Journal of Intelligent Information Systems (2024) 62:57–75
https://doi.org/10.1007/s10844-023-00802-3

RESEARCH

Deep latent representation enhancement method for social recommendation

Xiaoyu Hou[1] · Guobing Zou[1] · Bofeng Zhang[2] · Sen Niu[2]

Received: 14 February 2023 / Revised: 7 June 2023 / Accepted: 7 June 2023 /
Published online: 29 July 2023

Abstract
Social recommendation can effectively improve recommendation performance by leveraging social relationships to alleviate the sparsity of user-item interaction data. Because these connections in social recommendation can be easily represented as graph-structured data, social recommendation based on graph neural networks has received significant attention. However, existing works focus on modeling the long-term preferences of users and rarely consider the effect of temporal factors on preferences, resulting in a failure to accurately learn the representation of present preferences. Moreover, existing works mainly utilize similarity to connect different items. But items in the same category often have more connections and correlations with each other, which can be employed to enhance the learning of item representations. Therefore, this work proposes DLREM (Deep Latent Representation Enhancement Method for Social Recommendation) to address the above limitations. Specifically, DLREM exploits dual graph attention networks to learn long-term representations of users and items separately and exploits recurrent neural networks to capture the dynamic preferences of users. In addition, attention mechanisms are used to model user social relationships and item correlations, enhancing the learning of user and item representations. Combining the enhanced deep latent representations of users and items can improve the accuracy of social recommendation. Experimental results on two public datasets show that our model achieves competitive performance compared with state-of-the-art models.

Keywords Social recommendation · Graph neural networks · Social network · Representation learning

1 Introduction

Recommender systems are being used on a variety of web platforms due to their effectiveness in reducing information overload now. For a long time, the core idea of recommender systems has been collaborative filtering (Koren et al., 2022, 2009), but it is frequently hampered by the problem of data sparsity. With the growth of social media, social recommendation, which

✉ Guobing Zou
gbzou@shu.edu.cn

✉ Bofeng Zhang
bfzhang@sspu.edu.cn

Extended author information available on the last page of the article

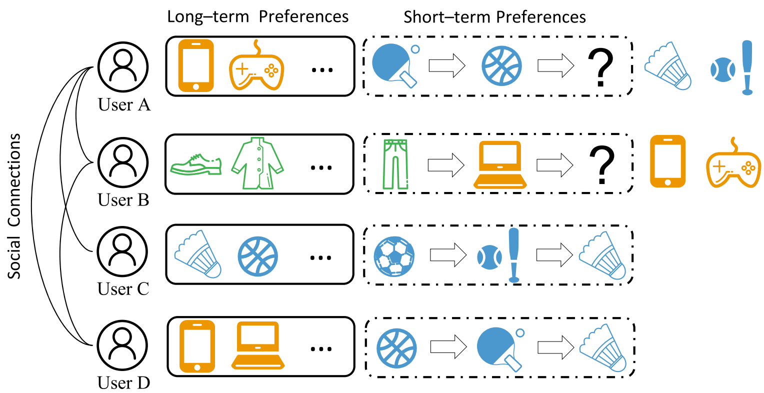


Fig. 1 An illustration of how long-term and short-term preferences of users are influenced by their friends

exploits social neighbors to capture the preferences of a sparsely interacting user, appears to be a promising solution to this problem. Social recommendation methods have been widely applied to different recommendation tasks with impressive performance(e.g., product (Du et al., 2022; Li et al., 2022a; Liu et al., 2022), location (Chen and Wong, 2021; Yu et al., 2020; Li et al., 2022b)). According to relevant sociological theories (McPherson et al., 2001; Marsden and Friedkin, 1993), social recommendation assumes that the preferences of a user may be similar to or influenced by those around him (or those who are connected).

Traditional social recommendation methods (Jamali and Ester, 2010; Li et al., 2017; Ma et al., 2008, 2011; Yang et al., 2016) are primarily based on a matrix factorization framework, in which social relationships are frequently represented as regular terms or as integration terms to affect the matrix factorization framework. In recent years, researchers have developed a number of social recommendation methods based on neural network techniques (Chen et al., 2019a, b), which introduce attention mechanisms into the modeling process. Graph neural networks(GNN) have recently demonstrated strong representational learning capabilities on graph-structured data. Since data in social recommendation can be well represented as graph data, researchers (Gao et al., 2022; Wu et al., 2022b) have focused on social recommendations based on GNN.

However, existing GNN-based social recommendation methods mainly use historical user-item interaction data to model long-term user preferences. In fact, the preferences of users can change dynamically over time. For example, a user may develop an interest in new sports to which he has never been exposed, and these short-term preferences are also influenced by his friends. Figure 1 is an illuminating example where the preferences of users are represented as long-term and short-term preferences. The figure shows that User A and his friends C and D have been following sports lately. User A may be influenced by his friends C and D to follow sports such as badminton or baseball next based on this fact. In addition, both user A and D have a long-term preference for electronics. User B may be influenced by the long-term preferences of his friends A and D to focus more on electronics such as mobile phones or game consoles next. Thus, the dynamic influence of friends on users can contribute to learning the current preferences of the target users. Furthermore, item attractiveness is usually related to attribute information such as item category and brand. The attractiveness provided by these attribute information is usually consistent over time. For example, a loyal iPhone user may be more likely to purchase Apple headphones when shopping for electronic products. Thus, there are correlated relationships between different items in the same category. Learning about the representation of item attractiveness can be improved through rational modeling of the correlations between items

This paper proposes a novel social recommendation method to address the aforementioned issues. Specifically, our model is divided into three main parts: user modeling, item modeling, and rating prediction. In user modeling, users are involved in two different views, the user-item graph and the user-user graph. Two graph attention networks(GAT) modules are introduced to learn from each of the two graph views, which aims to obtain long-term preferences and preferences based on the influence of social relationships. Furthermore, temporal contextual information is introduced in the user-item graph, and short-term trends in user preferences are captured by using recurrent neural networks. Constructing correlation graphs about the items by item categories is the first step in item modeling. Similarly, two GAT modules are introduced to learn from each of the two graph views in item modeling. In rating prediction, the possible rating of an item by a given user can be obtained by combining the user and item latent factor representations learned by the aforementioned two components. Our main contributions are as follows:

- This paper proposes DLREM that incorporates temporal contextual information into the learning of user preference representations, which can enhance deep latent representations of users.
- Unlike the existing works on social recommendation, DLREM constructs correlation graphs of items from category information and obtains more accurate deep latent representations of items by modeling the correlations.
- DLREM can exploit the enhanced deep latent representations of users and items to accurately capture the preferences of users for items, thereby improving the performance of recommendation.

