IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 December 2023, accepted 23 January 2024, date of publication 21 February 2024, date of current version 29 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3359751

RESEARCH ARTICLE

Fusion of Personalized Implicit Relations for Social Recommendation

WEI QIN, JIWEI QIN, AND TAO WANG
College of Computer Science and Technology, Xinjiang University, Urumqi, Xinjiang 830046, China
Key Laboratory of Signal Detection and Processing, Xinjiang University, Urumqi, Xinjiang 830046, China

Corresponding author: Jiwei Qin (Jwqin_xju@163.com)

This work was supported by the Science Fund for Outstanding Youth of Xinjiang Uygur Autonomous Region (No.2021D01E14).

ABSTRACT In recent years, research in social recommendation has shown that modeling implicit relations effectively alleviates data sparsity and cold-start problems. However, related research often overlooks the influence of user and item rating biases when constructing implicit relations. The divergence in user rating systems and item evaluation standards often results in ratings displaying starkly contrasting extremes, making it difficult to identify implicit relations among users (or items) to fulfill the requirements of personalized recommendations. To this end, we propose a fusion of personalized implicit relations for social recommendation (FIR-REC) based on graph neural networks. First, we regard the average ratings of users and items as benchmarks to eliminate the bias of user-item ratings. Based on this benchmark, user-user (item-item) pairs with the same preferences are regarded as implicit candidate pairs. Considering the heterogeneity of social relationships, we propose a preference hedging formula to calculate preference correlation coefficients for each implicit candidate pair and select the top-k ranked implicit friends for each node to construct a personalized implicit network. Specifically, this formula can utilize different scoring strategies to calculate the preference scores for consistency and inconsistency in implicit pairs. To enhance robustness, it introduces a balancing factor to mitigate the influence of less interacted implicit pairs. Next, we utilize a Graph Attention Network to aggregate neighbor node information in explicit and implicit social relations. Finally, we utilize a user-specific gating mechanism to integrate user representations from explicit and implicit social relations. This helps assess the importance of the two types of relations for different users, enabling precise and stable predictions. Extensive experiments on three open datasets demonstrate the superiority of our model compared with state-of-the-art social recommendation models.

INDEX TERMS Social recommendation, implicit social networks, recommender system, graph neural networks.

I. INTRODUCTION

In the era of information proliferation, the development and progress of recommendation systems have effectively mitigated the issue of information saturation. Traditional recommendation models that rely solely on user-item interaction data have not demonstrated ideal performance due to data sparsity and the cold start problem in practical applications. To address this issue, recommender systems that treat social relations as supplementary information have garnered widespread attention. As well supported by social influence theories and homophily, when users consider purchasing specific products, their friends within their social circle often play a significant role in their decision-making process, leading to similar preferences. With the widespread popularity of online social networks, communication between users has become more frequent and intimate, further amplifying social factors' impact. Consequently, researchers have started to explore methods to incorporate these social factors into recommendation models, opening valuable avenues to alleviate data sparsity and enhance the overall performance of recommendation systems.

To model the two relations in the social recommendation, namely user-user and user-item relations, existing research

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

always regards them as two graphs. They utilize Graph Neural Networks (GNNs) [1], [2] which mimic biological neural networks to aggregate neighbor information from both graphs and connect them to learn integrated representations. This approach effectively simulates social relations among users and interactions between users and items. For instance, in sRMGCNN [3], to obtain graph embeddings for users and items, Graph Neural Networks (GNNs) were employed. These embeddings were integrated with Recurrent Neural Networks (RNNs) to facilitate a diffusion process. GraphRec [4] utilizes Graph Neural Networks (GNNs) to capture and integrate information from bipartite graphs and social networks, effectively modeling graph data for social recommendation.

While there have been enhancements in the performance of these social recommendation models, we contend that the existing social recommendation models remain inadequate. The sparsity and unbalanced distribution of observable social connections have impeded the further development of social recommendations. Previous research [5] and [6] has already pointed out that the benefits of incorporating social data into recommendation systems are not ideal due to the limited quantity of social relations. To this end, some work [7], [8] mines implicit relations of users and items to enrich data. Implicit user relations are established between two users with similar preferences but unobserved social connections, while implicit item relations are indicated by items preferred by the same user. Implicit relations are better at reflecting user preferences compared to explicit relations. We can obtain more comprehensive data and make more accurate recommendations by combining these two types of relations.

In research related to the fusion of implicit relations [9], [10], we found that none of the research focused on addressing the bias issues in user and item ratings during the construction of implicit relations. Specifically, when using a 5-point scale, some users might think rating an item with a score of 3 is quite low, while others might consider a score of 1 indicative of poor quality. For different types of products, such as electronics and everyday goods, their rating distributions often exhibit distinct differences. Rating bias leads to biases in recommendation systems, making it more difficult to capture users' true interests and preferences. Users may only see items similar to what they have rated in the past while potentially missing out on other items that might align with their interests. This can lead to the recommendation system being unable to accurately capture user interests, resulting in a decrease in recommendation accuracy. Therefore, how to eliminate the bias of user and item ratings to construct more personalized implicit relations becomes an important issue that needs to be addressed urgently.

To address this, we propose a model for social recommendation based on graph neural networks called 'Fusion of Personalized Implicit Relations for Social Recommendation. To eliminate the bias of user and item ratings, we regard the average rating of each user and item as their preference benchmarks, enabling each individual to have a personalized preference standard. Considering the heterogeneity influenced by social relations, we categorize the preferences of implicit pairs into consistent and inconsistent preferences. We design a preference hedging formula to comprehensively measure the preference correlation of implicit pairs, constructing personalized implicit social networks and implicit item networks. This partially mitigates the problem of data sparsity. Finally, we employ a gating mechanism to assess the importance of these two social networks for different users, weakening unimportant features and strengthening important features, thereby improving recommendation performance.

In summary, the main contributions of this work are as follows:

(1) We propose a construction method for personalized implicit networks and design a new preference correlation formula that relieves data sparsity while eliminating user and item rating bias.
(2) We devise a user-specific fusion mechanism to construct the ultimate representation for each user within the context of both social networks. In this way, the importance of the two social networks for different users can be determined.
(3) We evaluate our model on a real-world dataset and demonstrate the effectiveness of our proposed model.

The rest of the paper consists of the following sections. The paper reviews related work on recommender systems and implicit relations modeling in Section II, Section III describes the basic concepts and detail of the algorithms of this paper. Experimental results on real datasets are showcased in Section IV, where a thorough analysis of the outcomes is provided. Section V summarizes the work of this paper and gives an outlook.

