [40].

In [41], user feature vectors were learned by sharing user feature vector matrices and decomposing the rating matrix and trust matrix. This method considers the effects of the user ratings and users' trust in user feature vectors. In [25], it was assumed that users and people they trust had similar feature vectors. Thus, the similarity between the feature vectors depended on the degree of trust. The algorithm described in [42] considered the influences of people that users trusted and did not trust for learning user feature vectors. The user feature vectors were similar for people they trusted and vice versa. However, these algorithms assume that users have similar preferences as people they trust, and the similarity between users and other users was not evaluated. Many factors are considered when users define who they trust, and users may not have similar preferences as the people they trust. The concept of trust correlation was proposed to determine if users trusted people with similar preferences [43]. It was assumed that only trusted people with similar ratings as the target users were friends of the target users. In addition, a trust propagation mechanism was incorporated to address data sparsity and the cold start problem.

III. PRELIMINARIES

This section introduces the definitions and symbols used in the proposed model and the matrix decomposition algorithm and network representation learning.

TABLE 1. Symbols and definitions.

Symbol	Definition
U	Set of users
V	Set of items
u, v	User u, item v
$R_{M\times N}$	User-item rating matrix
$r_{u,v}$	User u's rating for the item v
$T_{M\times M}$	User trust matrix
t_{u_i,u_l}	Trust score of the user u_i to the user u_l
$G=(U,V,E)$	Heterogeneous information network diagram

It is assumed that there are M users $U = \{u_1, u_2, \cdots, u_M\}$ and N items $V = \{v_1, v_2, \cdots, v_N\}$ in a personalized recommendation system. The users rate the items, and a rating matrix $R_{M\times N}$ with M rows and N columns is constructed, wherein the u-th row and i-th column represent the rating of the i-th item by the u-th user. The trust relationship-based recommender system consists of the user-item rating matrix R and the social relationship matrix $T = (T_{uk})_{m\times m} \in \{1, 0\}^{n\times m}$, where $T_{uk} \in \{0, 1\}$ denotes the trust value of user u in user k, 0 denotes distrust, and 1 denotes trust. In addition to explicit trust relationships, we also mine implicit trust relationships between users through network representation. The symbols are defined in Table 1.

A. NETWORK REPRESENTATION LEARNING MODEL

The purpose of network representation learning is to represent the nodes in the network in a low-dimensional, real-valued, and dense vector format to represent the information in a vector space. This format is used as the input of a machine learning model, and the obtained vector representation is used for common applications in social networks, such as visualization, node classification, and link prediction. Online representation learning has been widely used for personalized recommendation systems. We define the following three core concepts of network representation learning.

Definition 1 (Heterogeneous Information Network): It is assumed that the information network can be represented by a graph $G = \{N, E\}$, where N is the node-set, and E is the edge-set. Each entity $n \in N$ belongs to a certain entity type; similarly, each edge $e \in E$ belongs to a certain relationship type. If the number of entities or relationship types is greater than 1, the information network is heterogeneous. Otherwise, it is homogeneous. Suppose the movie user-item rating network can be expressed as $G = \{U, V, E\}$, where U and I represent the user and item node sets, respectively, and E is the edge set. The network consists of two entity types (user and item); thus, the network is heterogeneous and bipartite.

Definition 2 (Heterogeneous Information Network With Weights): It is assumed that the information network with weights can be represented by the graph $G = \{N, E, W\}$. Each entity $n \in N$ is a specific entity type, and each edge $e \in E$ is a specific relationship type. The weight of each edge $w \in W$ belongs to a specific weight attribute type. When the weight attribute type $|M| > 0$, the heterogeneous network has a weight. It is assumed that a movie recommender system contains user information ($User$, U), movie information ($Movie$, M), and user rating information for movies (1–5 scores). The node information includes the users and movies, and the edge information includes the user ratings of the movies, the movie names, and their categories. The user ratings of movies are edges with weights.

Definition 3 (Meta-Path With Weights): Given a weighted heterogeneous network $G = \{N, E, W\}$, the meta-path can be expressed as $P = T_1 \overset{(L_1,M_1)}{\rightarrow} T_2 \overset{(L_2,M_2)}{\rightarrow} \cdots \overset{(L_k,M_k)}{\rightarrow} T_{k+1}$, where T_k represents the entity type, L_k represents the relationship type of the entities, and M_k is the weight of the entities. Suppose there are user U, movie M, and user ratings in a movie recommender system. Meta-path $M(R)U(R)M$ can be used to represent the ratings of two movies by the same user. $M\_1(3)U\_1(5)M\_2$ is an example of the meta-path, indicating that user 1 provides three scores for movie 1 and five scores for movie 2. Meta-path $U(R)M(R)U$ represents the scores of the same movie by different users, and meta-path $U\_1(5)M\_2(3)U\_2$ indicates that user 1 provides five scores for the movie, and user 2 gives the movie three scores.

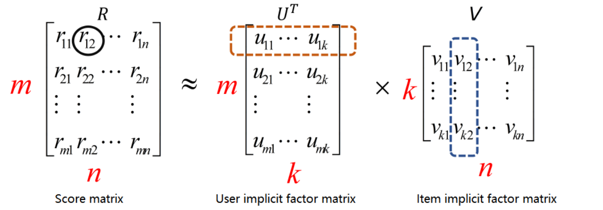


FIGURE 1. Illustration of matrix factorization model. where R is an $m \times n$ matrix, U is an $m \times k$ matrix and V is a $k \times n$ matrix. The j-th column V of the matrix represents the implicit factor vector of item j. k denotes the dimension of the implicit factor vector, usually $k \ll m, n$.

B. MATRIX FACTORIZATION MODEL

The matrix factorization model assumes that the user preferences and item attributes can be expressed as low-dimensional feature vectors, and the user ratings can be represented by the scalar product of the user and item feature vectors. In the personalized recommendation (Fig. 1), the score matrix R of m users for n items is decomposed into the product of matrix U of the user's implicit factor vector and matrix I of the item's implicit factor vector, as shown in (1).

$$R \approx U^T V. \tag{1}$$

The user (U_i) rating of item (V_j) can be predicted by the product of the user's implicit feature vector and the item's implicit feature vector, as shown in (2). The gap between the real and predicted rating values can be expressed by the loss function in (3), where D is the set of users and items after the rating, i.e., the users and items in the training set. The parameter λ controls the regularization parameters to prevent overfitting.

$$\hat{R}_{ij} = U_i^T V_j, \tag{2}$$

$$Loss = \sum_{(i,j)\in D} \left(R_{i,j} - U_i^T V_j\right)^2 + \lambda_u \|U\|_F^2 + \lambda_v \|V\|_F^2. \tag{3}$$

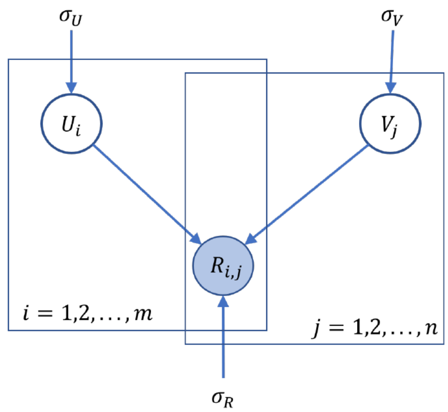


FIGURE 2. Illustration of probabilistic matrix factorization model.

Fig. 2 shows the PMF model. Low-rank latent factor feature matrices of the users and items are derived by decomposing the user-item rating matrix and predicting the missing ratings. R is the rating matrix of m items by n users, U and I are the user and item feature matrices, respectively, U_i and I_j are the user and item feature vectors, and r_{ij} is the rating of item j by user i. The conditional distribution of the rating matrix R can be defined as

$$P\left(R\middle|U,V,\sigma_R^2\right)=\prod_{i=1}^{m}\prod_{j=1}^{n}\left[N\left(r_{ijk}\middle|g\left(U_i^T V_j\right),\sigma_R^2\right)\right]^{I_{ij}^R}. \tag{4}$$

where $N\left(x\middle|\mu,\sigma^2\right)$ denotes that x obeys a Gaussian distribution with a mean value of μ and variance of σ^2; I_{ij}^R is an indicator function; if user U_i rates item I_j, its value is 1; otherwise, it is 0. The logical function $g(x)=1/(1+e^{-x})$ is to limit the value ${U_i}^T I_j$ to [0, 1]. The user and item features obey a spherical Gaussian prior distribution with a mean value of 0, as shown in (5).

$$P\left(U\middle|\sigma_U^2\right)=\prod_{u=1}^{N}N\left(U_u\middle|0,\sigma_U^2 I\right),$$
$$P\left(V\middle|\sigma_v^2\right)=\prod_{i=1}^{M}N\left(V_i\middle|0,\sigma_v^2 I\right). \tag{5}$$

The posterior Bayesian probability of the user and item features can be expressed as:

$$\begin{aligned}p(U,V|R,\sigma_R^2,\sigma_v^2,\sigma_V^2)&\\ \propto p(R|U,V,\sigma_R^2)&\\ p(U|\sigma_U^2)p(I|\sigma_v^2)&\\ =\prod_{u=1}^{N}\prod_{i=1}^{M}N\left[g(U_u^T V_i),\sigma_R^2\right]^{I_{ij}^R}&\\ \times\prod_{u=1}^{N}N(U_u|0,\sigma_U^2 I)\times\prod_{i=1}^{M}N(V_i|0,\sigma_v^2 I).&\end{aligned} \tag{6}$$

IV. THE PROPOSED RECOMMENDATION MODEL

This section introduces the proposed model, framework, and its steps. We discuss the influence of the user-trust relationship on user scoring. The user's interest similarity and item correlation are integrated, and their posterior probabilities are obtained. We use the gradient descent method to optimize the solution and obtain the user's score for the target item.