2 Related work

2.1 Traditional social recommendation methods

As one of the most important collaborative filtering methods, matrix factorization is widely used in the field of recommender systems (Koren et al., 2022, 2009). It maps users and items to a low-dimensional latent factor space and predicts user ratings on items based on the inner product of low-dimensional feature vectors. However, because of the sparsity of the user-item rating matrix, collaborative filtering has inherent drawbacks. Some works attempt to alleviate data sparsity and cold-start issues with the assistance of social relationships, which is called social recommendation. Traditional social recommendation algorithms usually employ a matrix factorization framework to incorporate social information into the recommendation process. SoRec (Ma et al., 2008) integrates trust relationships between users and the ratings of users by sharing a matrix of latent user characteristics and performing a collaborative factorization of the rating matrix and the relationship matrix. SocialMF (Jamali and Ester, 2010) incorporates the trust propagation mechanism into a matrix factorization framework such that the latent feature vectors of each user and their immediate neighbors in the social network are close to each other. SoReg Ma et al. (2011) models social information as a regularization term to constrain matrix factorization. TrustMF (Yang et al., 2016) is based on matrix factorization, which factors the trust network in terms of trust and the trusted, mapping users into two low-dimensional trustor and trusted spaces separately. SREE (Li et al., 2017) incorporates social trust information into a matrix factorization method based on Euclidean embedding to personalize recommendations for users using the preferences of trusted users. To capture indirect social relations, InSRMF (Liu et al., 2019) proposes a joint recommendation model that effectively combines indirect social relation detection and matrix factorization to extract valuable indirect relations and improve recommendation performance. To alleviate the limitations of using only explicit data, TSSR (Shokeen and Rana, 2021) proposes a method that combines the rating matrix and social relations to extract implicit data and thus discover top-k semantic friends.

The above traditional social recommendation methods use information from user social networks to obtain a better vector representation of user features, which alleviates the data sparsity issue of collaborative filtering methods to a certain extent. However, the heterogeneity of social relationships and the higher-order nonlinear features of users and items are not taken into account in most works.

2.2 GNN-based social recommendation methods

In recent years, graph neural network techniques have demonstrated significant promise in the field of recommendation (Wu et al., 2022b; Gao et al., 2022). The key of GNN is to aggregate and disseminate information from neighbor nodes, which is naturally associated with social recommendation (Sharma et al., 2022). GraphRec (Fan et al., 2019) is the first work to apply GNN to social recommendation. It incorporates an attention mechanism into user-user graph modeling processes to consider heterogeneous strengths of social relations and utilizes GNN aggregate neighbor information to obtain the embedding representations of users and items, which are used in rating prediction. In order to leverage information from related items to further alleviate the data sparsity problem, DANSER (Wu et al., 2019) constructs dual graph attention networks to model the dual social effects of users and items. GraphRec+ (Fan et al., 2020) is an enhancement to GraphRec that aims to alleviate the data sparsity issue by utilizing the relationships between items. Specifically, it incorporates item-item graph modeling during the learning process to improve item representation learning. To model higher-order information in user-related graphs, DiffNet++ (Wu et al., 2020) designs a GNN-based model to simulate social influence diffusion in social graphs and interest influence diffusion in interest graphs. DICER (Fu et al., 2021) aims to avoid the limitations of shallow context-aware aggregation. It proposes a novel GNN-based approach to learn multi-relation and high-order neighbor information effectively that can model more precise user side interests and item side attraction. To consider the bias offsets of users and items, GDSRec (Chen et al., 2022) designs a decentralized collaborative filtering approach in which the statistics are taken into account in the graph modeling. GSFR (Xiao et al., 2022) proposes a graph social fusion recommendation method that captures multiple social information simultaneously. It uses a dynamic attention mechanism to model changes in user interests in heterogeneous networks and uses a mutualistic mechanism to learn virtual and actual user-user subgraphs to obtain more precise representations of user latent factors. Recently, some works have introduced contrastive learning as an assistance task in social recommendation, which helps to obtain more precise representations of users and items and thus improve recommendation performance. CGL (Zhang et al., 2022) utilizes contrastive learning to efficiently combine social information and interaction information, and a self-supervised loss and a supervised pointwise loss are introduced into the model to further alleviate the data sparsity. DISGCN (Li et al., 2022a)

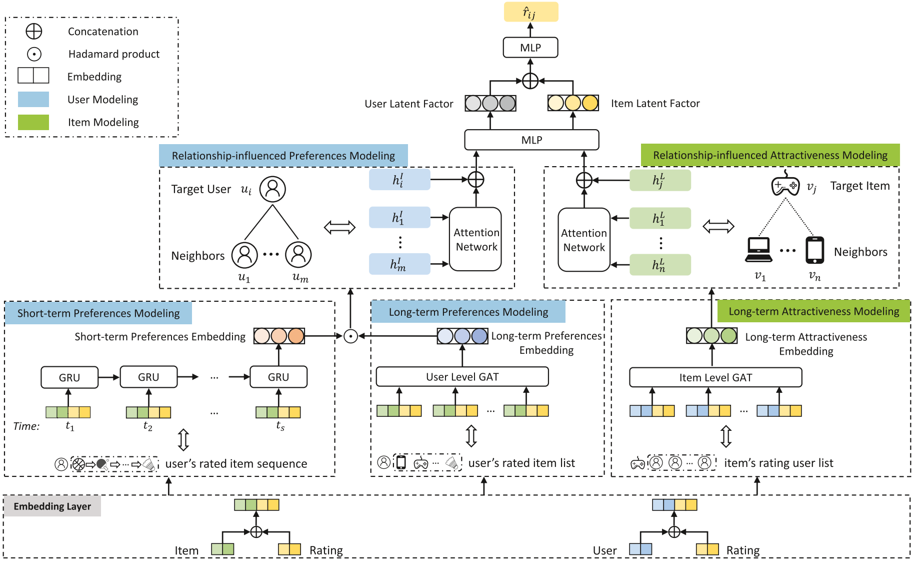


Fig. 2 The overview framework of the proposed model

designs a GCN-based embedding propagation mechanism to capture higher-order information in social-network graphs in two aspects: social homophily and social influence, and introduces a contrastive learning framework to assist in disentangling the effect of social influence. DcRec (Wu et al., 2022a) designs a disentangled contrastive learning framework to model the heterogeneous behavior patterns of users in the social domain and item domain. It can learn user representations from the two domains separately to obtain more precise representations of users and items, thus improving recommendation performance.