II. RELATED WORKS

A. GRAPH-BASED RECOMMENDATION

In recent years, the application of Graph Neural Network (GNN)-based methods has garnered significant attention due to their impressive performance in effectively modeling and learning from graph-structured data [11], [12], [13]. A prominent utility of Graph Neural Networks (GNNs) lies in their ability to capture a holistic and comprehensive representation of all nodes within a graph. This is often achieved by employing multiple graph propagation layers to aggregate influential information from higher-order neighbors. Hamilton's study [14] reduces the complexity of the recommendation system significantly by limiting the number of neighbors through sampling. Building upon this, PGE [15] introduced further enhancements to the sampling procedure, incorporating neighbor similarity as a weighted factor, thereby refining the information aggregation process. RGAT [16] delved into the realm of relational information-based attention mechanisms, encompassing both intra-relationship and cross-relationship graph attention. Zhu et al. proposed a recommendation model [17] applied to MOOCs, which utilizes the attention mechanism to extract information from heterogeneous graphs and

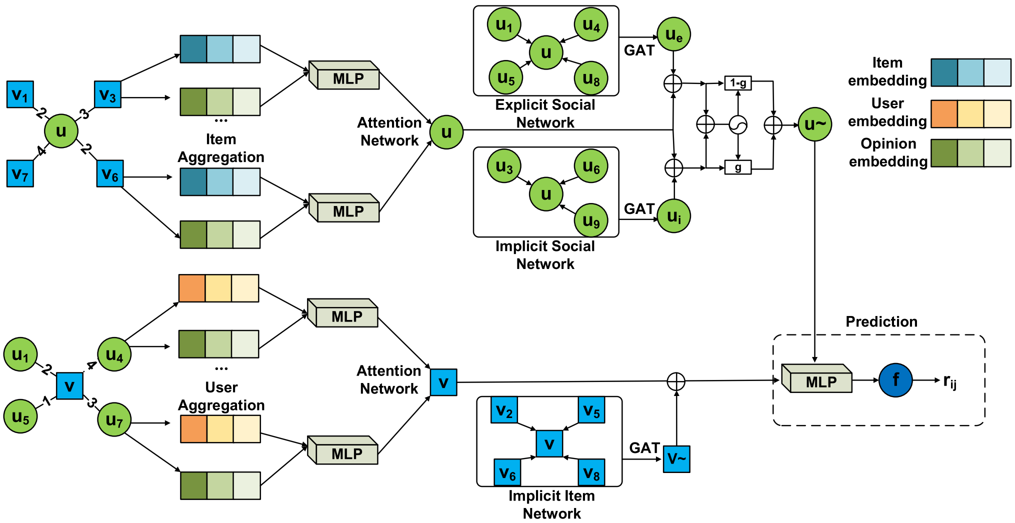

FIGURE 1. The overall architecture of the model. It consists of three main components: User modeling, item modeling, and score prediction.

knowledge graphs to eliminate noise and improve the robustness of the model.

B. SOCIAL RECOMMENDATION

Inspired by social theory studies such as homogeneity [18] and social influence [19], the choices made by users are frequently shaped by the preferences of their social friends, which has led to the proposal of a number of recommendation models that incorporate social relations.

In the early stages, research on social recommendation primarily revolved around discussing how to effectively leverage explicit social relations to enhance recommendation performance. Ma et al. [20] proposed a model that connects social information and rating data by sharing latent user features and conducting co-factorization on the rating matrix and relationship matrix. NSCR [21] employed users with social acquaintances as intermediaries to propagate user representations modelled through attribute-aware collaborative filtering. GraphRec [4] combines user-item interactions with the social graph to encompass interactions and opinions. This approach consolidates user and item data to capture user preferences.

Subsequent research findings [22] have indicated that employing explicit social relations yields less than ideal results, prompting a shift in focus towards implicit relations. CUNE [23] identifies trustworthy implicit friends through random walks and employs them to regularize matrix factorization prediction models. Wang and his colleagues [24] proposed leveraging strong and weak social relations between two nodes in the context of social recommendations. In [25], the model distinguishes between close friends and casual acquaintances by learning personalized similarity thresholds for different users. Furthermore, some studies have introduced trust metrics aimed at identifying trustworthy implicit connections by calculating and forecasting trust ratings derived from user interactions [26], [27]. Despite the improvements achieved by the aforementioned models, they tend to overlook the bias in user and item ratings.

We refer to a number of models based on scoring bias. In [28], The model combines individuals' trust bias with impersonal topological information to propose a classification method to solve the trust/distrust prediction problem. TrustTF [29] utilizes users' social trust information and implicit feedback to extend the bias tensor decomposition, effectively alleviating the data sparsity problem. Inspired by related papers, we propose a graph neural network-based approach that fuses implicit social networks for rating prediction while simultaneously addressing user and item rating bias, thereby enhancing the model's performance.

III. METHOD

We propose a comprehensive model structure, as depicted in Fig. 1, comprising three core modules: User Modeling, Item Modeling, and Rating Prediction. Our approach aims to overcome biases inherent in user and item ratings, constructing personalized implicit relation graphs for users and items. These graphs lay the foundation for capturing nuanced associations.

The User Modeling module encompasses three essential facets: the user-item interaction graph, the user-implicit relation graph, and the user-social graph. Each graph offers a distinct lens to comprehend user dynamics. At the foundational level, the Item Aggregation module deciphers user preferences by consolidating interactions alongside associated ratings. Subsequently, the Social Aggregation module, operating as the second layer, synergizes implicit user connections and explicit social ties. This approach enables a dual semantic perspective on user social information, enriching our understanding of their preferences. The resulting user embeddings from these two semantic viewpoints harmoniously merge to yield final latent factors.

Turning our attention to Item Modeling, we tackle two critical aspects: the item-user interaction graph and the item's implicit relation graph. To obtain features from these two different graphs, we use two different aggregations. First, user aggregation is implemented at the foundational level by analyzing user interactions to shape item representation. Then, the second layer enhances item representation by leveraging latent project relationships. Finally, model parameters are optimized within the score prediction module.

In the following sections, we provide detailed explanations for each component of the model. This holistic approach ensures a deep understanding of the proposed model architecture and functions.

A. PRELIMINARIES

In this section, to better understand the FIR-REC model, we first define the key concepts.

DEFINITION 1. User-user graphs (social networks). Represent a social network by a graph $G = (U - U, E_u)$, where $U = \{u_1, u_2, \ldots, u_n\}$ is the set of nodes (users), $U - U$ denotes a set of users and E_u denotes an edge between users. If $(u_i, u_j) \in E_u$ indicates that there is a relation between nodes u_i and u_j, (u_i, u_j) is also called a user pair. For a given node u_i, N_{ui} denotes the set of first-order neighbors of user u.

DEFINITION 2. User-item bipartite graph. Represent a user-item network by a graph $G = (U - V, E_v)$, here $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes (items), $U - V$ denotes a set of user-item pairs, and R_v denotes the ratings (edges) between users and items. If u_i rates v_j, r_{ij} is the rating score. Otherwise, we use 0 to denote the unknown rating from u_i to v_j, i.e., $r_{ij} = 0$.