In addition to direct trust relationships, indirect trust relationships and user interest also influence a user's purchase behavior. Therefore, users may like items associated with the items they need. Based on these considerations, we integrate the user's trust relationship and interest similarity, and item correlation into the proposed PMF recommendation model (Fig. 3).

The method has the following steps.

1) Calculate the degree of direct trust between users through the user-explicit trust relationship or rating matrix.
2) Construct a heterogeneous network using rating information and trust relationships.
3) For each node in the heterogeneous network, calculate the indirect trust degree using the DeepWalk algorithm.
4) Calculate the final user trust value based on the direct and indirect trust degrees to obtain the trust matrix for all users.
5) Calculate the similarity of the user's interest preferences to obtain a user similarity matrix.
6) Calculate the degree of correlation between items to obtain an item incidence matrix.
7) Integrate the user trust matrix, user interest similarity matrix, and item incidence matrix into the PMF model. Predict the user ratings of the items.

A. ESTABLISHMENT OF HETEROGENEOUS NETWORK BY INTEGRATING INFORMATION AND TRUST RELATIONSHIPS

Suppose that a given personalized recommendation system contains user information, movie information, and user rating information for movies (a score of 1–5). We abstract users and movies as network nodes and users' ratings of movies as edges. If a user rates a movie, an edge exists between the user and the movie, and the weight value equals the rating value. The definitions are as follows:

$$U=\left\{u_i\middle|i=1,2,\cdots,n\right\},$$
$$V=\left\{V_j\middle|j=1,2,\cdots,m\right\},$$
$$R=\left\{r_{ij}\middle|i=1,2,\cdots,n,j=1,2,\cdots,m\right\}. \tag{7}$$

where U, V, and R are the user, movie, and rating sets, respectively. The heterogeneous network graph G can be expressed as:

$$G=(N,E,W). \tag{8}$$

where the heterogeneous network graph G is composed of a node set N, edge set E, and weight set W. N is composed of user node U and movie node V. E represents the edges between nodes, which are composed of user ratings

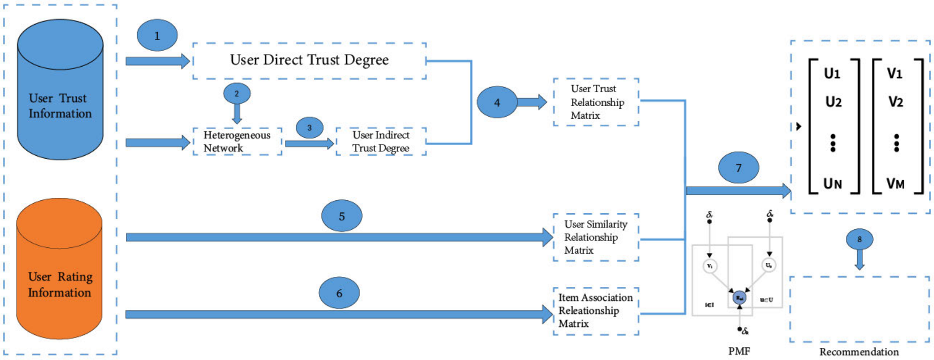


FIGURE 3. The proposed recommendation model. First, the user's direct trust degree is obtained through user trust information and user rating information. Then, the heterogeneous network is constructed through trust information and rating information, deriving the user's indirect trust degree. The user's final trust relationship is obtained by incorporating the direct and indirect trust degrees of the users, and the user trust matrix is constructed. The user similarity matrix and item correlation matrix are obtained from the user's trust information and rating information, respectively. This information is integrated into the probability matrix decomposition model to obtain the prediction score of the item.

TABLE 2. User-item rating matrix.

User	v_1	v_2	v_3	v_4	v_5
u_1					2
u_2		4			
u_3	4	5			2
u_4				4	5
u_5	3	5			2
u_6				3	4
u_7			3		
u_8					

of movies. If the user rates a movie, an edge exists; otherwise, it does not. W represents the weight of the edge. The expression can be defined as:

$$\begin{aligned} N &= U \cup V, \\ E &= \left\{ e_{ij} \middle| i = 1, 2, \cdots, n, j = 1, 2, \cdots, m \right\}, \\ e_{ij} &= \begin{cases} 1, & r_{ij} \text{ exists} \\ 0, & r_{ij} \text{ not exists} \end{cases} \end{aligned} \tag{9}$$

An example of a user rating information table is shown in Table 2. Based on the above definitions, we can obtain the user-item heterogeneous network diagram shown in Fig. 4. Items can connect orphaned users. For example, users u_3 and u_5 have rated items v_1 and v_2; therefore, a connection exists between u_3 and u_5. u_4 and u_6 are connected in the same manner.

We can also connect users to the people they trust to build a denser user-item heterogeneous network. The heterogeneous network in Fig. 5 is optimized from Fig. 4 by connecting users to others they trust. After integrating the trust relationship, the previously isolated users u_1 and u_8 are connected in the new heterogeneous network through u_7. This approach enables mining more reliable user relationships through network embedding. Besides, adding users with trust information to

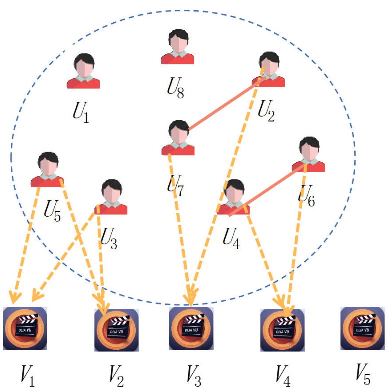


FIGURE 4. Schematic diagram of user-item heterogeneous network. If the user scores the item, the user is connected to the item with an arrow. A user who scores the same item establishes a contact.

the network facilitates finding similar users, alleviating the cold-start problem.

B. CALCULATION OF USER TRUST DEGREE

Trust relationships are crucial to select neighbors of target users because people are typically influenced by others they trust when making purchase decisions. Insufficient accuracy or data sparsity can be avoided by integrating trust relationships as auxiliary information into the rating matrix and mining users' latent information to make accurate recommendations.

The degree of trust between users can be determined by the similarity in behavior. If a user agrees with another's behavior, they will trust them. The user-trust relationship comprises explicit and implicit trust, and the asymmetry of the trust relationship should be considered.

$$w(u_i, u_j) = \begin{cases} p, & u_i, u_j \text{ directly related and weight } p \in [0, 1] \\ \perp, & u_i, u_j \text{ not directly related.} \end{cases} \tag{10}$$

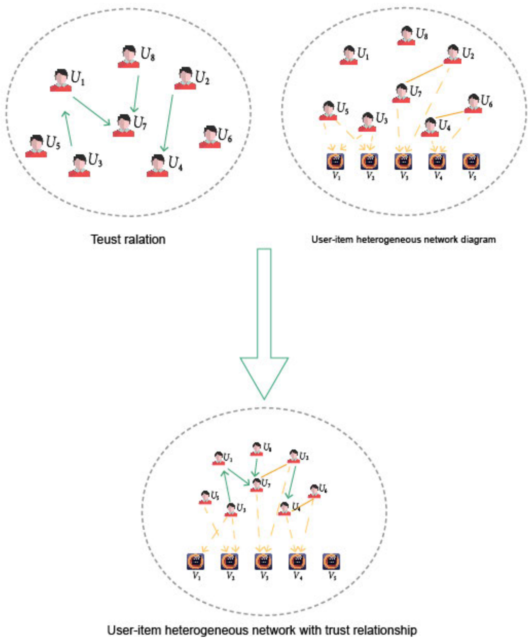


FIGURE 5. Diagram of user-item heterogeneous network with trust relationships. The direct trust relationship between users is an integrated trust relationship.

where the weight value w between users u_i and u_j represents their trust value. If the trust relationship between u_i and u_j is direct, w is 1. If u_i and u_j have no direct trust relationship but have common rating items, their direct trust degree can be calculated using a rating matrix. For example, as shown in Fig. 5, u_3 and u_5 watched the movie v_1 and scored it. If there is neither a direct trust relationship nor common rating items between u_i and u_j, their indirect trust relationship can be established through items to obtain the indirect trust degree.

1) DIRECT TRUST DEGREE

Suppose there are users U and V in the personalized recommender system. If U and V have an explicit trust relationship, the trust degree d_trust_{uv} is 1. If they have no explicit trust relationship but have common rating items, the degree of trust can be derived from the rating matrix. The calculation of the trust degree is based on the possible common rating behaviors of users U and V. The final direct trust degree can be expressed as follows using the Jaccard similarity coefficient

$$d\_trust_{u_i,u_j} = \frac{\left|I_{u_i,u_j}\right|}{\left|N_{u_i} \cup N_{u_j}\right|}, \tag{11}$$

where N_{u_i} is the number of ratings by the user u_i, N_{u_j} is the number of ratings by the user u_j, and I_{u_i,u_j} represents the number of common ratings by users u_i and u_j. The Jaccard similarity coefficient only considers the number of rating items but not the rating values and user rating preferences. For example, some users may particularly like an item and give it three scores, whereas some users give the same item four scores even though they do not like it. The differences between individuals are difficult to measure. Therefore, we propose a direct trust calculation method that integrates the user rating values and rating preferences as follows

$$dtrust_{u_i,u_j} = 1 - \frac{\sum\limits_{v_m \in v_{u_i} \cup v_{u_j}} \left(\left(r_{i,m} - \overline{r_i}\right) - \left(r_{j,m} - \overline{r_j}\right)\right)^2}{\left|I_{i,j}\right|}, \tag{12}$$

where $dtrust_{u_i,u_j}$ represents the direct trust degree between users u_i and u_j; $r_{i,m}$ and $r_{j,m}$ represent the ratings by users u_i and u_j, respectively; v_m is the item that both users u_i and u_j have rated; $\overline{r_i}$ and $\overline{r_j}$ are the average rating values by users u_i and u_j, respectively. The calculation integrates ratings from users u_i and u_j their rating preferences.