Although these methods have shown strong performance, they still have shortcomings. On the one hand, the above methods do not take into account the fact that user preferences in social recommendation may change dynamically. Capturing the dynamic trend of user preferences helps to better provide personalized recommendation services to users. On the other hand, constructing item implicit networks to capture the potential information and effects between items contributes to learning more precise representations of item latent factors. Some works calculate the similarity between items by using interactions or ratings and then connect items with high similarity. But we believe that item-item graphs constructed using extra attributes have more valuable information.

3 Deep latent representation enhancement method

This section gives the definition of social recommendation and describes DLREM model in detail. Figure 2 depicts the model framework. Our model can be divided into four parts: the embedding layer, the user modeling, the item modeling, and the rating prediction. Firstly, the embedding layer generates opinion-aware embeddings of interactions based on the original embeddings of users, items, and ratings. The outputs of the embedding layer are used as initial inputs for the user modeling and the item modeling, separately. Specifically, DLREM learns the long-term preferences of users using a GAT and the short-term preferences of users using gated recurrent unit (*GRU*). And the output of the GRU and the first GAT is used as the input to the second GAT. Then, DLREM learns the influence of social relationships on the preferences of the target user by using a GAT. Similarly, DLREM uses two GATs to learn the long-term attractiveness of items and the influence of same-category items on the target item, respectively. The output of the first GAT is the input of the second GAT, which is similar to user modeling. Secondly, the outputs of two parts are input to a MLP to get the latent factor representations of the target user and the target item. Lastly, the predicted rating is output through a MLP.

3.1 Formulation of social recommendation

Let $U = \{u_1, u_2, \ldots, u_M\}$ and $V = \{v_1, v_2, \ldots, v_N\}$ denote the set of users and items separately, where M denotes the number of users and N denotes the number of items. $R = [r_{ij}]_{M \times N}$ is assumed to be the user-item interaction matrix, which is also called the user-item interaction graph. Moreover, $T = \{< i,j >|\ r_{ij} \neq 0\}$ is used to denote the set of observed ratings and $F = \{< i,j >|\ r_{ij} = 0\}$ is used to denote the set of unobserved ratings.

In addition, we use $R_U(i)$ to denote the set of items that user i has interacted with and $R_V(j)$ to denote the set of users that have interacted with item j. Meanwhile, the interaction record of user i with item j at time t is denoted as a triplet (i, j, t), where t is the timestamp. The triplet of interaction records of each user i is ordered by the timestamp in ascending order as $R^t_U(i)$. For the item correlation graph $G_V = (V, E_V)$, where V is the set of items and E_V is the set of edges connecting two items that have an correlation. $N_V(j)$ is denoted as the set of items directly connected by item j in the item correlation graph G_V. For the user social relationship graph $G_U = (U, E_U)$, where U is the set of users and E_U is the set of edges connecting two users with social relationships. $N_U(i)$ is denoted as the set of users directly connected by user i in the social relation graph G_U.

So the social recommendation task is defined as: given the observed interaction records in the user-item interaction graph R and the user social relationship graph G_U, the goal of the task is to predict the unobserved interaction records in R, i.e., the possible rating values of the target user for the item.

3.2 Embedding layer

The user-item interaction diagram R includes information not only about the interaction between the user and the item, but also about the user's rating or opinion of the item (denoted by r). The rating reflects not only the user's preference for the item, but also the item's attractiveness to the user. As a result, incorporating rating information into the process of modeling latent user and item factors can lead to a more accurate representation of users and items. In general, ratings are discrete values. Rating value $r_{ij} \in \{1, 2, 3, 4, 5\}$ in a five-level rating platform.

In these works (Fan et al., 2019, 2020), they provide enlightening methods for capturing interactions and ratings. Our approach defines user representation matrix as $P = \{p_i\}_{D \times M}$, where D is the embedding dimension and p_i denotes the embedding vector of user i. Similarly, $Q = \{q_j\}_{D \times N}$ denotes item representation matrix, where q_j denotes the embedding vector of item j. In addition, each rating is mapped into the corresponding D dimensional vector separately and e_{ij} is used to denote the embedding vector of r_{ij}. The computing process for the opinion-aware embedding of the interaction between user i and item j is given below:

$$x_{ij} = f\left(\left[q_j \oplus e_{ij}\right]\right), \tag{1}$$

$$y_{ji} = f\left(\left[p_i \oplus e_{ij}\right]\right), \tag{2}$$

where x_{ij} denotes the embedding of the opinion-aware interaction from user i to item j for user modeling. Similarly, y_{ji} represents the embedding of the opinion-aware interaction from item j to user i for item modeling. f is a multi-layer perceptron (*MLP*) and $\oplus$ denotes the concatenation operation.

3.3 User modeling

The target of user modeling is to model the representation of user preferences from different graph views. DLREM first learns about the long-term and short-term preferences of users from the user-item interaction graph and then learns about the influence of relationships on user preferences from the user-user relationship graph.

3.3.1 Long-term preferences modeling

In the user space, the long-term preference representation of users is learned by aggregating the items that user u interacts with and their opinions on these items. Considering that each interaction between a user and an item contributes differently to the long-term preference representation of users, our method designs a user level GAT to learn the long-term preference representation h_i^L of user i, the specific calculation procedure is as follows:

$$h_i^L = \sigma\left(W_0 \sum_{j \in R_U(i)} \alpha_{ij} x_{ij} + b_0\right), \tag{3}$$

where σ is the nonlinear activation function of user level GAT, W_0 and b_0 denote the weight and bias of user level GAT, separately. α_{ij} denotes the attention weight of the interaction between user i and item j, which is computed by a two layer neural network as follows:

$$\alpha_{ij} = \frac{\exp\left(W_2 \cdot \sigma\left(W_1 \cdot \left[p_i \oplus x_{ij}\right] + b_1\right) + b_2\right)}{\sum_{j \in R_U(i)} \exp\left(W_2 \cdot \sigma\left(W_1 \cdot \left[p_i \oplus x_{ij}\right] + b_1\right) + b_2\right)}, \tag{4}$$

where $\left(W_1, b_1\right)$ and $\left(W_2, b_2\right)$ correspond to the weights and biases of the first and second layer of the neural network, separately.