DEFINITION 3. Adaptive Preference Benchmarks. To eliminate the influence of user and item rating biases, the average rating is used as a personalized benchmark to classify preferences.

$$\overline{\mathrm{r_u}} = \frac{\sum_{v_j \in N_u} r_{u,v_j}}{N_v} \tag{1}$$

$$\overline{r_v} = \frac{\sum_{u_i \in N_v} r_{v,u_i}}{N_u} \tag{2}$$

where N_u and N_v are the set of first-order neighbors of users u and v.

DEFINITION 4. Personalized preferences. According to the delineated adaptive preference cut-off. The preferred item of user $\{u : v|r_{u,v} \geq \overline{r}_u\}$. The non-preferred item of user $\{u : v|r_{u,v} < \overline{r}_u\}$. The preferred user of item $\{v : u|r_{v,u} \geq \overline{r}_v\}$. The non-preferred user of item $\{v : u|r_{v,u} < \overline{r}_v\}$.

DEFINITION 5. Implicit candidate pairs. If users u_i (item v_i) and u_j (item v_j) are not directly related in a social (item) network, but may have certain similar preferences (preferred), i.e., they have the same preference for interacting items (users), they are said to be implicit candidate pairs.

DEFINITION 6. Preference scores. According to the personalized preference calculation, implicit candidate friend pairs (item pairs) preference for the same set of items (users) are registered as N_p, the two preferences are not the same set of items (users) are registered as N_{up}, through the different preference scoring strategy to calculate the preference scores I_p of the two as well as non-preference scores I_{up}, in this paper, we directly use the implicit pairs of the two sets of the number of items (users) within the set (i.e., each interaction item (user) score value of 1) as the final scores.

DEFINITION 7. The preference correlation coefficient k. The preference scores I_p obtained according to Definition 6, as well as the non-preference scores I_{up} are passed through the preference hedging formula:

$$k = \frac{I_p - I_{up}}{N_p + N_{up} + \alpha} \tag{3}$$

the preference correlation coefficient $k \in [0, 1)$ of the two is obtained, where α is called the balancing factor, which can reduce the influence of low frequency interacting user pairs (item pairs) and improve the reliability of the implicit network.

B. BUILDING IMPLICIT RELATIONS

The user-item bipartite graph is a special kind of graph in which links between two disjoint nodes usually differ in type and have different semantics. Compared to modeling general graphs, bipartite graphs can provide additional information. First, we can directly model the main structure of a bipartite graph by observing the links between two nodes (user-item interactions in a user-item bipartite graph), which reveal the exchange of information flow between two distinct types of nodes, forming an interclass relationship.

In addition to considering modeling interclass relations between different sets of nodes, it is also important to model intraclass messages passed between nodes of the same type. Compared to ordinary homomorphic graphs, bipartite graphs exhibit a distinct characteristic: connections within a bipartite graph can exclusively occur between two separate sets of nodes, with no direct links between nodes of the same type. For example, in Fig. 2(a), connections between the U and V node sets create a bipartite graph, and nodes of the same type remain unconnected. Nevertheless, the bipartite graph embedding model must not solely focus on direct links between distinct node sets, as implicit associations among nodes of the same type also contain valuable semantic information.

Although there is no direct explicit link between nodes of the same type, there may be an implicit relation between nodes of the same type. If two nodes from the set of nodes U(V) can be connected through a node of the set of nodes V(U), then we consider these two nodes from U(V) to have higher connectivity. We can extend this straightforward observation into a broader theory: in a bipartite graph, when two nodes of the same type are connected by a path, it implies the existence of an implicit intra-class relationship between them.

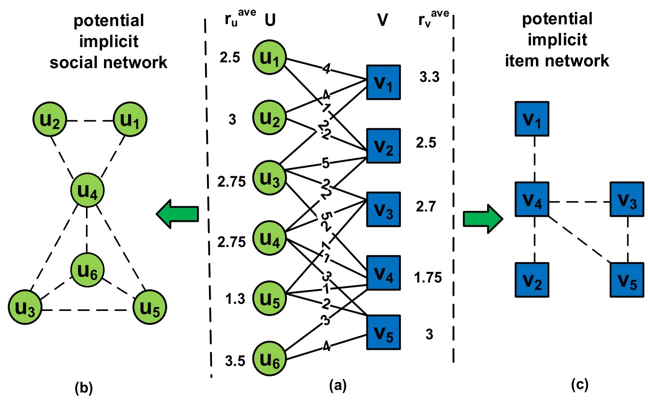

FIGURE 2. Building potential implicit networks.

As in Fig. 2 (b) and (c), in this paper, we take all the user (item) pairs with the same preference for interacting items (users) as implicit candidate pairs and filter each implicit candidate pair by defining 5 to the set of items (users) with the same preference denoted as N_p and the set of items (us- ers) with dissimilar preference denoted as N_{up}, and compute the preference scores I_p as well as the non-preference scores I_{up} of the two according to different preference scoring strategies. At this point, we find that there is a problem. For an implicit pair, when the count of items (users) with the same preference is 8, and the count of differing preference items (users) is 4, its similarity will be the same as that of an implicit pair when the count of items (users) with the same preference is 2, and the count of differing items (users) is 1. However, the former with the total interactions as 12 will have a higher degree of credibility of the similarity than that of the latter with the total interactions as 3 because, according to the statistical principle of the law of large numbers, the total number of the same interacting items The more, the closer to the true probability, i.e., the true similarity, so in Definition 7 we add a balancing factor α to reduce the weight of implicit pairs with fewer identical interactions, while α has little effect on the weight of implicit pairs with more identical interactions.

After obtaining all implicit candidate pairs and their corresponding preference correlation coefficients for each interacting item (user), to further determine similar users (items), we sort the preference correlation coefficients of all implicit candidate pairs for each user (item). We then select the top-k similar users (items) for each user (item) to construct the final implicit relation graph.

The pseudo-code of the implicit network generation algorithm proposed in this paper is shown in Table 1.

C. USER MODELING

The objective of user modeling is to uncover underlying user factors. For this purpose, we segment user modeling into three components: user-item interaction graph, user-implicit relationship graph and user-explicit relationship graph. The initial layer, referred to as item aggregation, is employed to acquire user latent factors within the item space through the user-item graph. Subsequently, the second layer, known as social aggregation, is utilized to capture latent representations of users within implicit and explicit social relationships. The two representations in the second layer are then adaptively fused to obtain the final user embedding.

TABLE 1. Implicit network generation algorithm based on personalized preferences.