The direct trust degree is normalized to a range of [0, 1]:

$$dtrust_{u_i,u_j} = \frac{dtrust_{u_i,u_j} - \min\left(dtrust_{u_i,u_j}\right)}{\max\left(dtrust_{u_i,u_j}\right) - \min\left(dtrust_{u_i,u_j}\right)}. \tag{13}$$

where $\max\left(dtrust_{u_i,u_j}\right)$ represents the maximum degree of trust between users, and $\min\left(dtrust_{u_i,u_j}\right)$ represents the minimum degree of trust between users.

After obtaining the direct trust degree between users, the heterogeneous network based on the user-item rating matrix can be modified. The specific algorithm is shown in Algorithm 1.

Algorithm 1 Establishment of a Heterogeneous Network Incorporating the User's Direct Trust Degree

Input: User set U, item set I, user-item rating matrix R, heterogeneous network $G = V, E, W$, edge set E between users, edge set W between users, list $dtrust$ = [], target user u_i
Output: New heterogeneous network G'
Begin
  while user $u_l \in U$, user $u_l \neq u_i$ **do**
    Calculate $dtrust_{u_i,u_l}$ between user u_l and user u_i according to (13) and store it in $dtrust$ list
  end while
  while $e \in E$, $w \in W$ **do**
    if $(dtrust_{u_i,u_l} \neq 0)$ **then**
      $e_{i,l} \neq 0$ and $w_{i,l} = dtrust_{u_i,u_l}$
    end if
  end while
  return G'
End

By integrating user rating data and trust relationships, the improved heterogeneous network can calculate the direct trust degree between users. This network contains more information than the previous one.

2) INDIRECT TRUST DEGREE

Node link prediction is used for users with neither a direct trust relationship nor common ratings of items with others to establish an indirect trust relationship between the user and user nodes. The random-walk model is used for link prediction. As shown in Fig. 6, Deepwalk is a common random walk model, which can learn the hidden information of the network and represent the nodes in the graph as a vector containing potential information. By drawing lessons from the idea of the algorithm, We first find the target user and take it as the meta-path of the starting node using the heterogeneous information network obtained in the previous section. Representation learning is used to calculate the indirect trust degrees between the target user and other users by calculating the similarity of the node vectors.

Meta-path models are typically used to mine heterogeneous networks. As shown in Fig. 5, users u_2 and u_5, which were not connected, are linked by item node v_3, establishing an implicit trust relationship. However, it is impossible to establish a trust relationship for an isolated node using the trust transitivity or item nodes. Therefore, the meta-path model has the following constraints.

1) All starting nodes of the meta-path are connected, and the breakpoints of the meta-path model are the users.
2) The selection of user nodes takes precedence over that of item nodes.
3) The meta-path includes only node types with a significant impact on the users' rating behaviors.
4) The length of the meta-path does not exceed four nodes.

For example, suppose that in Section IV-B1, we obtain the new heterogeneous network $G' = \{N, E, W\}$, where N is the node-set, E is the edge set, and W is the weight set. The set of the meta-path is $P = \{\rho_1, \rho_2, \rho_3, \cdots, \rho_l, \}$, where ρ_l represents the l-th meta-path; a meta-path can be expressed as $\rho : u_1 \rightarrow u_2 \rightarrow u_3 \rightarrow \cdots \rightarrow u_n$, where u_n is the user node.

To ensure that all meta-paths start from the user nodes, we use the heterogeneous network as the input and obtain ρ meta-paths with length l for each user (non-isolated) node by using the node2vec random walk algorithm. Similar to DeepWalk, the node2vec method uses maximum likelihood estimation to calculate the similarity between nodes based on the jump probability for a certain random walk distance. In the random walk method, node2vec uses a biased random walk rather than an equal probability for the next jump transition.

Suppose that given the current node u_i, the probability of accessing the next fixed node u_j is

$$P\left(c_i = u_j \middle| c_{i-1} = u_i\right) = \begin{cases} \frac{\pi_{u_i u_j}}{Z}, & (u_i, u_j) \in E, \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

The meta-path of the node can be obtained by the node2vec algorithm described in Algorithm 2.

We use the node2vec method to obtain the node sequence and learn the vector representation of each node by representation learning. Then, we calculate the indirect trust degree between users based on the similarity between nodes. DeepWalk and LINE have been used to represent node vectors, but

Algorithm 2 Improved Random Walk Algorithm to Obtain Node Meta-Path

Input: User set U, item set V, user-item rating matrix R, heterogeneous network $G' = \{N, E, W\}$, set E of the edge between users, weight set W of the edge between users, start node r, path length l
Output: Path set walks with the target user u_i as the starting point and friends with indirect trust relationships as the nodes
Begin
  Initialize walks to **Empty**
  for iter = 1 to r **do**
    for all nodes $u_i \in U$ **do**
      walk = node2vecWalk(G', u_i, l)
      Append walk to walks
    end for
  end for
  return walks
End

these methods are only suitable for homogeneous networks. Thus, we use node learning based on user-item bipartite heterogeneous networks and choose the heterogeneous Skip-Gram model [44] to learn node representation in the heterogeneous network.

The heterogeneous Skip-Gram model with a certain window size can maximize the heterogeneous probability of a given node v by inputting a given heterogeneous sequence containing different types of nodes, whose form is expressed as

$$\arg\max_{\theta} \sum_{v \in V} \sum_{c_t \in N_t(V)} \log p\left(c_t \middle| v, \theta\right), \quad (15)$$

where $N_t(v)$ represents the v-th neighborhood of the t-th node. $p\left(c_t \middle| v; \theta\right)$ is usually defined as a softmax function, which can be expressed as

$$p\left(c_t \middle| v, \theta\right) = \frac{e^{X_{ct} \cdot X_v}}{\sum_{u \in V} e^{X_{ct} \cdot X_v}}, \quad (16)$$

where X_v represents the v-th row of the matrix $X \in \mathbb{R}^{|v| \times d}$ and is the embedding vector of node v.

Calculating $p\left(c_t \middle| v; \theta\right)$ directly is time consuming in large networks. We applied the negative sampling technique to learning according to [45]. Given the context N_t and the number of negative samples M, and maximizing the occurrence probability of node, the objective function can be updated as

$$O(\mathbf{X}) = \log \sigma\left(X_{c_t} \cdot X_v\right) + \sum_{m=1}^{M} \mathbb{E}_{u_t^m \sim P_t(u_t)} \left[\log \sigma\left(X_{u_t^m} \cdot X_v\right)\right]. \quad (17)$$

We obtain the final node vector by the gradient descent method and calculate the indirect trust degree between users

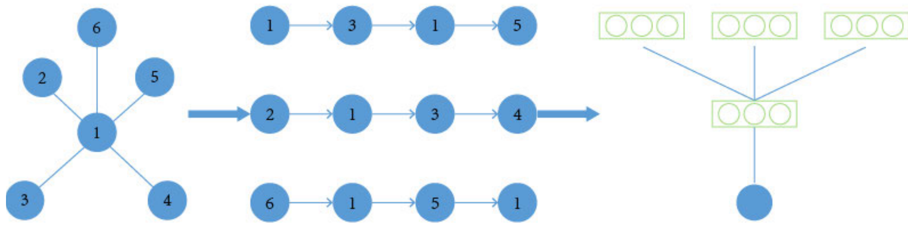


FIGURE 6. Schematic diagram of the DeepWalk method, which is similar to word2vec but uses the co-occurrence relationship between nodes in the graph to learn the vector representation of nodes. Two steps are used in wireless graphs: step 1 performs random walks on the nodes to generate a node sequence; step 2 runs the Skip-Gram model to learn the embedding of each node according to the node sequence obtained in step 1.

using the cosine similarity:

$$i\_trust_{u_i \to u_j} = \frac{\sum_{j=0}^{d-1} \boldsymbol{E}_{u_i}^{j} \cdot \boldsymbol{E}_{u_j}^{j}}{\sqrt{\sum_{j=0}^{d-1} \left(\boldsymbol{E}_{u_i}^{j}\right)^2} \sqrt{\sum_{j=0}^{d-1} \left(\boldsymbol{E}_{u_j}^{j}\right)^2}}, \tag{18}$$

where $i\_trust_{u_i \to v_j}$ is the degree of indirect trust between users, and $E_{u_i}^{j}$ represents the j-th dimension of the implicit vector of the user u_i.

We use (19) and (20) to obtain the normalized direct and indirect trust degrees of the users, with ranges of [0, 1]. We employ weights for the two trust degrees using (21) to obtain the comprehensive trust degree.

$$dtrust_{u_i,u_j} = \frac{dtrust_{u_i,u_j} - \min\left(dtrust_{u_i,u_j}\right)}{\max\left(dtrust_{u_i,u_j}\right) - \min\left(dtrust_{u_i,u_j}\right)}, \tag{19}$$

$$i\_trust_{u_i \to u_j} = \frac{i\_trust_{u_i \to u_j} - \min\left(i\_trust_{u_i \to u_j}\right)}{\max\left(i\_trust_{u_i \to u_j}\right) - \min\left(i\_trust_{u_i \to u_j}\right)}, \tag{20}$$

$$trust_{u_i,u_j} = \lambda trust_{u_i,u_j} + (1-\lambda) i\_trust_{u_i,u_j}. \tag{21}$$

Equation (21) represents the comprehensive degree of trust between users u_i and u_l, where λ is the adjustment parameter. When λ is greater than 0.5, a direct trust relationship occurs; otherwise, an indirect trust relationship exists between users. When $\lambda = 1$, a direct trust relationship occurs. When $\lambda = 0$, an indirect trust relationship exists.