3.3.2 Short-term preferences modeling

The rating opinion information is also introduced into the modeling of short-term preferences of users. Recurrent neural networks are used to model sequence $R_U^t(i)$ in order to capture the short-term preferences of users. Specifically, a GRU is used in our model to learn the sequence as follows:

$$h_i^S = GRU\left(R_U^t(i)\right), \tag{5}$$

where GRU denotes a multi-layer gated recurrent unit. Since our model only tries to capture the dynamic trend of user preferences in the short term, which is denoted as h_i^S, DLREM only uses the output of GRU of the last hidden layer. Longer sequences of items are truncated to save memory and time overhead.

Long-term and short-term preferences are fused to obtain the overall preference representations of the users. The calculation is given as follows:

$$h_i^I = h_i^L \odot h_i^S, \tag{6}$$

where h_i^I denotes the overall preference representations of user i and $\odot$ is the Hadamard product, which denotes the product of the elements of two vectors.

3.3.3 Relationship-influenced preferences modeling

On the one hand, the preferences of users are usually similar to or influenced by those of their direct social friends. On the other hand, social relationships have varying degrees of strength, which means that different friends have different levels of influence over users. The social relationship-based preferences h_i^R of users i are computed as follows:

$$h_i^R = \sigma\left(W_0 \sum_{u \in N_U(i)} \beta_{iu}\, h_u^I + b_0\right), \tag{7}$$

$$\beta_{iu} = \frac{\exp\left(W_2 \cdot \sigma\left(W_1 \cdot \left[p_i,\ h_u^I\right] + b_1\right) + b_2\right)}{\sum_{u \in N_U(i)} \exp\left(W_2 \cdot \sigma\left(W_1 \cdot \left[p_i,\ h_u^I\right] + b_1\right) + b_2\right)}, \tag{8}$$

where h_u^I denotes the overall preference representation of user u and user u is a friend of user i. β_{iu} represents the degree of influence among users.

3.4 Item modeling

Item modeling aims to model the representation of item attractiveness from two graph views. In this section, we will detail how to learn the long-term attractiveness representation of items from the user-item interaction graph and how to use the item correlation graph to enhance the representation of items.

3.4.1 Long-term attractiveness modeling

In the item space, the long-term attractiveness representation of items is modeled by aggregating the users who interact with item j and the opinions of users on items. Similarly, DLREM designs an item level GAT to learn the long-term attractiveness representation h_j^L of item j, specifically as follows:

$$h_j^L = \sigma\left(W_0 \sum_{i \in R_V(j)} \alpha_{ji} y_{ji} + b_0\right), \tag{9}$$

$$\alpha_{ji} = \frac{\exp\left(W_2 \cdot \sigma\left(W_1 \cdot \left[q_j, y_{ji}\right] + b_1\right) + b_2\right)}{\sum_{i \in R_V(j)} \exp\left(W_2 \cdot \sigma\left(W_1 \cdot \left[q_j, y_{ji}\right] + b_1\right) + b_2\right)}, \tag{10}$$

where α_{ji} denotes the attention weight of the interaction between item j and user i.

3.4.2 Relationship-influenced attractiveness modeling

There are usually many similarities between items in the same category, and these similarities are a reflection of the attractiveness of the items. Therefore, it is reasonable to further enrich the representation of item from the item correlation graph.

Construct its corresponding correlation graph G_V for each item is the first step. Specifically, different items in the same category are picked as direct neighbors of the target item. Considering the large number of items in the same category, this work uses the simplest method of random sampling, which is to randomly select K items from the same category. This paper discusses the effect of parameter K on the performance of model in the Section 4.4.1. We will explore better selection methods in our future work. Then, DLREM learns the relationship-influenced attractiveness h_j^R of item j as follows:

$$h_j^R = \sigma\left(W_0 \sum_{o \in N_V(j)} \beta_{jo}\, h_o^L + b_0 \right), \tag{11}$$

$$\beta_{jo} = \frac{\exp\left(W_2 \cdot \sigma\left(W_1 \cdot \left[q_j,\ h_o^L\right] + b_1\right) + b_2\right)}{\sum_{o \in N_V(j)} \exp\left(W_2 \cdot \sigma\left(W_1 \cdot \left[q_j,\ h_o^L\right] + b_1\right) + b_2\right)}, \tag{12}$$

where h_j^R denotes the attractiveness representation of item j based on the influence of o of the associated item. β_{jo} denotes the influence weight between items.

3.5 Rating prediction and model training

This section firstly describes how to obtain latent factor representations from user modeling and item modeling and then introduces rating prediction and the model training method.

Latent factor representations of users and items are taken as input for the rating prediction module. The latent factor representations of users are obtained by fusing long-term and short-term preferences and preferences based on relational influences. Similarly, the latent factor representations of items are obtained by fusing long-term attractiveness and relationship-influence-based attractiveness. The latent factor representations h_i^{user} for user i and h_j^{item} for item j are calculated separately as follows:

$$h_i^{user} = f\left(\left[\, h_i^I \oplus h_i^R \right]\right), \tag{13}$$

$$h_j^{item} = f\left(\left[\, h_j^L \oplus h_j^R \right]\right), \tag{14}$$

where $\oplus$ denotes the concatenation operation. DLREM is applied to the rating prediction task of social recommendation. The possible rating $\hat{r}_{ij}$ for item j by user i is calculated as follows:

Table 1 Statistics of the two datasets

Dataset	Users	Items	Ratings	Category	Social Links
Ciao	2379	16861	36065	6	57544
Epinions	22167	296277	922267	27	355813

$$\hat{r}_{ij} = g\left(\left[h_i^{user} \oplus h_j^{item}\right]\right), \tag{15}$$

where g is a MLP. The target training function for this work is given as follows:

$$L = \frac{1}{2|T|} \sum_{(i,j)\in T} \left(\hat{r}_{ij} - r_{ij}\right)^2, \tag{16}$$

where r_{ij} is the true rating of item j by user i.

4 Experiments

This section will firstly describe the experimental setup in detail, then analyze the overall performance comparison, and finally conduct ablation experiments and parameter sensitivity studies.

4.1 Experiment settings

4.1.1 Datasets

DLREM is evaluated on two real-world datasets, Ciao and Epinions[1], which have been widely used as benchmark datasets for social recommendation. These two datasets contain users, items and item categories, ratings and timestamps of when the ratings occurred, and social relationships, where the ratings are from 1 to 5. The statistics of Ciao and Epinions are shown in the Table 1. In addition, the users and items with fewer than five interactions in both datasets, as well as users without social friends, are removed.