Input: user-item interaction set, user-item rating set, item-user interaction set, item-user rating set.
Output: user-user (implicit friend) set, item-item (implicit item) set.
1. According to Definition 3, the adaptive preference baseline of the user (item) is calculated.
2. According to Definition 4, filtering (non)preferred items (users) for users (items) based on preference benchmarks.
3. According to Definition 5, construct the set of implicit candidate pairs.
4. According to Definition 6, compute the preference scores I_p as well as the non-preference scores I_{up} for implicit candidate pairs.
5. According to Definition 7, compute the preference correlation coefficient k for implicit candidate pairs.
6. Sort the relative preference coefficients of the implicit candidate pairs containing the users (items) and take the top-k as the final implicit friends (items) of u_i (v_j).
7. Returns the set of implicit friends (items).

1) ITEM AGGREGATION

This section is dedicated to extracting latent item characteristics from the user-item interaction graph to gain insights into user preferences. Within the user-item graph, one can find interactions between users and items and users' assessments and ratings of these items. By considering user-item interactions and ratings, we can aggregate this information to uncover latent user factors within the item space. Specifically, we use the following equation to aggregate this information to form initial user preferences.

$$h_i^S = \sigma(W \cdot f_{agg\_item}(\{x_{ia}, \forall a \in N(i)\}) + b) \tag{4}$$

where $N(i)$ is the set of items with which user u_i interacts, x_{ia} is a representation vector representing the opinion-aware interaction between u_i and item v_a, f_{agg_item} denotes the item aggregation function which is a nonlinear activation function, and W and b are the weights and biases of the neural network, respectively.

Opinions on items ($r \in \{1, 2, 3, 4, 5\}$) capture the user's preference for the items, so an opinion embedding. vector ($e_r \in R^d$) is introduced to represent each type of opinion r. We then employ MLP to process the item embedding q_a and e_r to obtain the opinion-aware interaction representation x_{ia}:

$$x_{ia} = f_{MLP}(q_a \oplus e_r) \tag{5}$$

where $\oplus$ represents the operation of concatenate vectors.

The aggregation strategy for f_{agg_item} is inspired by the attention mechanism [30], [31] to assign personalized weights to each user-item pair:

$$h_i^I = \sigma(W \cdot \{\sum\nolimits_{a \in N(i)} \alpha_{ia} x_{ia}\} + b) \tag{6}$$

where α_{ia} represents the opinion perception of the user u_i direct interaction items obtained by normalizing the attention score of x_{ia} for v_i embedded p_i using the SoftMax function:

$$\alpha_{ia} = \frac{\exp(w_2^T \cdot \sigma(W_1[x_{ia} \oplus p_i] + b) + b)}{\sum_{a \in N(i)} \exp(w_2^T \cdot \sigma(W_1[x_{ia} \oplus p_i] + b) + b)} \tag{7}$$

2) SOCIAL AGGREGATION

According to social correlation theory [32], users' preferences are influenced by their friends. For both the obtained implicit and explicit relations of users, we process them through GAT:

$$h_i^S = \sigma(W \cdot f_{agg\_social}(\{h_o^I, \forall o \in N(i)\}) + b) \tag{8}$$

$$h_i^S = \sigma(W \cdot \{\sum\nolimits_{o \in N(i)} \beta_{io} h_o^I\} + b) \tag{9}$$

$$\beta_{io} = \frac{\exp(w_2^T \cdot \sigma(W_1[h_o^I \oplus p_i] + b) + b)}{\sum_{o \in N(i)} \exp(w_2^T \cdot \sigma(W_1[h_o^I \oplus p_i] + b) + b)} \tag{10}$$

where β_{io} can be regarded as the degree of correlation between users.

3) LEARNING USER LATENT FACTORS

To enhance the acquisition of user latent factors, it's essential to collectively consider latent factors related to items and those linked to social connections because user-item graphs and social graphs offer distinct perspectives on user information. We use MLP to fuse the user's preference profile with each of the two social profiles:

$$h_i^{\exp} = \sigma(W_1[h_i^I \oplus h_i^{S_1}]) \tag{11}$$

$$h_i^{imp} = \sigma(W_1[h_i^I \oplus h_i^{S_2}]) \tag{12}$$

4) USER EMBEDDING FUSION

In the above section, we have obtained the embedded representations of each user under two different relations, corresponding to the user's explicit and implicit social relations. Now we need to design a strategy to integrate these two representations. For this purpose, we can employ a variety of common strategies such as concatenation and average pooling. Next, we will delve into the detailed implementation of these strategies individually.

a: CONCATENATION

To obtain a higher-level feature representation, we can employ a concatenation strategy to connect the explicit and implicit social relation embedding representations of users. Specifically, we concatenate these two embedding representations along the dimension direction and map them to a D-dimensional vector space using a learnable weight matrix. For each user, the strategy can be formulated as follows:

$$h_u = W_c[h_i^{\exp} \oplus h_i^{imp}] \tag{13}$$

b: AVERAGE POOLING

To combine the explicit and implicit social relation embedding representations of users, we can adopt an average pooling strategy. Specifically, we compute the average of these two embedding representations in each dimension and form a D-dimensional vector of these averages as the integrated embedding representation. For each user u, the strategy can be described as follows:

$$h_u = \frac{1}{2}[h_i^{\exp} + h_i^{imp}] \tag{14}$$

c: GATING MECHANISMS

Although the above two strategies are simple and easy to understand, we believe that they cannot generate satisfactory embedding representations for each user. Specifically, for each user, these two separate embedding representations should play different contributions to generate an information-rich user representation. Therefore, drawing on the idea of LSTM [33] modeling, we employ a user-specific gating mechanism, serving as an adaptive fusion strategy, to assess the significance of the two social relations in the ultimate representation of a specific user. The computational form of this gating mechanism is as follows:

$$g_u = \sigma(W_1 h_i^{\exp} + W_2 h_i^{imp}) \tag{15}$$

$$h_u = g_u \odot h_i^{\exp} + (1 - g_u) \odot h_i^{imp} \tag{16}$$

where $\odot$ denotes the elementwise product operation between two vectors, σ denotes the sigmoid function, W_1 and W_2 denote the learnable weight matrix. Through the gating mechanism, the user's embedding based on implicit and explicit relations is fused into the final user representation.

D. ITEM MODELING

Item modeling is used to learn the underlying factors of item v_j. Items are not only associated with users in the user-item graph, but also with items in the item-item implicit graph, and to intrinsically combine these two graphs, we model the item-user graph and the item-item graph through GAT in the same way as for user modeling, respectively.

1) USER AGGREGATION

The aggregation of user information involves consolidating data from all users associated with a project in the userproject graph. The user's opinions on the item is pivotal in acquiring item characteristics. Therefore, we aggregate users' score to represent item embeddings:

$$h_j^S = \sigma(W \cdot f_{agg\_user}(\{g_{jt}, \forall t \in N(j)\}) + b) \tag{17}$$

$$g_{jt} = f_{MLP}(p_t \oplus e_r) \tag{18}$$

$$h_j^S = \sigma(W \cdot \{\sum\nolimits_{t \in N(j)} \mu_{jt} g_{jt}\} + b) \tag{19}$$

$$\beta_{io} = \frac{\exp(w_2^T \cdot \sigma(W_1[g_{jt} \oplus q_j] + b) + b)}{\sum_{t \in N(j)} \exp(w_2^T \cdot \sigma(W_1[g_{jt} \oplus q_j] + b) + b)} \tag{20}$$

user attention μ_{jt} is employed to capture heterogeneous influences from user-item interactions to learn the underlying factors of user-space items.