After obtaining the degrees of trust between all users, we derive the user's trust matrix. By normalizing the user's interest similarity matrix, we create $\sum_{u \in N_{u_i}} T_{u_i u_j} = 1$, where N_{u_i} represents the set of users trusted by the user u_i, i.e., $N_u = \left\{u_j \middle| u_j \in U, S_{u_i u_j} > 0\right\}$.

3) CALCULATION OF SIMILARITY DEGREE OF USER PREFERENCES

In addition to trust relationships, the users' preferences determine the choice of items. Many psychology and marketing theories have shown that people's preferences for items mainly depend on their preferences for the corresponding attributes. For example, each movie has important attribute information, such as the movie genre, year, and region, and each attribute has its attribute value. An example of a user's item rating record is listed in Table 3.

TABLE 3. Rating information.

Movie	Genre	Year	Region	Score
Movie A	Romance	The 80s	USA	3
Movie B	Children, animation	The 90s	USA	5
Movie C	Animation	The 80s	UK	4

Suppose u_i represents the i-th user, and A is the item attribute set. There are m attributes in total, and each attribute has a different value. For example, the movie genre has 18 values action, love, disaster, horror, history, science fiction, comedy, suspense, magic, war, adventure, etc, the region includes the United States, the United Kingdom, Japan, and the years include the 80s and the 90s. The item attribute set is $A = \{a_{11}, \cdots, a_{1d}, a_{21}, \cdots, a_{2k}, \cdots, a_{m1}, \cdots, a_{mn}\}$, where a_{mn} represents the n-th value of the m-th attribute of the item. If an item has an attribute value, it is 1; otherwise, it is 0. We can then obtain the movie attribute rating by the user (Table 4).

The preference of user u for a certain attribute value can be calculated as

$$C_{ij} = \frac{Count_{ij}}{sum}, \tag{22}$$

where C_{ij} represents the preference of user u for the j-th value of the i-th attribute of the item. For example, Equation (22) can be used to calculate the preference of user A for action movies. $Count_{ij}$ is the cumulative number of movies watched with an attribute in the movie set, and the sum represents the number of movies the user has watched. However, the rating information is not included. The final preference of user u for the j-th value of the i-th attribute can be expressed as

$$T_{u_i} = p_{u_i} \cdot C_{u_i} \cdot r_{u_i}, \tag{23}$$

where p_{u_i} represents the degree of importance of an attribute, C_{u_i} is the calculated preference for an attribute, and T_{u_i} is the final preference of user u for an attribute. The preference

TABLE 4. Movie attribute rating by the user.

User	Movie	Movie type			Film age			...			Scores
		a_{11}	...	a_{1d}	a_{21}	...	a_{2k}	a_{m1}	...	a_{mk}	
u_1	A	1	0	0	0	0	1	0	0	1	4
	B	0	0	1	1	0	0	1	0	0	4
	C	1	0	0	1	0	0	0	0	1	4
u_2	D	1	0	0	1	0	0	0	0	1	3
	E	1	0	0	0	0	1	0	0	1	4
	F	0	0	1	0	0	1	1	0	0	4
u_3	G	0	0	1	0	0	1	1	0	0	5
	H	0	0	1	1	0	0	1	0	0	3
	I	0	0	1	1	0	0	1	0	0	4

similarity between users can be calculated based on the user preferences for attributes by

$$sim_E(u_i, u_j) = \frac{\sum_{i \in A} T_{u_i,i} \cdot T_{u_j,i}}{\sqrt{\sum_{i \in A} (T_{u_i,i})^2} \sqrt{\sum_{i \in A} (T_{u_j,i})^2}}, \tag{24}$$

where $sim_E(u, v)$ represents the similarity of the explicit preference for user u_i and user u_j, and T_{u_i}, T_{u_j} are the preferences of user u_i and user u_j for attribute i, respectively. The user preference similarity matrix can be obtained based on the preference similarity between all users. Then we normalize the user preference similarity matrix and ensure that $\sum_{v \in B_{ij}} S_{u_i u_j} = 1$ where B_u is the set of users that have similarities with user u, and $B_u = \{u | u \in U, S_{u_i u_j} > 0\}$.

C. CALCULATION OF ITEM CORRELATION

The mining of item correlations is crucial for determining user purchasing decisions [46]. For example, Walmart analyzed users' shopping carts using data mining and obtained the potential correlation between diapers and beer. Retailers can determine which combination of items is purchased frequently by mining the correlation and developing personalized recommendation strategies to ensure precision marketing.

The support and confidence of the correlation are assessed to determine the strength of the correlation between items:

$$support = \frac{|V_{m,n}|}{|N|}, \tag{25}$$

$$confidence(v_m \to v_n) = \frac{V_{m,n}}{N_{v_m}}. \tag{26}$$

Support in (25), refers to the degree of support, $V_{m,n}$ in the user-item rating matrix is the number of people that have rated both items i and j, and N denotes the number of people that have rated the items. $confidence(v_m \to v_n)$ in (26) refers to the confidence of the item $v_m \to v_n$ and equals the quotient of the number of people that have rated items v_m and v_n divided by the number of people that have rated items v_m. The support degree reflects the strength of the correlation between the two items. The confidence degree $confidence(i \to j) \neq confidence(j \to i)$ reflects the strength of the confidence. We can obtain the correlation between items based only on the item rating matrix without using external information as follows:

$$drelation_{v_m \to v_n} = \frac{support(v_m, v_n)}{support(v_m, v_n) + \phi \cdot confidence(v_m \to v_n)}, \tag{27}$$

where $drelation_{v_m \to v_n}$ refers to the direct correlation degree between items v_m and v_n, which depends on the degrees of support and confidence between the items; ϕ is the hyperparameter. When ϕ is greater than 0, the higher the *support*, the greater the correlation degree between items is. When the *support* is low, the direct correlation degree is low, even if the confidence degree is high.

After obtaining the item-item correlation relationship, we can obtain matrix D based on the item correlation. Normalizing matrix D yields $\sum_{n \in B_n} D_{mn} = 1$, where B_n represents the item set associated with item m.

1) THE PROPOSED MODEL

It is assumed that the matrix R includes the ratings of M items by N users. $U \in R^{d \times N}$ and $V \in R^{d \times N}$ represent the potential feature vector matrices of the users and items, respectively, U_u and V_v represent the potential feature vectors of the users and items, respectively, and d represents the dimension of the feature vector. The posterior probability distribution of the implicit vectors of users and items can be expressed by (6).

Inspired by the SocialMF algorithm [25], we integrate the similarity and trust relationships between users into the matrix factorization model using the weighted average method and correct the target user-based feature matrix obtained from their trusted neighbors and similar neighbors as follows

$$\hat{U}_u = \omega_T \sum k \in B_u S_{uk} U_k + \omega_s \sum_{v \in N_u} T_{uv} U_v, \tag{28}$$

where B_u is the set of similar neighbors of users; N_u is the set of users' trusted neighbors; the weight coefficients ω_T and ω_s represent the influence of the user similarity and user trust degree on the rating of the target user, respectively.

The user feature matrix obeys a Gaussian distribution with an average of 0. Its conditional probability is calculated as follows after the matrix has been corrected by the user's similar neighbor set and the user's trust neighbor set and based on the given similarity matrix and trust matrix:

$$\begin{aligned} p(U|T,S,\sigma_U^2,\sigma_W^2) &\propto p(U|\sigma_U^2)p(T,S,\sigma_W^2) \\ &= \prod_{u=1}^{m} N\left(U_u \middle| \omega_T \sum_{k\in B_u} S_{uk}U_k + \omega_s \sum_{v\in N_u} T_{uv}U_v, \sigma_W^2 I\right) \\ &\quad \times \prod_{u=1}^{m} N(U_u|0,\sigma_U^2 I), \end{aligned} \tag{29}$$

where σ_W^2 represents the dispersion of the user feature matrix and the user feature matrix of their trusted friends. Based on the Bayesian inference, we can determine the joint probability distribution of U and V.

$$\begin{aligned} p\left(U,V \middle| R,S,T,\sigma_R^2,\sigma_W^2,\sigma_U^2,\sigma_V^2\right) &\propto p\left(R \middle| U,V,\sigma_R^2\right) \\ &\quad \times p\left(U \middle| S,T,\sigma_W^2,\sigma_U^2\right) p\left(V \middle| \sigma_V^2\right) \\ &= \prod_{u=1}^{m} N\left(U_u \middle| \omega_T \sum_{k\in B_u} S_{uk}U_k + \omega_s \sum_{v\in N_u} T_{uv}U_v, \ \sigma_W^2 I\right) \\ &\quad \times \prod_{r_{u,i}\in R}^{m} \left[N\left(r_{ui} \middle| U_u V_i^T\right), \sigma_R^2\right] \times \prod_{u=1}^{m} N\left(U_u \middle| 0,\sigma_U^2 I\right) \\ &\quad \times \prod_{i=1}^{m} N\left(U_i \middle| 0,\sigma_V^2 I\right). \end{aligned} \tag{30}$$