4.1.2 Evaluation metrics

To evaluate the performance of DLREM in the rating prediction task, this paper uses two widely used metrics (Sharma et al., 2022; Shokeen and Rana, 2020), MAE and RMSE, to evaluate the prediction accuracy of the model. Smaller values of MAE and RMSE indicate higher prediction accuracy for the model, specifically defined as follows:

$$MAE = \frac{1}{|T|} \sum_{(i,j)\in T} |\hat{r}_{ij} - r_{ij}|, \tag{17}$$

[1] Ciao and Epinions available from http://www.cse.msu.edu/~tangjili/trust.html.

$$RMSE = \sqrt{\frac{1}{|T|}\sum_{(i,j)\in T}\left(\hat{r}_{ij} - r_{ij}\right)^2}. \tag{18}$$

4.1.3 Baselines

To compare the performance, a comparison is made with two representative groups of methods. For the fairness of the experiments, the methods we chose were proposed for application to the rating prediction task.

Traditional Social Recommendation Methods:

- SoRec *(CIKM2008)* (Ma et al., 2008): performs a collaborative factorization of the user-item rating matrix and the user-user social relationship matrix.
- SoReg *(WSDM2011)* (Ma et al., 2011): introduces social network information into the traditional matrix factorization framework and models it as a regularized term to capture strong dependencies.
- SocialMF *(RecSys2010)* (Jamali and Ester, 2010): introduces a trust propagation mechanism in the matrix factorization framework.
- SREE *(IJCNN2017)* (Li et al., 2017): adds social network information to the matrix factorization method based on the Euclidean embedding.

GNN-based Social Recommendation Methods:

- Graphrec *(WWW2019)* (Fan et al., 2019): utilizes GNN to model the representation of users and items in social recommendation for rating prediction.
- Danser *(WWW2019)* (Wu et al., 2019): proposes a dual graph attention network model to learn the representation of dual social effects in users and items.
- Graphrec+ *(TKDE2020)* (Fan et al., 2020): is an improvement of Graphrec to better learn the representation of items in social recommendation by adding the modeling of item-item graph.
- GDSRec *(TKDE2022)* (Chen et al., 2022): treats rating bias as a vector and takes it into account in the process of modeling user- and item-related graph structure data.
- GNNDSR *(DASFAA2022)* (Lin et al., 2022): proposes GNN-based social models with dynamic and static representations to learn representations of users and items.

4.1.4 Parameter settings

This work uses RMSprop as an optimizer to optimize the objective function and the Dropout strategy to alleviate the overfitting problem during the model training. In this paper, 80% and 60% of the dataset are selected as the training datasets to train the model parameters, separately, and the rest of dataset is equally divided into a validation dataset for adjusting the hyperparameters and a test dataset for the final performance comparison. The learning rate and batch size are searched in $\left\{10^{-5}, 10^{-4}, 10^{-3}, 5\times 10^{-3}\right\}$ and $\{64, 128, 256\}$, separately. The embedding size D is tested in $\{16, 32, 64, 128, 256\}$ and the number of sample neighbors K is manually set in $\{10, 20, 30, 40, 50, 60\}$. The number of hidden layers of GRU is experimentally set to 4. DLREM will stop training if the sum of MAE and RMSE increases by 5 epochs in a row on the validation set.

Table 2 The overall performance of each model (mean±std. Deviation) on Ciao and Epinions, where the evaluation metrics are MAE and RMSE

Comparison methods	Ciao(80%)		Ciao(60%)		Epinions(80%)		Epinions(60%)	
	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓
SoReg	0.7686±0.005	1.0373±0.005	0.802±0.006	1.079±0.007	0.8848±0.001	1.1562±0.001	0.9095±0.001	1.1752±0.002
SocialMF	0.7615±0.004	1.0358±0.001	0.7782±0.002	1.058±0.003	0.8657±0.001	1.1449±0.002	0.8835±0.001	1.1708±0.002
SoRec	0.7494±0.003	1.0256±0.008	0.7804±0.002	1.065±0.002	0.8588±0.004	1.1457±0.004	0.9003±0.002	1.1577±0.002
SREE	0.7488±0.001	0.9734±0.001	0.7623±0.001	0.9827±0.001	0.8428±0.001	1.0905±0.001	0.8507±0.001	1.0998±0.001
Graphrec	0.7311±0.002	0.9586±0.002	0.7329±0.002	0.9686±0.002	0.8201±0.003	1.0683±0.001	0.8292±0.006	1.0692±0.006
Danser	0.7275±0.003	0.9463±0.003	0.734±0.002	0.9555±0.002	0.8222±0.002	1.088±0.002	0.8411±0.002	1.112±0.002
Graphrec+	0.7146±0.001	0.9473±0.001	0.7276±0.001	0.9585±0.002	0.8127±0.004	1.0637±0.002	0.8277±0.002	**1.067±0.002**
GDSRec	0.7023±0.001	0.9438±0.001	0.7106±0.005	**0.945±0.002**	0.8174±0.002	1.063±0.001	0.8288±0.002	1.072±0.001
GNNDSR	0.6834±0.009	0.947±0.009	0.7229±0.003	0.9645±0.002	0.8028±0.001	1.0713±0.003	0.8151±0.002	1.073±0.002
DLREM	**0.6681±0.004***	**0.9248±0.006****	**0.7031±0.001***	0.962±0.003	**0.7946±0.004****	**1.058±0.002****	**0.8019±0.003****	1.0691±0.003
p-value	1e-2	3.09e-4	1e-2	-	6.76e-4	3.9e-3	7.85e-5	-

The p-value is used to test for statistical significance. * and ** indicate the statistical significance for $p < 0.05$ and $p < 0.01$, respectively, compared to the best baseline. Best results are shown in bold

Table 3 Effect of item graph construction strategy on Ciao and Epinions datasets

Comparison methods	Ciao(80%)		Epinions(80%)	
	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓
DLREM(Similarity)	0.694	0.9653	0.8025	1.0693
DLREM	0.6681	0.9248	0.7946	1.058
Improv.	3.73%	4.19%	0.98%	1.05%