2) IMPLICIT ITEM AGGREGATION

We use GAT to model the implicit item relations. The implicit relations between items allow the recommender systems to gain a more comprehensive understanding of users who share preferences for similar items and make it easier to extract deeper connections from the network:

$$h_j^V = \sigma(W \cdot f_{item\_item}(\{k_{jt}, \forall t \in N(j)\}) + b) \tag{21}$$

$$h_j^V = \sigma(W \cdot \{\sum\nolimits_{t \in N(j)} k_{jt} I_{jt}\} + b) \tag{22}$$

$$k_{jt} = \frac{\exp(w_2^T \cdot \sigma(W_1[I_{jt} \oplus q_j] + b) + b)}{\sum_{t \in N(j)} \exp(w_2^T \cdot \sigma(W_1[I_{jt} \oplus q_j] + b) + b)} \tag{23}$$

where f_{item_item} is the aggregation function of the implicit item graph G_{item}. Additionally, we incorporate an attention mechanism to discern the significant weight of relevant items using a two-layer neural attention network.

3) LEARNING ITEM LATENT FACTOR

Similar to user modeling, we fused two different representations of the item with MLP to obtain a final latent representation:

$$h_v = \sigma(W[h_j^S \oplus h_j^V]) \tag{24}$$

E. RATING PREDICTION

With the latent factors of users and items, we combine them and input them into the MLP for the purpose of rating prediction.

$$r_{uv} = f_{MLP}(h_u \oplus h_v) \tag{25}$$

F. MODEL TRAINING

To accomplish the task of rating prediction, we employ a widely utilized loss function to optimize the model parameters in FIR-REC:

$$Loss = \frac{1}{2|O|} \sum\nolimits_{i,j \in O} (r'_{ij} - r_{ij})^2 \tag{26}$$

$|O|$ is the number of observed ratings, r_{ij} is the factual rating of item v_j by user u_i.

We use the RMSprop optimizer [34] to optimize the loss, learn the randomly initialized item embeddings q_j, user embeddings p_i, and opinion embeddings e_r in the model, and then predict the ratings. To prevent model overfitting, we use a dropout strategy.

IV. EXPERIMENT

A. EXPERIMENTAL SETTINGS

1) DATASETS

We evaluate the effectiveness of our framework using four publicly representative datasets and compare the outcomes to a state-of-the-art baseline followed by parameter sensitivity experiments.

TABLE 2. Statistics of the datasets.

Dataset	Ciao	Epinions	Yelp	Flixster
Users	7317	18,088	21,461	58470
Items	10,4975	261,649	102,433	38076
Ratings	283,319	764,352	894,435	3,619,736
Ratings Density	0.0368%	0.0161%	0.0407%	0.1625%
Social Relations	111,781	355,813	497,206	667,313
Link Density	0.2087%	0.1087%	0.1079%	0.0195%

TABLE 3. Experimental environment configuration.

Name	Configuration Information
Operating System	Windows 10
Development Language	Python 3.8.17
Framework	Pytorch 1.13.0 + cuda 11.7
CPU	AMD R7-5800H
GPU	RTX 3060
RAM	500G

(1) Ciao and Epinions: These two datasets are from popular social networking sites (http://www.ciao.co.uk) and (http://www.Epinions.com). Users can socially networking platforms to rate, comment and make friends with various items. As a result, these platforms provide rich data on ratings (ratings at [1] and [5]) as well as social relations.

(2) Yelp and Flixster: Yelp (https://www.yelp.com) is a popular online review platform where users can make friends, review and give ratings in the [1] and [5] range. The Flixster (https://www.flixster.com) dataset also contains rating and social information. The rating values of the Flixster dataset are 10 discrete numbers in the range [0.5 5], with a step size of 0.5.

We tested our FIR-REC model on four representative datasets Ciao, Epinions, Yelp and Flixster, details of which can be seen in Table 2.

The experimental environments and hardware-related configurations in which the models are trained in this paper are shown in Table 3.

2) BASELINES

For performance evaluation, we conducted a comparative analysis of our FIR-REC model against three categories of methods: traditional recommender systems, traditional social recommender systems, and deep neural network-based recommender systems. Within each category, we select representative baselines, and these are outlined below:

- PMF [36]: Probabilistic matrix factorization via Gaussian distribution using user-item rating matrix to model user and item latent factors.
- SoRec [20]: Probabilistic matrix decomposition of user-item rating matrix and user-user social relations matrix for social recommendation.

TABLE 4. Comparison of the performance of all methods on the three datasets for the two metrics MAE and RMSE. The best baseline is underlined and the best performance among all methods is in bold.

Datasets		Ciao		Epinions		Yelp		Flixster	
Metrics		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Algorithms	PMF	1.1238	0.9021	1.2128	0.9952	1.1282	0.7559	1.0042	0.7559
	SoRec	1.0652	0.8410	1.1437	0.8961	1.0736	0.7303	0.9693	0.7303
	SoReg	1.0848	0.8611	1.1703	0.9119	1.0621	0.7415	0.9727	0.7415
	SociaiMF	1.0501	0.8270	1.1328	0.8837	1.0368	0.8137	0.9705	0.7378
	TrustMF	1.0479	0.7690	1.1395	0.8410	1.0680	0.8226	0.9380	0.7091
	NeuMF	1.0617	0.8062	1.1476	0.9072	1.0281	0.7946	0.9698	0.7420
	DeepSoR	1.0316	0.7739	1.0972	0.8383	1.0435	0.8132	0.9548	0.7190
	GCMC+SN	0.9931	0.7526	1.0711	0.8590	1.0296	0.7993	0.9613	0.7351
	GraphRec	0.9824	0.7416	1.0647	0.8187	1.0254	0.7962	0.9481	0.7182
	SMIN	<u>0.9671</u>	0.7268	**1.0448**	**0.7875**	**1.0182**	**0.7823**	<u>0.9135</u>	**0.6836**
	GL-HGNN	0.9679	<u>0.7252</u>	<u>1.0474</u>	<u>0.7957</u>	<u>1.0207</u>	<u>0.7858</u>	0.9229	0.7041
	GDSRec	0.9740	0.7323	1.0566	0.8047	1.0228	0.7914	0.9285	0.6988
	GraphRec+	0.9815	0.7349	1.0592	0.8112	1.0235	0.7934	0.9326	0.7049
	FIR-REC	**0.9658**	**0.7234**	1.0512	0.8020	1.0216	0.7875	**0.9126**	<u>0.6842</u>
Improvement (%)		0.14%	0.25%	-0.60%	-1.80%	-0.33%	-0.66%	0.10%	-0.09%