Similarly, to obtain a higher-quality item feature vector V_i, we incorporate an item incidence matrix in the matrix factorization model. $V \in R^{f\times N}$ and $Z \in R^{f\times N}$ represent the item's implicit and auxiliary feature matrices, respectively, and f represents the dimension of the implicit feature matrix. The condition distribution of the item correlation can be expressed as

$$p(D|V,Z,\sigma_d^2) = \prod_{i=1}^{M}\prod_{j=1}^{M}\left[N\left(D_{ij}\middle| g(V_i^T Z_j),\sigma_d^2\right)\right]^{I_{u_i}^d}, \tag{31}$$

where D_{ij} represents the degree of correlation between items I_i and I_j. The prior Gaussian distribution with a zero-mean value based on the auxiliary feature vector is

$$p(Z|\sigma_Z^2) = \prod_{j=1}^{M} N(Z_j|0,\sigma_Z^2 I). \tag{32}$$

The posterior Bayesian probability of the item's implicit feature vector is defined as follows using (32) and the item's incidence matrix D:

$$\begin{aligned} p(V,Z|D,\sigma_D^2,\sigma_V^2,\sigma_Z^2) &\propto p(D|V,Z,\sigma_D^2) \\ &\quad \times p(V|\sigma_V^2)p(Z|\sigma_Z^2) \\ &= \prod_{i=1}^{M}\prod_{j=1}^{N}[N(D_{ij}|g(V_i^T Z_j),\sigma_D^2)]^{I_{ij}^D} \\ &\quad \times \prod_{i=1}^{M} N(V_i|0,\sigma_V^2 I) \times \prod_{j=1}^{N} N(Z_j|0,\sigma_Z^2 I). \end{aligned} \tag{33}$$

The proposed model has been established based on user-item ratings, item correlations, user trust relationships, and similarity relationships. The posterior Bayesian probability of the model can be defined as

$$\begin{aligned} p(U,V,Z|R,S,T,D,\sigma_R^2,\sigma_W^2,\sigma_D^2,\sigma_U^2,\sigma_V^2) &\propto p(R|U,V,\sigma_R^2)p(U|S,T,\sigma_W^2,\sigma_U^2) \\ &\quad \times p(D|V,Z,\sigma_D^2)p(V|\sigma_V^2)p(Z|\sigma_Z^2) \\ &= \prod_{u=1}^{m} N\left(U_u \middle| \omega_T \sum_{k\in B_u} S_{uk}U_k + \omega_s \sum_{v\in N_u} T_{uv}U_v, \ \sigma_W^2 I\right) \\ &\quad \times \prod_{r_{u,i}\in R}^{m}\left[N\left(r_{ui}\middle| U_u V_i^T\right),\sigma_R^2\right] \\ &\quad \times \prod_{i=1}^{M}\prod_{j=1}^{N}[N(D_{ij}|g(V_i^T Z_j),\sigma_D^2)]^{I_{ij}^D} \\ &\quad \times \prod_{u=1}^{m} N\left(U_u|0, \ \sigma_U^2 I\right)\prod_{i=1}^{n} N\left(V_i|0, \ \sigma_V^2 I\right) \\ &\quad \times \prod_{j=1}^{N} N\left(Z_j|0, \ \sigma_Z^2 I\right). \end{aligned} \tag{34}$$

We describe the target function as follows by maximizing the probability model:

$$\begin{aligned} L(R,U,V,T,S,D) &= \frac{1}{2}\sum_{u=1}^{M}\sum_{i=1}^{N} I_{u_i}(r_{ui} - g(U_u^T V_i))^2 \\ &\quad + \frac{\lambda_u}{2}\sum_{u=1}^{M}\|U_u\|^2 + \frac{\lambda_v}{2}\sum_{i=1}^{N}\|V_i\|^2 \\ &\quad + \frac{\lambda_w}{2}\sum_{u=1}^{M}\|U_u - \hat{U}_u\|^2 + \frac{\lambda_z}{2}\sum_{i=1}^{N}\|Z\|_F^2, \end{aligned} \tag{35}$$

The minimum solution to the above target function can be obtained using the stochastic gradient descent method:

$$\begin{aligned} \frac{\partial L}{\partial U_u} &= \sum_{r_{ui}\in R}\left(r_{u,i} - U_u^T V_i\right)V_i + \lambda_u U_u + \lambda_w\left(U_u - \hat{U}_u\right) \\ &\quad - \varpi_i(u)\lambda_w \sum_{\{v|v\in N_U\}} T_{u,v}\left(U_u - \hat{U}_u\right) \\ &\quad - \varpi_i(u)\lambda_w \sum_{\{v|v\in B_U\}} S_{u,v}\left(U_u - \hat{U}_u\right), \end{aligned} \tag{36}$$

$$\frac{\partial L}{\partial V_i} = \sum_{u=1}^{N} I_{ui}^R U_u g'\left(U_u^T V_i\right)\left(g\left(U_u^T V_i\right) - R_{ui}\right) + \lambda_v U_i$$

Algorithm 3 The Proposed Algorithm

Input: Rating matrix R, trust matrix T weighted with trust relationship, item incidence matrix D, preference similarity matrix S of the user for items, λ_w, λ_s, λ_z, learning rate α, maximum of iterations, *threshold*
Output: User implicit feature matrix U, item implicit feature matrix V
Initialize λ_w, λ_s, λ_z, learning rate α, U, and V
Calculate user trust degree by (21) to obtain trust matrix T
Calculate the similarity of interest between users by (24) to obtain user interest similarity matrix S
Calculate item correlation degree by (27) to obtain incidence matrix D
while ($i < MaxNum$) **do**
  for $r_{i,j} \in R$ **do**
    Update $U_u \leftarrow U_u - \alpha \frac{\partial L}{\partial U_u}$ based on (36)
    Update $V_i \leftarrow V_i - \alpha \frac{\partial L}{\partial V_i}$ based on (37)
    Update $Z_j \leftarrow Z_j - \alpha \frac{\partial L}{\partial Z_j}$ based on (38)
  end for
  Calculate the new target function $Lnew : L \leftarrow Lnew$ according to (36)
  if $Lnew : L \leftarrow threshold$ **then**
    break
  end if
  $t \leftarrow t + 1$
end while
Output U, V, Z
Calculate the rating of item i by user u according to $\hat{r}_{ui} = \sum_{k=1}^{K} u_{uk} v_{ki}$

$$+\lambda_s \sum_{i=1}^{M} I_{ij}^{S} Z_j g'\left(U_u^T V_i\right)\left(g\left(U_u^T V_i\right) - S_{ij}\right), \quad (37)$$

$$\frac{\partial L}{\partial Z_j} = \lambda_S \sum_{i=1}^{M} I_{ij}^{S} V_i g'\left(Z_j^T V_i\right)\left(g\left(Z_j^T V_i\right) - S_{ij}\right) + \lambda_Z Z_j, \quad (38)$$

where $g'(x)$ is the derivative of the logistic function. The proposed algorithm is described in Algorithm 3.

The items requiring long calculation times in the TIAPMP algorithm include the objective function L and the gradient descent function. The time complexity for calculating the objective function L is $O\big(d|R^{tr}|+d|T|+d|S|+d|D|\big)$, where $|T|$ is the number of trust relationships, $|S|$ is the number of similarity relationships, and $|D|$ is the number of correlation relationships. The complexity for calculating one iteration of the gradient is $O\big(d|R^{tr}|\bar{r} + d|T| + d|S| + d|D|\big)$, where $\bar{r}$ represents the average number of item ratings. Since $\bar{r} \ll \big(|R^{tr}|, |T|, |S|, |D|\big)$, the overall complexity of the algorithm is linearly related to the number of ratings, the number of trust relationships, the number of similar relationships, and the number of correlations.

TABLE 5. Statistics of the datasets.

Information	FilmTrust	CiaoDVD
User quantity	1,508	17,615
Item quantity	2,071	16,121
Rating record	35,497	72,665
Social relationship	1,853	40,133

V. EXPERIMENTAL DESIGN AND RESULTS

A. DATASET AND EXPERIMENTAL ENVIRONMENT

We used two public datasets, FilmTrust [20] and CiaoDVD [47], to verify the influence of different factors on personalized recommendation performances. Both datasets contain trust relationships and rating information. The difference is that the rating range is [0.5, 4] in the FilmTrust dataset and [1, 5] in the CiaoDVD dataset.

The statistics of the datasets used in the experiment are listed in Table 5.

We divide the dataset randomly into a training set and a test set with a ratio of 4:1. The training set is used to learn the parameters of the recommendation algorithm, and the test set is used to evaluate the accuracy of the algorithm. The experimental software environment is Windows 10 (64 bits), Anaconda 3, and Python 3.7. The hardware environment is a six-core CPU with an Intel Core i7-8750H @ 2.20 GHz, and the internal storage is 16 GB.

B. EVALUATION METRICS

The purpose of the personalized recommendation is to use algorithms to help enterprises obtain greater benefits by recommending items that users like. Therefore, recommendation accuracy is critical for evaluating recommender systems. It can be measured from two aspects: the first is the *Top-N* ranking of the recommendation results obtained from the precision rate ($P@N$) and recall rate ($R@N$). The second is the rating prediction, i.e., the mean absolute error (MAE) and root mean squared error (RMSE). In addition to accuracy evaluations, diversity has been commonly usually used in personalized recommendation systems to measure personalized characteristics. The evaluation metrics are defined as follows.

1) METRICS TO MEASURE *Top-N* RANKING OF RECOMMENDER SYSTEMS

The $P@N$ is the ratio of correctly predicted samples to the total number of samples. In personalized recommendation, $P@N$ refers to the ratio of successfully recommended items to all recommended items:

$$P@N = \frac{\sum_u P(u) \cap T(u)}{\sum_u |P(u)|}, \quad (39)$$

where $P(u)$ denotes all recommended items and $T(u)$ denotes the items in the test set.

The $R@N$ is the ratio of correctly classified positive samples to the actual positive samples. In personalized recommendation, $R@N$ describes the ratio of items recommended successfully to all items that should be recommended:

$$R@N = \frac{\sum_u P(u) \cup T(u)}{\sum_u |T(u)|}, \tag{40}$$

where R_u represents the item set recommended to the user U_u, and T_u represents the set of items liked by the user U_u.