4.2 Overall performance

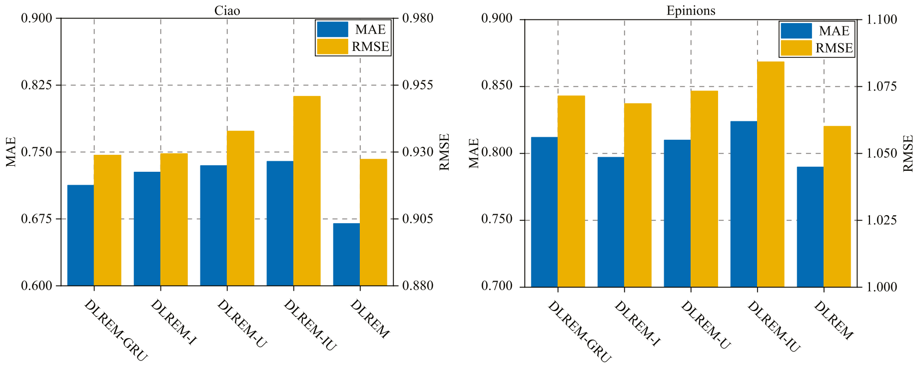
Table 2 shows the performance comparison of the different methods for the rating prediction task. Our observations from comparison and analysis are as follows.

- In the baseline methods, the GNN-based methods perform significantly better than the matrix factorization-based methods. This suggests that graph neural networks have very good learning abilities for graph-structural data representation.
- When the training set accounts for 80% of the dataset, DLREM outperforms all other baseline methods, proving the effectiveness of our model. Compared to the best baseline model, our method improves the MAE and RMSE by 2.23% and 2.01%, separately, on the Ciao dataset and by 1.02% and 0.47%, separately, on the Epinions dataset. Despite the small relative percentages of improvement, (Koren, 2008) points out that even minor improvements in MAE and RMSE can have a significant effect on recommendation results in practice. The performance improvement is mainly attributed to the fact that our approach provides advanced components for modeling short-term user preferences and a novel method for constructing item correlation graphs.
- When the training set accounts for 60% of the dataset, in the baseline approach, DLREM is not optimal in the RMSE metric but still optimal in the MAE metric, which improves by 1.05% and 1.61% on the two datasets, separately. On the one hand, this is because noise may be introduced when capturing short-term preferences of users when the training data is small, RMSE is sensitive to outliers. On the other hand, this validates the effectiveness of our approach to capturing the short-term preferences of users. When the dataset is large enough, our method can accurately capture dynamic trends in user preferences and better learn latent factor representations of users, resulting in better prediction results.

4.3 Ablation study

4.3.1 The effect of main components

To investigate the effectiveness of the different components of our model, it is compared with the following variants. We only conduct ablation experiments when the training set comprises 80% of the dataset. 1) DLREM(Similarity): Replace the item graph construction strategy in our model with the similarity-based strategy in Graphrec+ and GNNDSR. 2) DLREM-GRU: Only the long-term preferences of users are modeled. 3) DLREM-I: Remove the modeling of item attractiveness based on the effect of correlation relationships. 4) DLREM-U: Remove the modeling of user preferences



(a) Ciao (b) Epinions

Fig. 3 Effect of different components on Ciao and Epinions datasets

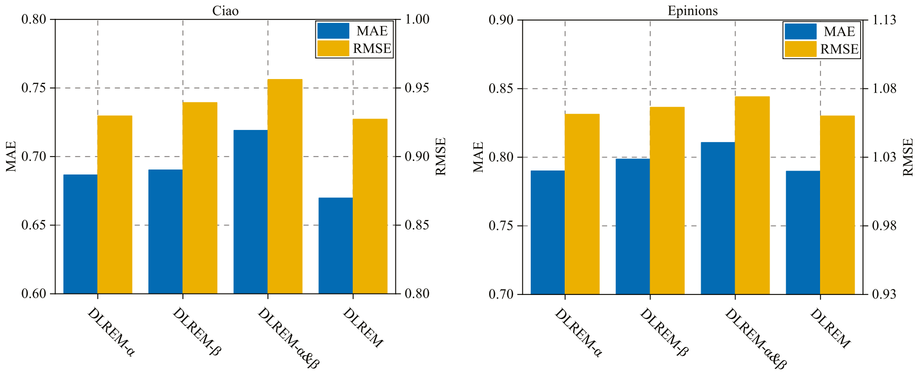
based on social relationship influence. 5) DLREM-IU: Remove the modeling of relationship-based effect representations of users and items.

Table 3 shows that DLREM has better performance than DLREM(Similarity), which proves that the strategy of constructing the item graph in our method is more reasonable. Moreover, Fig. 3 shows that DLREM always outperforms DLREM-GRU, which suggests that capturing short-term preferences of users is meaningful. It is interesting to note that we observe that DLREM-I consistently outperforms DLREM-U on both datasets. This suggests that the influence of the relationship between users has a bigger influence on the recommendations than the relationship between items. Meanwhile, DLREM-I and DLREM-U consistently outperform DLREM-IU, indicating that reasonably modeling the influence of user-side and item-side relationships can improve recommendation performance nicely. In conclusion, DLREM consistently and significantly outperforms all variants, demonstrating the importance of capturing dynamic trends in user preferences as well as the efficacy of our approach in modeling the effect of item-to-item relationships.

4.3.2 The effect of attention mechanisms

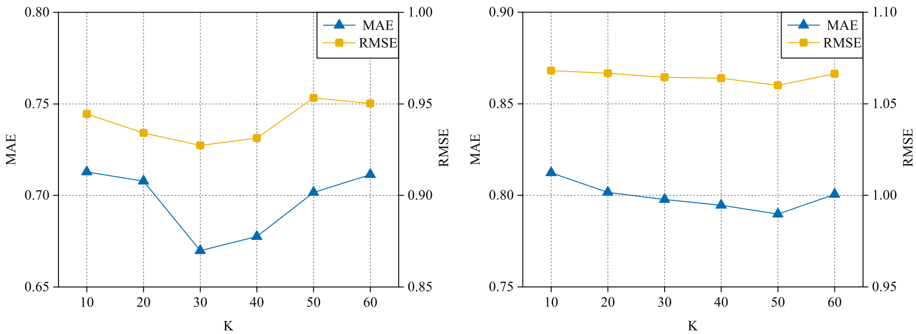
To better understand our model, this section further evaluates the effect of attention mechanisms on modeling user and item representations based on relational influences. DLREM is compared with the following variants. 1) DLREM-α: Remove the learning module on user relationship heterogeneity and model the effect from user friends equally.2) DLREM-β: Remove the learning module on the heterogeneity of item relationships and model the effect from linked items equally. 3) DLREM-$\alpha\&\beta$: Remove both user-side and item-side relational heterogeneity learning modules.