- SoReg [37]: Constraining matrix decomposition by modeling social information through regularization terms.
- SocialMF [38]: Applies preferences as well as preference propagation to the matrix decomposition model to make a user's behavior closer to the average preferences of his neighbors.
- TrustMF [39]: The method employs matrix decomposition techniques to decompose rating and trust data by sharing user latent vectors, mapping users into two low-dimensional spaces: the trustor space and the trusted space.
- NeuMF [40]: It extracts low-dimensional and high-dimensional latent features through matrix decomposition and MLP.
- DeepSoR [41]: This model employs deep neural networks to extract potential features of users from social relations and then predicts ratings through probabilistic matrix decomposition.
- GCMC+SN [42]: This model processes bipartite graphs with graph convolution, which fully uses node information and topology for recommendation and leads to state-of-the-art results.
- GraphRec [4]: This model incorporates attention mechanisms into its framework to capture the interplay of opinions within the user-item graph. Simultaneously, it aggregates neighborhood information from the user-item bipartite graph and social information from the user-user social network.
- SMIN [29]: This model utilizes a heterogeneous graph neural network guided by meta-paths to learn the social and knowledge dependencies between users and items. It introduces a self-supervised learning framework to enhance the modeling of graph structure information.
- GL-HGNN [43]: This model captures high-level semantic relationships and topological information through a heterogeneous global graph. It utilizes graph regularization to design graph learners for reducing computational complexity.
- GDSRec [44]: This model has dealt with the original graph by transforming it into a decentralized graph using statistical information. It treats the rating biases of users and items as vectors, providing a decentralized perspective for learning latent factor offsets of users and items.
- GraphRec+ [9]: The model builds implicit item-item graphs by cosine similarity and adds them to GraphRec to better learn user and project representations.

3) EVALUATION METRICS

To assess the prediction accuracy of recommendation algorithms, two widely employed metrics, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), are frequently utilized. Smaller MAE and RMSE values correspond to higher prediction accuracy. It is worth emphasizing that even minor enhancements in both metrics can substantially enhance the quality of top-N recommendations [34].

4) PARAMETER SETTINGS

For each dataset, we use 80% of it as a training set for learning the parameters, 10% as a validation set for tuning the hyperparameters, and 10% as a test set. The embedding size was set to {8, 16, 32, 64, 128, 256}, the learning rate was set to {0.005, 0.001, 0.005, 0.01, 0.05, 0.1}, and in order to test the validity of the balancing factor α, we set α to

$\{0, 1, 2, 3, 4, 5\}$, and in order to build an implicit network of users and items, The k-values of the most similar users/items were tested in $\{3, 5, 8, 10, 15, 20, 30, 50, 70\}$.

B. EXPERIMENTAL RESULTS AND ANALYSIS

We compare the recommendation performance of all methods on the four datasets in Table 4. We have the following main findings:

- In terms of both metrics, we can observe that FIR-REC performs better on the Ciao and Flixster datasets compared to other models. Compared with the latest model performance, it shows comparable performance on the Yelp dataset, while exhibiting a slight decrease in performance on the Epinions dataset. This suggests the effectiveness of our approach to constructing implicit social networks, particularly for densely rated datasets such as Ciao and Flixster, where rich statistical information enhances the reliability of the implicit social network we have constructed.
- Among various baselines, the performance of Graph Neural Network (GNN) social recommendation models such as GraphRec and GDSRec surpasses that of models not utilizing GNN. This suggests that the graph structure embedding aggregation paradigm is an effective solution for social-aware recommendation systems, validating the effectiveness of GNN in social recommendations. Additionally, SMIN and GL-HGNN achieve better performance when constructing implicit project graphs, indicating that adding extra implicit relationship connections in the user-item graph may contribute to social recommendations. Our model not only constructs implicit project relationships but also builds implicit social relationships, achieving further performance improvement through attention mechanisms for information aggregation.
- We can observe that recently proposed models employing meta-path-guided heterogeneous networks (i.e., SMIN and GL-HGNN) exhibit significantly improved performance compared to models utilizing homogeneous networks (i.e., GraphRec and GraphRec+). This highlights the superiority of heterogeneous networks in capturing high-order complex semantic relationships. Despite employing a homogeneous graph, our FIR-REC achieves performance that matches or even surpass+ses models using heterogeneous networks, demonstrating the superiority of our implicit social network construction and fusion approach.
- While GDSRec also took into account the impact of user and project rating biases. Through the comparison between FIR-REC and GDSRec, we can observe that removing rating biases and adopting feature-adaptive selection methods during the construction of implicit social networks is more effective in enhancing performance.

TABLE 5. The impact of fusion strategy of FIR-REC.

Dataset	Metrics	AF	CC	AR
Ciao	MAE	0.7234	0.7315	0.7294
	RMSE	0.9658	0.9734	0.9708
Epinions	MAE	0.8020	0.8116	0.8137
	RMSE	1.0512	1.0557	1.0542

TABLE 6. The impact of different part of FIR-REC.

Dataset	Metrics	FIR-REC	Without IS	Without IT	Without AF
Ciao	MAE	0.7234	0.7303	0.7324	0.7315
	RMSE	0.9658	0.9709	0.9678	0.9734
Epinions	MAE	0.8020	0.8084	0.8068	0.8113
	RMSE	1.0512	1.0545	1.0551	1.0582

C. EFFECTS OF MODEL COMPONENTS AND MODEL HYPERPARAMETERS

1) EFFECTS OF INDIVIDUAL MODEL COMPONENTS

To better understand the effect of the different parts of the model, we compared our model with three variants: i.e., without Implicit Social relations (IS), without Implicit Item relations (IT) and without Adaptive Fusion (AF), and the results are shown in Table 6. We can observe that the worst performance without the adaptive module is because feature splicing does not distinguish well the importance of the two networks for a specific user. The performance of the other two variants is also significantly lower with respect to the original model since modeling the implicit relations between users and items reveals different semantics than the explicit one and can indirectly represent the potential relationship between users/items from the item/user point of view, thus improving the recommendation accuracy. It can also be seen that FIR-REC, which only employs implicit item relations, has a substantial performance improvement relative to GraphRec+, which also uses implicit item relations, reflecting the superiority of our implicit network construction strategy.

2) IMPACT OF FUSION STRATEGIES

To investigate the impact of our designed user-specific gating mechanism on recommendation performance, we employed alternative fusion strategies, namely Concatenation (CC) and Average Pooling (AR), for comparison. We compared the performance of different fusion strategies on the Ciao and Epinions datasets, as shown in Table 5. As anticipated, our FIR-REC model with the user-specific gating mechanism (AF) outperforms other strategies in terms of performance. This result further validates the rationality and effectiveness of the fusion strategy we devised.