2) METRICS TO MEASURE RATING PREDICTION OF RECOMMENDER SYSTEMS

The *MAE* and *RMSE* are used to describe the degree of deviation between the predicted and actual scores. They are expressed as

$$MAE = \frac{\sum\limits_{(u,i)\in R_{test}} \left|r_{ui} - \hat{r}_{ui}\right|}{|R_{test}|}, \tag{41}$$

$$RMSE = \sqrt{\frac{\sum\limits_{(u,i)\in R_{test}} \left|r_{ui} - \hat{r}_{ui}\right|}{|R_{test}|}}, \tag{42}$$

where R_{test} represents the user-item set in the test set; $|R_{test}|$ represents the number of elements in the test set.

When the values of *MAE* and *RMSE* are small, the error between the predicted and actual scores is small, and the accuracy of the algorithm is high.

3) METRICS TO MEASURE *DIVERSITY* OF RECOMMENDER SYSTEMS

Since users have diverse interests, *diversity* has been used to describe the performances of recommender systems. It is defined as

$$Diversity = 1 - \frac{\sum_{i,j\in R(u), i\neq j} s(i,j)}{\frac{1}{2}|R(u)|\big(|R(u) - 1|\big)}, \tag{43}$$

where $s(i, j)$ represents the degree of similarity between items i and j; $R(u)$ represents the recommendation list for user u.

4) COMPARISON ALGORITHM AND HYPER-PARAMETER SETTING

We compare the following five popular recommendation algorithms to verify the influences of the trust relationship and the correlation on the model performance.

1) BasicMF is a personalized recommendation algorithm based on matrix factorization proposed by Koren et al. [15].
2) SoReg is a social recommendation algorithm proposed by Ma et al. [26]. It uses social regularization as the social constraint of the recommender system and stipulates that users have similar feature vectors as users they trust and that the similarity degree of feature vectors depends on the user's trust degree in others.
3) SociaMF is an algorithm proposed by Jamali and Ester [25]. It considers social trust relationships in PMF and the influences of direct and indirect trust relationships on the recommendation performance.
4) TrustMF is an algorithm proposed by Yang et al. [23] that considers the influence of the correlation between trusted users on the recommendation performance.
5) SVD++ is an algorithm proposed by Koren [48]. It considers user and item bias information and user implicit feedback information predicting ratings and has high recommendation accuracy.
6) ReHI is an algorithm proposed by Li et al. [53]. In this paper,authors consider the user and item bias information and user implicit feedback information predicting ratings and has high recommendation accuracy.

We use λ_u to represent the hyperparameter for matrix U operation and λ_v for matrix V operation. Since we find only a hyperparameter λ in [15] then we employ $\lambda = \lambda_u = \lambda_v$. We use the rest as it is.Table 6 lists the optimal hyperparameters of the five recommendation algorithms in the datasets.

In order to keep the consistency with other comparative experiments, the performance of each model is verified when the eigenvector dimension f is 5 and 10 respectively, and a large number of experiments are carried out to find the optimal parameters of each model. Except for the learning rate of 0.01, the configuration of other parameters is shown in Table 6.

C. COMPARISION OF Top-N RANKING PERFORMANCES

We used the $P@N$ and $R@N$ to compare the *Top-N* ranking performances of the algorithms based on the public datasets FilmTrust and CiaoDVD for different N values (5 and 10). The results are listed in Table 7. The performances of SoReg, SociaMF, and TrustMF, which consider social relationships, are better than that of BasicMF, indicating that social relationships can help improve the recommendation performance. Our proposed algorithm considers the direct and indirect trust degrees, providing more information on the potential social relationships between users and resulting in the best recommendation performance. Since it also considers the item correlations, it can accurately describe user preferences.

D. RATING PREDICTION ACCURACY FOR DIFFERENT FEATURE VECTOR DIMENSIONS

To verify the accuracies of the algorithms in cold-start and normal states, we divide the dataset into the Warm set and the Cold set. If the user's rating records do not exceed 5 in the training set, the records in the test set belong to the Warm set. Otherwise, they belong to the Cold set.

Table 8 lists the *MAE* and *RMSE* of different algorithms in the Cold set, and Table 9 shows the results for the Warm set for feature vector dimensions of $d = 5$ and 10.

Table 9 shows that, The recommendation performance of SVD++ is significantly higher than that of BasicMF in the Warm set. It can be concluded that combining the user and item bias factors and the implicit feedback information improves the rating prediction accuracy of the recommender system. In addition, the recommendation performances of SoReg, SociaMF, and TrustMF show slight differences but

TABLE 6. Hyper-parameter settings of the recommendation algorithms.

Algorithm	FilmTrust	CiaoDVD
PMF	$\lambda_u = \lambda_v = 0.1$	$\lambda_u = \lambda_v = 0.1$
SoReg	$\lambda_u = \lambda_v = 0.001, \beta = 0.1$	$\lambda_u = \lambda_v = 0.001, \beta = 0.1$
SociaMF	$\lambda_u = \lambda_v = 0.001, \lambda_T = 5$	$\lambda_u = \lambda_v = 0.001, \lambda_T = 1$
TrustMF	$\lambda_u = \lambda_v = 0.001, \lambda_T = 1$	$\lambda_u = \lambda_v = 0.001, \lambda_T = 1$
SVD++	$\lambda_b = \lambda_u = \lambda_v = 0.1$	$\lambda_b = \lambda_u = \lambda_v = 0.1$
Our method	$\lambda_u = \lambda_v = 0.1, \lambda_w = 3, \lambda_z = 1$	$\lambda_u = \lambda_v = 0.1, \lambda_w = 3, \lambda_z = 1$

TABLE 7. Performance comparison of ***TOP-N*** rankings.

Dataset	Metrics	BasicMF	SoReg	SociaMF	TrustMF	SVD++	ReHI	Our method
FilmTrust	P@5	0.362	0.381	0.384	0.383	0.375	0.384	0.386
	P@10	0.358	0.372	0.376	0.374	0.368	0.372	0.379
	R@5	0.281	0.295	0.292	0.293	0.296	0.297	0.299
	R@10	0.283	0.299	0.296	0.298	0.299	0.302	0.306
CiaoDVD	P@5	0.353	0.371	0.376	0.372	0.365	0.371	0.377
	P@10	0.349	0.363	0.367	0.362	0.361	0.365	0.371
	R@5	0.292	0.285	0.282	0.283	0.285	0.286	0.288
	R@10	0.293	0.289	0.286	0.288	0.288	0.292	0.296

TABLE 8. Performance comparison on cold user dataset.

Dataset	Dim	Metrics	BasicMF	SoReg	SociaMF	TrustMF	SVD++	ReHI	Our method
FilmTrust	5	MAE	0.905	0.661	0.688	0.666	0.692	0.684	0.602
		RMSE	1.171	0.853	0.912	0.864	0.915	0.906	0.881
	10	MAE	0.861	0.659	0.672	0.664	0.691	0.688	0.601
		RMSE	1.107	0.849	0.905	0.858	0.913	0.911	0.778
CiaoDVD	5	MAE	1.498	0.767	0.755	0.746	0.734	0.728	0.712
		RMSE	1.855	0.981	0.955	0.952	0.975	0.951	0.948
	10	MAE	1.138	0.715	0.729	0.719	0.728	0.720	0.711
		RMSE	1.431	1.069	0.957	0.951	0.968	0.945	0.942

TABLE 9. Performance comparison on warm user dataset.

Dataset	Dim	Metrics	BasicMF	SoReg	SociaMF	TrustMF	SVD++	ReHI	Our method
FilmTrust	5	MAE	0.735	0.654	0.614	0.612	0.841	0.758	0.598
		RMSE	0.969	0.858	0.815	0.852	1.015	0.992	0.698
	10	MAE	0.753	0.645	0.623	0.617	0.811	0.765	0.596
		RMSE	0.988	0.856	0.833	0.802	1.064	1.157	0.693
CiaoDVD	5	MAE	1.383	0.743	0.562	0.561	0.713	0.711	0.705
		RMSE	1.468	1.167	1.234	0.975	1.153	0.985	0.845
	10	MAE	1.071	0.774	0.729	0.758	0.942	0.721	0.704
		RMSE	1.369	0.996	0.977	0.954	1.082	0.960	0.935

are higher than those of BasicMF and SVD++, demonstrating that incorporating the trust relationship into matrix factorization increases the prediction accuracy.

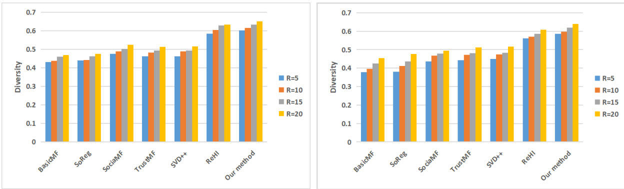
Table 8 shows that, The prediction accuracy of BasicMF is relatively low in the Cold set because there are few user's rating records, making it difficult for BasicMF to learn the accurate feature vector of the user. The accuracies of SoReg, SociaMF, and TrustMF are higher than that of BasicMF, because integrating the trust relationship in matrix factorization alleviates the cold-start problem. However, there are few user trust relationships in the Cold set, and the implicit vectors learned by the model does not reflect the actual relationships between users. The noise level is high during parameter learning, resulting in low performance. The proposed method outperforms the other methods because it considers the relationships between items. The results of the cold start scenario show that the user's trust relationship has a limited positive influence on the algorithm. However, since the item correlations are considered, the algorithm provides high rating prediction accuracy.

The accuracy of all models is lower for the CiaoDVD dataset than for the Filmtrust dataset. The reason is that the number of user's ratings is significantly larger in the FilmTrust dataset than in the CiaoDVD dataset.

The proposed method achieves the highest accuracies on the Warm and Cold datasets. The reason is that it considers direct and indirect trust relationships, forming a relatively dense trust network and alleviating the cold start problem. The algorithm comprehensively considers the user's trust, similarity, and item correlations during matrix decomposition, ensuring the accuracy of the feature vectors learned in the training phase and achieving high recommendation accuracies on both datasets.