The results in Fig. 4 show that all variants are less effective than the original model. This not only indicates that social relationships between users are heterogeneous, with different social friends having different influence weights on preferences of users, but also indicates that the influence of relationships between items is equally heterogeneous.



(a) Ciao (b) Epinions

Fig. 4 Effect of attention mechanisms on Ciao and Epinions datasets



(a) Ciao (b) Epinions

Fig. 5 Effect of parameter K on Ciao and Epinions datasets, where K is the number of item neighbors in the item correlation graph

4.4 Parameter study

This section performs parameter sensitivity experiments on DLREM for key hyperparameters. The number K of item neighbors is explored for its effect on performance, which is a key parameter in item correlation graph construction. In addition, the effect of parameter D on the performance is also investigated, which is the embedding dimension of users and items. We only conduct parameter sensitivity experiments when the training set comprises 80% of the dataset.

4.4.1 The effect of parameter K

The results of the study about the parameter K are shown in Fig. 5. As the value of K gradually increases, the performance of our model on both datasets first gradually improves, which indicates that a reasonable use of the influence of the relationship between items facilitates learning a better latent factor representation of the items, thus improving the

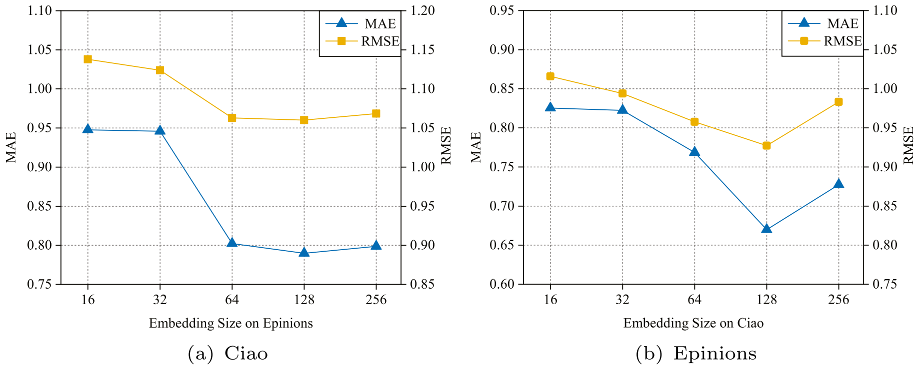


Fig. 6 Effect of parameter D on Ciao and Epinions datasets, where D is the embedding dimension of users and items

performance of our model. However, when the value of K increases to a certain value, the performance of the model starts to degrade, probably because introducing too many item neighbors in the relationship graph introduces noise to the model. It is interesting to note that in the Ciao dataset, the turning point of K is 30, while in the Epinions dataset, the turning point of K is 50. This may be due to the fact that Epinions contains more items and interaction records, and therefore the influence of relationships with more associated items needs to be considered when modeling item latent factors.

4.4.2 The effect of embedding dimension

The embedding size D is also a key parameter that affects the performance and complexity of the model, which is searched in $\{16, 32, 64, 128, 256\}$. As the experimental results in Fig. 6 show, it can be observed that as the embedding size gradually increases, both MAE and RMSE decrease to different degrees, and the model performance gradually improves, reaching a peak when the embedding size is taken as 128. However, as the embedding size increases further, the performance decreases, which indicates an overfitting situation. Therefore, we need to find a suitable embedding size D to balance the performance and complexity of the model.

5 Conclusion

This paper proposes DLREM that can enhance deep latent representations of users and items in social recommendation. Specifically, DLREM employs GAT to learn long-term and relationship-influence-based preferences of users from user-item and user-user graphs, separately, and employs gated recurrent units to capture short-term preference trends of users in user modeling, and finally fuses the preference representations from the three aspects to obtain the final latent factor representation of users. Moreover, DLREM constructs item-item graphs based on item category information and models user-item graphs and item-item graphs using GAT to obtain the final latent factor representation of items in item modeling. Extensive experiments on two public datasets show that our model outperforms mainstream social recommendation models on the rating prediction task. In the future, we will investigate the incorporation of more auxiliary information in the user and item learning processes to alleviate the issue of sparse data and improve the interpretability of recommendation results.

Acknowledgements This work was supported by National Natural Science Foundation of China (No. 62272290, 62172088), and Shanghai Natural Science Foundation (No. 21ZR1400400).

Author contributions All authors contributed to the study conception and design. The main manuscript text was written by Xiaoyu Hou and Bofeng Zhang. The key code was written by Xiaoyu Hou and Guobing Zou. The experiments were conducted by Xiaoyu Hou and Guobing Zou. The data analysis was conceived and designed by Sen Niu. All authors read and approved the final manuscript.

Funding National Natural Science Foundation of China (No. 62272290, 62172088), Shanghai Natural Science Foundation (No. 21ZR1400400).

Availability of supporting data Ciao and Epinions are openly available datasets and can be downloaded from http://www.cse.msu.edu/~tangjili/trust.html.

Declarations

Ethical approval Not applicable

Competing interests Not applicable

References

Chen, C., Zhang, M., & Liu, Y., et al. (2019a). Social attentional memory network: Modeling aspect-and friend-level differences in recommendation. *Proceedings of the twelfth ACM international conference on web search and data mining,* pp. 177–185. https://doi.org/10.1145/3289600.3290982.

Chen, C., Zhang, M., Wang, C., et al. (2019b). An efficient adaptive transfer neural network for social-aware recommendation. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 225–234. https://doi.org/10.1145/3331184.3331192.

Chen, J., Xin, X., & Liang, X., et al. (2022). Gdsrec: Graph-based decentralizedcollaborative filtering for socialrecommendation. *IEEE Transactions on Knowledge and Data Engineering,* pp. 4813–4824. https://doi.org/10.1109/TKDE.2022.3153284.

Chen, T., & Wong, R.C.W. (2021). An efficient and effective framework for session-based social recommendation. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining,* pp. 400–408. https://doi.org/10.1145/3437963.3441792.

Du, J., Ye, Z., & Yao, L., et al. (2022). Socially-aware dual contrastive learning for cold-start recommendation. *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 1927–1932. https://doi.org/10.1145/3477495.3531780.

Fan, W., Ma, Y., & Li, Q., et al. (2019). Graph neural networks for social recommendation. *The world wide web conference,* pp. 417–426. https://doi.org/10.1145/3308558.3313488.