The Concatenation strategy exhibited the poorest performance, possibly due to the insufficiency of the weight matrix

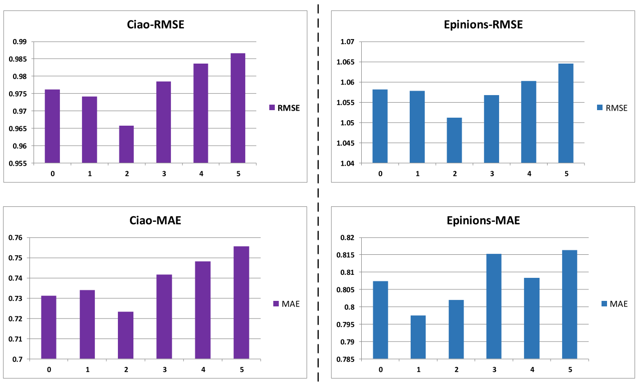

FIGURE 3. Effect of α on ciao and epinions datasets.

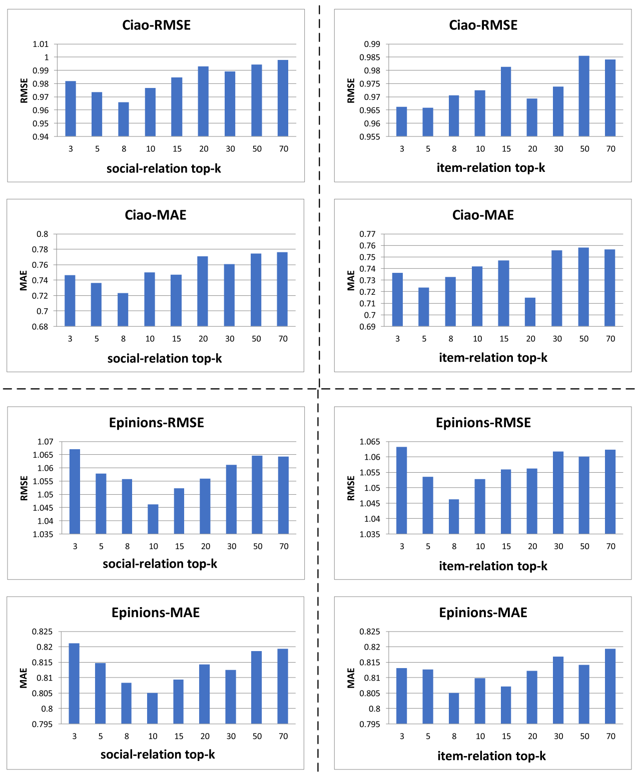

FIGURE 4. Effect of top-k related users/items on ciao and epinions. datasets.

to extract each user feature from the two spaces. While Average Pooling showed a slight improvement over Concatenation, it still fell short of the gating mechanism's performance. This could be attributed to the fact that averaging the two features does not adequately capture the significance of each network for the users.

3) SENSITIVITY ANALYSIS OF HYPERPARAMETER α

In this section, to test the validity of the balancing factor α we consider different values of $\alpha \in (0, 1, 2, 3, 4, 5)$ on the Ciao and Epinions datasets to compare model performance. As shown in Fig. 3, when $\alpha = 2$, FIR-REC demonstrates better

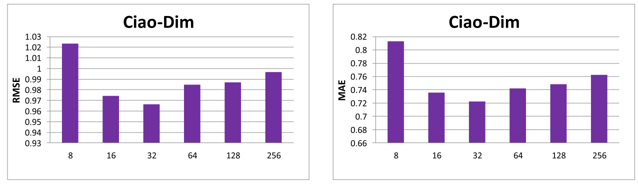

FIGURE 5. Effaect of embedding size on ciao dataset.

performance on both datasets. As α increases, it impacts the implicit candidate pairs with many of the same interactions too heavily, resulting in a decrease in performance. Conversely, when α decreases, the effect on the implicit candidate pairs with a small number of the same interactions becomes too marginal to yield a significant performance increase.

4) THE EFFECT OF TOP-K RELATED USERS/ITEMS

We empirically select the top-k similar users/items for each item to construct the implicit user and implicit item graphs. In this subsection, we investigate the impact of the value of k on the proposed model's performance. Fig. 4 illustrates the performance of FIR-REC across different values of k. The optimal performance on the Ciao dataset is achieved with an implicit social relations parameter of $k = 8$ and an implicit item relation parameter of $k = 5$. On the Epinions dataset, the best performance is observed with $k = 8$ for the implicit social relation and $k = 15$ for the implicit item relation. Initially, the performance tends to improve as the value of k increases, indicating that incorporating the most-similar users/items are beneficial. However, when the value of k is too large, performance tends to degrade noticeably, as it introduces excessive noise in implicit social and project relationships.

5) THE EFFECT OF EMBEDDING SIZE

We assess the impact of embedding size on model performance using the Ciao dataset. As illustrated in Fig. 5, we observe a trend there is a notable performance enhancement as the embedding size increases from 8 to 32. However, an excessively large embedding size can also increase the complexity of our model, leading to a decrease in performance. Thus, it becomes imperative to identify an optimal embedding length that strikes a balance between model performance and complexity.

V. CONCLUSION

To construct a personalized implicit network by eliminating the bias of user and item ratings, we propose an FIR-REC social recommendation model based on the GNN. The exemplary performance of this framework can be attributed to two key factors: Firstly, the devised method for constructing personalized implicit networks facilitates precise learning of user and item representations. Secondly, the user-specific gating mechanism discerns the significance of explicit and implicit social networks for distinct users, con- sequently facilitating a more precise fusion of embeddings across both networks. Notably, our approach to constructing implicit relations exhibits generalizability, rendering its applicability to other models noteworthy. Importantly, empirical results across four real-world datasets demonstrate the substantial enhancement of social recommendation performance achieved by our proposed method.

Expanding upon our future research endeavors, we will delve into a more exhaustive examination of the impacts of various preference scoring strategies on model performance. Our focus will particularly revolve around the precise computation of preference scores, and we intend to shed light on this aspect through a series of rigorous experiments. Furthermore, it is imperative to recognize that both implicit and explicit relational networks frequently carry a considerable amount of noise within them. While the attention mechanism has proven to be effective in mitigating this noise, it still faces limitations in entirely eradicating it. Consequently, a pivotal facet of our forthcoming research will involve the identification and subsequent removal of these noise nodes that can potentially be more detrimental than beneficial to the network. In essence, our mission is to enhance the robustness and efficiency of our models in the face of noisy data and evolving preferences.

ACKNOWLEDGMENT

The author expresses gratitude to the anonymous reviewers for dedicating valuable time to provide constructive feedback and suggestions. Their insights have elevated the quality of the work. They would also acknowledge Prof. Qin Jiwei for his guidance and encouragement.