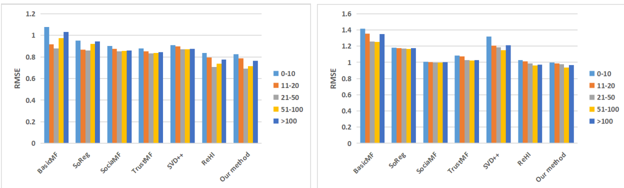
E. COMPARISON OF DIVERSITY METRIC

Diversity reflects the ability of a recommender system to mine the potential interests of a user. The greater the diversity value, the more types of items exist in the system. The comparison of the diversity metric of the algorithms for different numbers of recommended items (R) on the FilmTrust



(a) (b)

FIGURE 7. **Comparison of diversity metric of algorithms for different numbers of recommended items (R). (a) FilmTrust dataset, (b) CiaoDVD dataset.**



(a) (b)

FIGURE 8. **Influences of the number of trust relationships on recommendation performance. (a) FilmTrust dataset, (b) CiaoDVD dataset.**

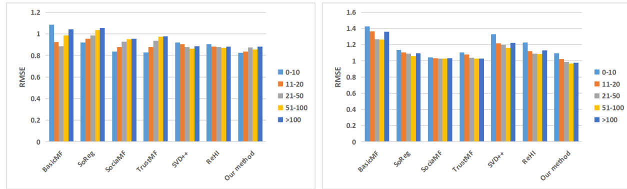
and CiaoDVD datasets is shown in Fig. 7. The algorithms that consider social relationships have higher diversity than the traditional matrix factorization model. As R increases, the diversity values of all algorithms increase; our method exhibits the best diversity performance. The reason is that our method considers the explicit and implicit trust relationships, providing more potential friends for the user and mining the potential interest of the user.

F. INFLUENCE OF THE NUMBER OF USER RATINGS ON RECOMMENDATION PERFORMANCE

To compare the recommendation performances of the algorithms with different rating numbers, we divide the user rating number into 5 groups: [0:10], [11:20], [21:50], [51:100], and over 100. The *RMSE*s of the algorithms with different rating numbers are shown in Fig. 8.

As the number of user rating increases, the recommendation performance of all algorithms shows an upward trend (the *RMSE* shows a downward trend); however, this trend is not maintained until the end. The *RMSE* begins to increase when the number of user ratings exceeds 100, and the recommendation performance begins to decrease. The reason is that the model cannot learn the user's potential feature vector when the number of user ratings is small. When the number increases to a certain value and the user's interest becomes more diverse, it is difficult for the model to learn the potential feature vectors corresponding to the user's extensive interests.

Our method exhibits better recommendation performance than the other algorithms, regardless of the sample size. When the sample size is small, there are few rating data, with limited information on social relationships. Our method can integrate user information and item correlation, improving recommendation accuracy. As the sample size increases, the



(a) (b)

FIGURE 9. **Influences of the number of trust relationships on recommendation performance. (a) FilmTrust dataset, (b) CiaoDVD dataset.**

proposed algorithm adjusts the parameters, such as the user trust relationship, user similarity relationship, and item correlation, to learn the optimized parameters and improve the recommendation accuracy.

G. INFLUENCE OF THE NUMBER OF TRUST RELATIONSHIPS ON RECOMMENDATION PERFORMANCE

We divide the number of user connections into 5 groups: [0:10], [11:20], [21:50], [51:100], and over 100. So we can compare the influence of the number of trust relationships on the recommendation performance. The RMSE of the algorithms with different user connections is shown in Fig. 9.

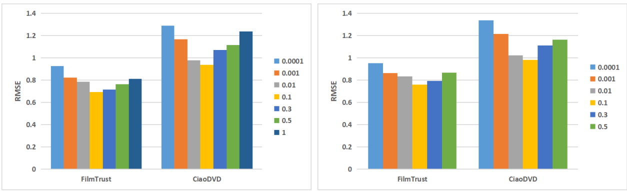
As the number of user connections increases, the recommendation performances of the recommendation models integrating social relationships increase, decrease and stabilize sequentially. When there are few rating data, the trust relationships can help the model learn the user's potential feature vector. As the number of trust relationships increases, the users' interests are affected by their trusted friends and become complex and diverse. It is not possible to learn the real potential feature vectors of the users because of an increase in noise. However, in the FilmTrust dataset, the RMSE value of the recommendation models with social relationships increases with an increase in the number of trust relationships due to an increase in noise.

The RMSE trends of the BasicMF and SVD++ are not affected by the number of social relationships because these algorithms do not consider them. As the number of social relationships increases, the user rating data increase, improving the recommendation performances of the two algorithms.

The recommendation performance of the proposed method exceeds those of the other algorithms on the FilmTrust and CiaoDVD data sets because it considers many factors, such as the user rating, the user's explicit and implicit trust information, and item correlations.

H. INFLUENCE OF HYPER-PARAMETER SETTINGS ON RECOMMENDATION PERFORMANCE

The influences of the hyper-parameter settings λ_w and λ_z on the proposed model are evaluated. The hyper-parameter λ_w controls the influence of the user trust relationship on the recommendation results. Its values are 0.0001, 0.001, 0.01, 0.1, 0.3, 0.5, and 1. The hyper-parameter λ_z controls the influence of the item correlation on the recommendation results; its values are 0.0001, 0.001, 0.01, 0.1, 0.3, and 0.5, respectively.



(a) (b)

FIGURE 10. Influence of hyper-parameters. (a) λ_W, (b) λ_Z.

Users are more likely affected by other users if the λ_w value is higher. As shown in Fig. 10(a), λ_w has a significant effect on the RMSE. Initially, the RMSE decreases with an increase in the hyper-parameter λ_w, indicating that the trust relationships between users influence user behavior when the rating data are insufficient. As λ_w increases, the RMSE rises, demonstrating that the user does not always rely on trusted friends and the trust relationship has limited influence on user ratings. As shown in Fig. 10(b), when λ_z increases from 0.00001, the RMSE value exhibits a downward trend, and when it reaches the threshold, its value begins to rise. This finding shows that the item correlation significantly influences the recommendation results when there are few rating data, affecting user decision-making.

VI. CONCLUSION

We proposed a PMF-based recommendation model that integrates the user trust relationship, user similarity, and item correlation to alleviate the cold-start problem and prevent low recommendation accuracy due to data sparsity. The DeepWalk method is used in the model to determine the direct and indirect trust relationships between users in the user-item heterogeneous network. In addition to calculating the interest similarity between users, the proposed model considers the preference degree of the user for the item attributes, facilitating the calculation of the user interest differences. the proposed model also considers the item correlation to increase the diversity of the recommendation system. The integration of the user trust relationship, user similarity, and item correlation enables the prediction of user ratings of items using a PMF model. The results of experiments on the FilmTrust and CiaoDVD datasets demonstrate the high accuracy and robustness of the proposed method.

Since user trust relationships and users' interests change over time, we aim to establish a more appropriate model in a future study to predict item ratings by users to provide them with more accurate personalized item recommendations.

In addition, the cold-start problem is a challenge in personalized recommendation systems. In a future study, we will focus on this problem by establishing a more appropriate model to minimize this issue. For example, we will examine the integration of user information (user demographic information), item information (especially visual features of items, knowledge graph of objects), situational information (time, place, etc.), user networks, commodity networks, and other important information, into appropriate in-depth representation models to learn the user's preference for cold-start items and address this problem. Federated learning has been widely used in recommendation systems and achieved good performance [49], [50]. Thus, we aim to combine federated learning with PMF to establish a new recommendation model with improved accuracy.

REFERENCES

[1] A. Edmunds and A. Morris, "The problem of information overload in business organisations: A review of the literature," *Int. J. Inf. Manage.*, vol. 20, no. 1, pp. 17–28, Feb. 2000.
[2] M. Ge and F. Persia, "A survey of multimedia recommender systems: Challenges and opportunities," *Int. J. Semantic Comput.*, vol. 11, no. 3, pp. 411–428, Sep. 2017.
[3] M. Kunaver and T. Požrl, "Diversity in recommender systems—A survey," *Knowl.-Based Syst.*, vol. 123, pp. 154–162, May 2017.
[4] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2010, pp. 1–35.
[5] A. Said and V. Torra, "Data science: An introduction," in *Studies in Big*, Cham, Switzerland: Springer, 2018, pp. 1–6.
[6] C. A. Gomez-Uribe and N. Hunt, "The Netflix recommender system," *ACM Trans. Manage. Inf. Syst.*, vol. 6, no. 4, pp. 1–19, Jan. 2016.
[7] J. Davidson, B. Livingston, D. Sampath, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, and M. Lambert, "The YouTube video recommendation system," in *Proc. RecSy*, Barcelona, Spain, Sep. 2010, pp. 293–296.
[8] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, pp. 1–19, Jan. 2009.
[9] J. H. Su and T. W. Chiu, "An item-based music recommender system using music content similarity," in *Intelligent Information and Database Systems*. Berlin, Germany: Springer, 2016, pp. 179–190.
[10] W. Li, X. Zhou, S. Shimizu, M. Xin, J. Jiang, H. Gao, and Q. Jin, "Personalization recommendation algorithm based on trust correlation degree and matrix factorization," *IEEE Access*, vol. 7, pp. 45451–45459, 2019.
[11] G. Guo, J. Zhang, and D. Thalmann, "Merging trust in collaborative filtering to alleviate data sparsity and cold start," *Knowl.-Based Syst.*, vol. 57, pp. 57–68, Feb. 2014.
[12] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, Hong Kong, May 2001, pp. 285–295.
[13] K. Yoon, J. Lee, and M. U. Kim, "Music recommendation system using emotion triggering low-level features," *IEEE Trans. Consum. Electron.*, vol. 58, no. 2, pp. 612–618, May 2012.
[14] S. K. Lee, Y. H. Cho, and S. H. Kim, "Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations," *Inf. Sci.*, vol. 180, no. 11, pp. 2142–2155, Jun. 2010.
[15] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
[16] M. G. Manzato, "Discovering latent factors from movies genres for enhanced recommendation," in *Proc. RecSys*, Dublin, Ireland, Sep. 2012, pp. 249–252.
[17] J. Qin, L. Cao, and H. Peng, "Collaborative filtering recommendation algorithm based on weighted item category," in *Proc. CCDC*, Yinchuan, China, May 2016, pp. 2782–2786.
[18] C. Zhang, L. Yu, Y. Wang, C. Shah, and X. Zhang, "Collaborative user network embedding for social recommender systems," in *Proc. SDM*, Houston, TX, USA, Apr. 2017, pp. 381–389.
[19] P. Avesani, P. Massa, and R. Tiella, "A trust-enhanced recommender system application," in *Proc. ACM Symp. Appl. Comput. (SAC)*, Santa Fe, NM, USA, 2005, pp. 1589–1593.
[20] J. Golbeck and J. Hendler, "FilmTrust: Movie recommendations using trust in web-based social networks," in *Proc. CCNC*, Las Vegas, NV, USA, Jan. 2006, pp. 93–104.
[21] M. Jamali and M. Ester, "TrustWalker: A random walk model for combining trust-based and item-based recommendation," in *Proc. KDD*, Paris, France, Jun. 2009, pp. 397–406.