Fan, W., Ma, Y., & Li, Q., et al. (2020). A graph neural network framework for social recommendations. *IEEE Transactions on Knowledge and Data Engineering,* pp. 2033–2047. https://doi.org/10.1109/TKDE.2020.3008732.

Fu, B., Zhang, W., Hu, G., et al. (2021). Dual side deep context-aware modulation for social recommendation. *Proceedings of the Web Conference 2021*, pp. 2524–2534. https://doi.org/10.1145/3442381.3449940

Gao, C., Wang, X., & He, X., et al. (2022). Graph neural networks for recommender system. *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining,* pp. 1623–1625. https://doi.org/10.1145/3488560.3501396.

Jamali, M., & Ester, M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks. *Proceedings of the fourth ACM conference on Recommender systems,* pp. 135–142. https://doi.org/10.1145/1864708.1864736.

Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining,* pp. 426–434. https://doi.org/10.1145/1401890.1401944.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer,* pp. 30–37. https://doi.org/10.1109/MC.2009.263.

Koren, Y., Rendle, S., & Bell, R. (2022). Advances in collaborative filtering. *Recommender systems handbook,* pp. 91–142. https://doi.org/10.1007/978-1-0716-2197-4_3.

Li, N., Gao, C., & Jin, D., et al. (2022a). Disentangled modeling of social homophily and influence for social recommendation. *IEEE Transactions on Knowledge and Data Engineering,* pp. 5738–5751. https://doi.org/10.1109/TKDE.2022.3185388.

Li, W., Gao, M., & Rong, W., et al. (2017). Social recommendation using euclidean embedding. *2017 International Joint Conference on Neural Networks (IJCNN),* pp. 589–595. https://doi.org/10.1109/IJCNN.2017.7965906.

Li, Y., Zhan, Z., Li, H., et al. (2022b). Interest-aware influence diffusion model for social recommendation. *Journal of Intelligent Information Systems,* pp. 363–377. https://doi.org/10.1007/s10844-021-00684-3.

Lin, J., Chen, S., Wang, J. (2022). Graph neural networks with dynamic and static representations for social recommendation. *International Conference on Database Systems for Advanced Applications,* pp. 264–271. https://doi.org/10.1007/978-3-031-00126-0_18.

Liu, C., Li, Y., Lin, H., et al. (2022). Gnnrec: Gated graph neural network for session-based social recommendation model. *Journal of Intelligent Information Systems,* pp. 1–20. https://doi.org/10.1007/s10844-022-00733-5.

Liu, H., Jing, L., Yu, J., et al. (2019). Social recommendation with learning personal and social latent factors. *IEEE transactions on knowledge and data engineering,* pp. 2956–2970. https://doi.org/10.1109/TKDE.2019.2961666.

Ma, H., Yang, H., & Lyu, M.R., et al. (2008). Sorec: social recommendation using probabilistic matrix factorization. *Proceedings of the 17th ACM conference on Information and knowledge management,* pp. 931–940. https://doi.org/10.1145/1458082.1458205.

Ma, H., Zhou, D., & Liu, C., et al. (2011). Recommender systems with social regularization. *Proceedings of the fourth ACM international conference on Web search and data mining,* pp. 287–296. https://doi.org/10.1145/1935826.1935877.

Marsden, P.V., Friedkin, N.E. (1993). Network studies of social influence. *Sociological Methods & Research,* pp. 127–151. https://doi.org/10.1177/0049124193022001006.

McPherson, M., Smith-Lovin, L., Cook, J.M. (2001). Birds of a feather: Homophily in social networks. *Annual review of sociology,* pp. 415–444. https://doi.org/10.1146/annurev.soc.27.1.415.

Sharma, K., Lee, Y.C., & Nambi, S., et al. (2022). A survey of graph neural networks for social recommender systems. arXiv preprint, https://doi.org/10.48550/arXiv.2212.04481.

Shokeen, J., & Rana, C. (2020). Social recommender systems: techniques, domains, metrics, datasets and future scope. *Journal of Intelligent Information Systems,* pp. 633–667. https://doi.org/10.1007/s10844-019-00578-5.

Shokeen, J., & Rana, C. (2021). A trust and semantic based approach for social recommendation. *Journal of Ambient Intelligence and Humanized Computing,* pp. 10,289–10,303. https://doi.org/10.1007/s12652-020-02806-1.

Wu, J., Fan, W., & Chen, J., et al. (2022a). Disentangled contrastive learning for social recommendation. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management,* pp. 4570–4574. https://doi.org/10.1145/3511808.3557583.

Wu, L., Li, J., Sun, P., et al. (2020). Diffnet++: A neural influence and interest diffusion network for social recommendation. *IEEE Transactions on Knowledge and Data Engineering,* pp. 4753–4766. https://doi.org/10.1109/TKDE.2020.3048414.

Wu, Q., Zhang, H., Gao, X., et al. (2019). Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. *The World Wide Web Conference,* pp. 2091–2102. https://doi.org/10.1145/3308558.3313442.

Wu, S., Sun, F., Zhang, W., et al. (2022b). Graph neural networks in recommender systems: a survey. *ACM Computing Surveys,* pp. 1–37. https://doi.org/10.1145/3535101.

Xiao, X., Wen, J., Zhou, W., et al. (2022). Multi-interaction fusion collaborative filtering for social recommendation. *Expert Systems with Applications,* p. 117610. https://doi.org/10.1016/j.eswa.2022.117610.

Yang, B., Lei, Y., Liu, J., et al. (2016). Social collaborative filtering by trust. *IEEE transactions on pattern analysis and machine intelligence,* pp. 1633–1647. https://doi.org/10.1109/TPAMI.2016.2605085.

Yu, J., Yin, H., Li, J., et al. (2020). Enhance social recommendation with adversarial graph convolutional networks. *IEEE Transactions on Knowledge and Data Engineering,* pp. 3727–3739. https://doi.org/10.1109/TKDE.2020.3033673.

Zhang, Y., Huang, J., Li, M., et al. (2022). Contrastive graph learning for social recommendation. *Frontiers in Physics,* p. 35. https://doi.org/10.3389/fphy.2022.830805.

Authors and Affiliations

Xiaoyu Hou[1] · Guobing Zou[1] · Bofeng Zhang[2] · Sen Niu[2]

Xiaoyu Hou
houxiaoyu_1998@shu.edu.cn

Sen Niu
niusen@sspu.edu.cn

[1] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China

[2] School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai 201209, China