REFERENCES

[1] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2008.

[2] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Mar. 2020.

[3] F. Monti, M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3700–3710.

[4] W. Fan, "Graph neural networks for social recommendation," in *Proc. World Wide Web Conf.*, New York, NY, USA, May 2019, pp. 417–426.

[5] J. Tang, X. Hu, and H. Liu, "Social recommendation: A review," *Social Netw, Anal. Mining*, vol. 3, pp. 1113–1133, Jan. 2013.

[6] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2011, pp. 1082–1090.

[7] J. Yu, M. Gao, J. Li, H. Yin, and H. Liu, "Adaptive implicit friends identification over heterogeneous network for social recommendation," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 357–366.

[8] H. Liu, L. Jing, J. Yu, and M. K. Ng, "Social recommendation with learning personal and social latent factors," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 7, pp. 2956–2970, Jul. 2021.

[9] W. Fan, Y. Ma, Q. Li, J. Wang, G. Cai, J. Tang, and D. Yin, "A graph neural network framework for social recommendations," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2033–2047, May 2022.

[10] A. Salamat, X. Luo, and A. Jafari, "HeteroGraphRec: A heterogeneous graph-based neural networks for social recommendations," *Knowl.-Based Syst.*, vol. 217, Apr. 2021, Art. no. 106817.

[11] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–14.

[12] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 974–983.

[13] Y. Zhu, F. Cong, D. Zhang, W. Gong, Q. Lin, W. Feng, Y. Dong, and J. Tang, "WinGNN: Dynamic graph neural networks with random gradient aggregation window," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2023, pp. 3650–3662.

[14] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.

[15] Y. Hou, H. Chen, C. Li, J. Cheng, and M.-C. Yang, "A representation learning framework for property graphs," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 65–73.

[16] B. Dan, S. Dane, C. Pietro, and H. Nils, "Relational graph attention networks," in *Proc. ICLR*, 2019.

[17] Y. Zhu, Q. Lin, H. Lu, K. Shi, D. Liu, J. Chambua, S. Wan, and Z. Niu, "Recommending learning objects through attentive heterogeneous graph convolution and operation-aware neural network," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 4178–4189, Apr. 2023.

[18] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annu. Rev. Sociol.*, vol. 27, no. 1, pp. 415–444, Aug. 2001.

[19] P. V. Marsden and N. E. Friedkin, "Network studies of social influence," *Sociolog. Methods Res.*, vol. 22, no. 1, pp. 127–151, 1993.

[20] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proc. 17th ACM Conf. Inf. Knowl. Manage.*, Oct. 2008, pp. 931–940.

[21] X. Wang, X. He, L. Nie, and T.-S. Chua, "Item silk road: Recommending items from information domains to social users," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 185–194.

[22] H. Park, H. Jeon, and J. Kim, "UniWalk: Explainable and accurate recommendation for rating and network data," 2017, *arXiv:1710.07134*.

[23] C. Zhang, L. Yu, Y. Wang, C. Shah, and X. Zhang, "Collaborative user network embedding for social recommender systems," in *Proc. SIAM Int. Conf. Data Mining (SDM)*, 2017, pp. 381–389.

[24] W. Xin, L. Wei, E. Martin, W. Can, and C. Chen, "Social recommendation with strong and weak ties," in *Proc. 25th ACM Int. ACM*, 2016, vol. 5, no. 14, pp. 5–14.

[25] X. Wang, S. C. H. Hoi, M. Ester, J. Bu, and C. Chen, "Learning personalized preference of strong and weak ties for social recommendation," in *Proc. 26th Int. Conf. World Wide Web*, Apr. 2017, pp. 1601–1610.

[26] S. Fazeli, B. Loni, A. Bellogin, H. Drachsler, and P. Sloep, "Implicit vs. explicit trust in social matrix factorization," in *Proc. 8th ACM Conf. Recommender Syst.*, Oct. 2014, pp. 317–320.

[27] S. M. Taheri, H. Mahyar, M. Firouzi, E. Ghalebi K., R. Grosu, and A. Movaghar, "Extracting implicit social relation for social recommendation techniques in user rating prediction," in *Proc. 26th Int. Conf. World Wide Web Companion WWW Companion*, 2017, pp. 1343–1351.

[28] J. Zhao, W. Wang, Z. Zhang, Q. Sun, H. Huo, L. Qu, and S. Zheng, "TrustTF: A tensor factorization model using user trust and implicit feedback for context-aware recommender systems," *Knowl.-Based Syst.*, vol. 209, Dec. 2020, Art. no. 106434.

[29] X. Long, C. Huang, Y. Xu, H. Xu, P. Dai, L. Xia, and L. Bo, "Social recommendation with self-supervised metagraph informax network," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2021, pp. 1160–1169.

[30] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proc. World Wide Web Conf. World Wide Web*, 2018, pp. 1583–1592.

[31] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1480–1489.

[32] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 1994.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[34] T. Tieleman and G. Hinton, *Coursera: Neural Networks for Machine Learning-Lecture 6.5: RMSprop*. Toronto, ON, Canada: Univ. Toronto, 2012.

[35] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2008, pp. 426–434.
[36] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, 20.
[37] 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*, Feb. 2011, pp. 287–296.
[38] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst.*, New York, NY, USA, Sep. 2010, pp. 135–142.
[39] 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.
[40] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web (WWW)*, 2017, pp. 173–182.
[41] W. Fan, Q. Li, and M. Cheng, "Deep modeling of social relations for recommendation," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1.
[42] R. van den Berg, T.N. Kipf, and M. Welling, "Graph convolutional matrix completion," 2017, *arXiv:1706.02263*.
[43] Y. Zhang, L. Wu, Q. Shen, Y. Pang, Z. Wei, F. Xu, E. Chang, and B. Long, "Graph learning augmented heterogeneous graph neural network for social recommendation," *ACM Trans. Recommender Syst.*, vol. 1, no. 4, pp. 1–22, Dec. 2023.
[44] J. Chen, X. Xin, X. Liang, X. He, and J. Liu, "GDSRec: Graph-based decentralized collaborative filtering for social recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 4813–4824, May 2023.

WEI QIN received the bachelor's degree in mechanical design, manufacturing and automation from the Shandong University of Science and Technology, in 2020. He is currently pursuing the master's with the College of Computer Science and Technology, Xinjiang University. His research interests include deep learning and recommendation algorithms.

JIWEI QIN received the Ph.D. degree from the School of Telecommunications, Xi'an Jiaotong University, in 2013. She is currently a Professor with the College of Computer Science and Technology, Xinjiang University. Her primary research interests include big data analysis, emotion computing, and intelligent network learning.

TAO WANG is currently pursuing the master's degree with the School of Information Science and Engineering, Xinjiang University. His main research interests include deep learning and recommendation algorithms.