[22] H. Ma, H. Yang, M. R. Lyu, and I. King, ''SoRec: Social recommendation using probabilistic matrix factorization,'' in *Proc. 17th ACM Conf. Inf. Knowl. Mining (CIKM)*, Napa Valley, CA, USA, 2008, pp. 931–940.
[23] B. Yang, Y. Lei, J. Liu, and W. Li, ''Social collaborative filtering by trust,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1633–1647, Aug. 2017.
[24] B. Tian, H. Yang, and J. Fang, ''Recommendation algorithm based on probability matrix factorization and fusing trust,'' *J. Comput. Appl.*, vol. 39, no. 10, pp. 2834–2840, 2019.
[25] M. Jamali and M. Ester, ''A matrix factorization technique with trust propagation for recommendation in social networks,'' in *Proc. 4th ACM Conf. Recommender Syst. (RecSys)*, Barcelona, Spain, 2010, pp. 135–142.
[26] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, ''Recommender systems with social regularization,'' in *Proc. 4th ACM Int. Conf. Web Search Data Mining (WSDM)*, Hong Kong, 2011, pp. 287–296.
[27] B. Perozzi, R. Al-Rfou, and S. Skiena, ''DeepWalk: Online learning of social representations,'' in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2014, pp. 701–710.
[28] A. Grover and J. Leskovec, ''Node2vec: Scalable feature learning for networks,'' in *Proc. KDD*, San Francisco, CA, USA, Aug. 13, 2016, pp. 855–864.
[29] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, ''LINE: Large-scale information network embedding,'' in *Proc. 24th Int. Conf. World Wide Web*, Florence, Italy, May 2015, pp. 1067–1077.
[30] D. Wang, P. Cui, and W. Zhu, ''Structural deep network embedding,'' in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 1225–1234.
[31] Y. Dong, N. V. Chawla, and A. Swami, ''Metapath2vec: Scalable representation learning for heterogeneous networks,'' in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Halifax, NS, Canada, Aug. 2017, pp. 135–144.
[32] M. Gao, L. Chen, X. He, and A. Zhou, ''BiNE: Bipartite network embedding,'' in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Ann Arbor, MI, USA, Jun. 2018, pp. 715–724.
[33] M. Gao, X. He, L. Chen, T. Liu, J. Zhang, and A. Zhou, ''Learning vertex representations for bipartite networks,'' *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 379–393, Jan. 2022.
[34] J. Sybrandt and I. Safro, ''FOBE and HOBE: First- and high-order bipartite embeddings,'' in *Proc. MLG*, New York, NY, USA, Aug. 24, 2019, pp. 1–8.
[35] Y. Li, W. Chen, and H. Yan, ''Learning graph-based embedding for time-aware product recommendation,'' in *Proc. ACM Conf. Inf. Knowl. Manage.*, Singapore, Nov. 2017, pp. 2163–2166.
[36] H. Wu, H. Zhang, P. He, C. Zeng, and Y. Zhang, ''A hybrid approach to service recommendation based on network representation learning,'' *IEEE Access*, vol. 7, pp. 60242–60254, 2019.
[37] W. X. Zhao, J. Huang, and J. R. Wen, ''Learning distributed representations for recommender systems with a network embedding approach,'' in *Information Retrieval Technology*. Cham, Switzerland: Springer, 2016, pp. 224–236.
[38] X. Zheng, Y. Luo, L. Sun, and F. Chen, ''A new recommender system using context clustering based on matrix factorization techniques,'' *Chin. J. Electron.*, vol. 25, no. 2, pp. 334–340, Mar. 2016.
[39] V. Faridani, M. V. Jahan, and M. Jalali, ''Combining trust in collaborative filtering to mitigate data sparsity and cold-start problems,'' in *Proc. 4th Int. Conf. Comput. Knowl. Eng. (ICCKE)*, Mashhad, Iran, Oct. 2014.
[40] H. Ma, I. King, and M. R. Lyu, ''Learning to recommend with social trust ensemble,'' in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, Boston, MA, USA, 2009, pp. 203–210.
[41] X. Ren, M. Song, E. Haihong, and J. Song, ''Context-aware probabilistic matrix factorization modeling for point-of-interest recommendation,'' *Neurocomputing*, vol. 241, pp. 38–55, Jun. 2017.
[42] H. Parvina, P. Moradi, S. Esmaeilib, and M. Jalilic, ''An efficient recommender system by integrating non-negative matrix factorization with trust and distrust relationships,'' in *Proc. DSW*, Lausanne, Switzerland, Jun. 2018, pp. 135–139.
[43] S. Xu, H. Zhuang, F. Sun, S. Wang, T. Wu, and J. Dong, ''Recommendation algorithm of probabilistic matrix factorization based on directed trust,'' *Comput. Electr. Eng.*, vol. 93, Jul. 2021, Art. no. 107206.
[44] J. Tang, M. Qu, and Q. Mei, ''PTE: Predictive text embedding through large-scale heterogeneous text networks,'' in *Proc. KDD*, Sydney, NSW, Australia, Aug. 2015, pp. 1165–1174.
[45] C. Luo, W. Pang, Z. Wang, and C. Lin, ''Hete-CF: Social-based collaborative filtering recommendation using heterogeneous relations,'' in *Proc. IEEE Int. Conf. Data Mining*, Shenzhen, China, Dec. 2014, pp. 917–922.
[46] Z. Zhang, G. Xu, P. Zhang, and Y. Wang, ''Personalized recommendation algorithm for social networks based on comprehensive trust,'' *Appl. Intell.*, vol. 47, no. 3, pp. 659–669, Apr. 2017.
[47] F. Yuan, L. Yao, and B. Benatallah, ''Adversarial collaborative neural network for robust recommendation,'' in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Paris, France, Jul. 2019, pp. 1065–1068.
[48] Y. Koren, ''Factorization meets the neighborhood,'' in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Las Vegas, NV, USA, 2008, pp. 426–434.
[49] O. A. Wahab, G. Rjoub, J. Bentahar, and R. Cohen, ''Federated against the cold: A trust-based federated learning approach to counter the cold start problem in recommendation systems,'' *Inf. Sci.*, vol. 601, pp. 189–206, Jul. 2022.
[50] Z. Teimoori, A. Yassine, and M. S. Hossain, ''A secure cloudlet-based charging station recommendation for electric vehicles empowered by federated learning,'' *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6464–6473, Sep. 2022.
[51] G. Guo, J. Zhang, and N. Yorke-Smith, ''A novel recommendation model regularized with user trust and item ratings,'' *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1607–1620, Jul. 2016.
[52] L. Cui, D. Pi, and J. Zhang, ''DMFA-SR: Deeper membership and friendship awareness for social recommendation,'' *IEEE Access*, vol. 5, pp. 8904–8915, 2017, doi: 10.1109/ACCESS.2017.2704115.
[53] Z. Li, F. Xiong, X. Wang, Z. Guan, and H. Chen, ''Mining heterogeneous influence and indirect trust for recommendation,'' *IEEE Access*, vol. 8, pp. 21282–21290, 2020, doi: 10.1109/ACCESS.2020.2968102.

LIFENG HAN (Member, IEEE) received the M.S. degree in software engineering from Shandong University, in 2011. He is currently pursuing the Ph.D. degree with Northwest University. His main research interests include recommendation systems and natural language processing. He is a Senior Member of CCF.

LI CHEN received the Ph.D. degree in circuits and systems from Xidian University, in 2003. She is currently a Professor with Northwest University. She is the Chief Expert of the National Science and Technology Support Plan Project. She presides over and undertakes National Key Research and Development Projects, National Science and Technology Support Plan Projects, major instrument projects of the Ministry of Science and Technology and the National Natural Science Foundation, and other projects. She has published more than 150 academic articles, coauthored two monographs, and received more than ten patents. Her main research interests include intelligent information processing, data mining, and network security. She received the First Prize of Scientific and Technological Achievements among provincial colleges and universities.

XIAOLONG SHI received the B.S. degree in electronic science and technology from Northwest University, Xi'an, China, in 2019. He is currently pursuing the M.S. degree in computer science and technology with Xidian University, Xi'an. His research interests include recommender systems, natural language processing, and data visualization.