The Journal of Supercomputing (2023) 79:8295–8321
https://doi.org/10.1007/s11227-022-04985-4

MRAN: a attention-based approach for social recommendation

Yiyang Fu[1] · Xiaojun Xie[2] · Tao Zhang[2,3]

Accepted: 28 November 2022 / Published online: 16 December 2022

Abstract

Graph Neural Networks have been widely used in social recommendation systems. However, with the increase of graph nodes and diffusion depth, they tend to suffer from graph sparsity and over-smoothing, which inhibit their performance. In this work, we propose the multi-relational attention network, named as MRAN, for social recommendation. Our model has three distinctive characteristics: (i) it alleviates the data sparsity problem in social recommendation scenarios by incorporating both user social relations and item homogeneous relations as supplementary information; (ii) it mimics the structure of influence diffusion in user and item domain via an iteratively aggregating structure; (iii) it has a two-level attention mechanism at the diffusion and aggregating level, enabling it to differentiate importance of embeddings to overcome the over-smoothing problem. Experiments conducted on two large-scale representative datasets demonstrate that the proposed model outperforms previous methods substantially. The ablation study shows that the performance of MRAN can be further improved avoid over-smoothing by increasing the diffusion depth.

Keywords Recommender system · Social network · Graph attention network

✉ Yiyang Fu
fuyiyang@sjtu.edu.cn

Xiaojun Xie
xiaojunxie6@126.com

Tao Zhang
taozhang@jiangnan.edu.cn

[1] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, 800 Dongchuan RD. Minhang District, Shanghai 200240, Shanghai, China

[2] School of Artificial Intelligence and Computer, Jiangnan University, No. 1800, Lihu Avenue, Wuxi 214122, Jiangsu, China

[3] China Ship Scientific Research Center, No. 222, Shanshui East Road, Binhu District, Wuxi 214122, Jiangsu, China

1 Introduction

With the rapid development of computer technology and the maturity of the Internet economy, recommendation system has become a hot topic for researchers. Traditional recommendation systems mainly face the problem of data sparsity: only a few interactions can be observed in the data. In practical applications, the number of users and items is huge, but the historical behavior records between users and items are few. Therefore, when the number of users and items in the recommendation system increases, the user preferences cannot be accurately learned from the extremely sparse historical behavior matrix. The accuracy of the recommendations will be greatly reduced. This problem also leads to the cold start problem, which means that it is difficult to provide accurate personalized recommendations for newly added users or projects.

The problem of data sparsity can be alleviated by merging the information of users' social neighbors. With the rise of online social applications, users on social platforms can establish their social circles. The social neighbors in these circles will influence the user's purchase decision to a certain extent, thus leading to the social recommendation scenario. For example, users may consult friends when considering whether to buy products. In addition, in social networks, users tend to establish social relationships with other users with similar behavioral preferences. Therefore, in social networks, the behavior similarity between user pairs with links is higher than that between user groups without links. Therefore, the goal of social recommendation is to integrate social network information into the recommendation system, solve the sparse data problem and improve the recommendation performance.

In order to explore the interests of users, methods are proposed to learn the hidden state of user interests such as the traditional matrix factorization. With the development of deep learning, deep models have been used to extract more accurate features of users and items to improve the recommendation accuracy. Researchers have designed some more advanced neural network recommendation models based on the traditional matrix factorization methods in recent years. Some works use neural networks to model the deeper complex relationship between the user embedding and the item embedding, which has improved the recommendation accuracy [1]. Graph-based neural networks are used to model the user preference generation process, and predict the current latent interests and preferences of users [2, 3]. Compared with the traditional models, these models can extract the implicit characteristics of users and items to improve the recommendation accuracy.

The recommendation system in the social network is one of the core supporting technologies of many social network applications by capturing users' interests and hobbies in the social network to enable them to obtain personalized information services. However, these researches only consider and analyze some of the elements. Especially, the social attributes are not emphasized enough. The shortcomings of social recommendation multi-relational attention network research can be elaborated and analyzed from two aspects: objective factors and subjective

factors. Most of the existing recommendation algorithms assume that there is only one user relationship in the social network, but in fact, there are many types of user relationships in the social network and its extended applications, such as interpersonal relationships, including friends, colleagues, etc., and interactive relationships, including common purchase of the same goods, online communication, etc. Different relationships play different roles in specific recommendation tasks. It is very important to make full use of multiple user relationships to achieve high-quality social network recommendation systems. Beside, the data sparsity problem and the over-smoothing problem didn't not been solved well.

In this paper, we focus on the social recommendation methods. In order to alleviate the data sparsity problem of the traditional recommendation methods and the over-smoothing problem of the existing social recommendation methods, we propose a new method MRAN based on graph neural network-based method MRAN, which jointly models the three graphs, namely the user-user graph, the item-item graph, and the user-item graph, as shown in Fig. 1. The main contributions of this paper are as follows:

First, we propose a new social recommendation framework called Multi-Relational Attention Network (MRAN) for social recommendation, which jointly captures the influence and interest diffusion in multi-relational context neighbors;

Secondly, we introduce homogeneous information between items to solve the problem of data sparsity and simulate the high-order influence diffusion process in the context of multiple relationships;

Thirdly, we propose a two-level attention mechanism to select the most distinctive feature and the most important neighbor. So that it can distinguish the importance of embedding, so as to overcome the problem of over-smoothing.

The experimental results show that our proposed framework outperforms the existing methods on two real-world datasets. The ablation study verified that the proposed method is effective for over-smoothing.

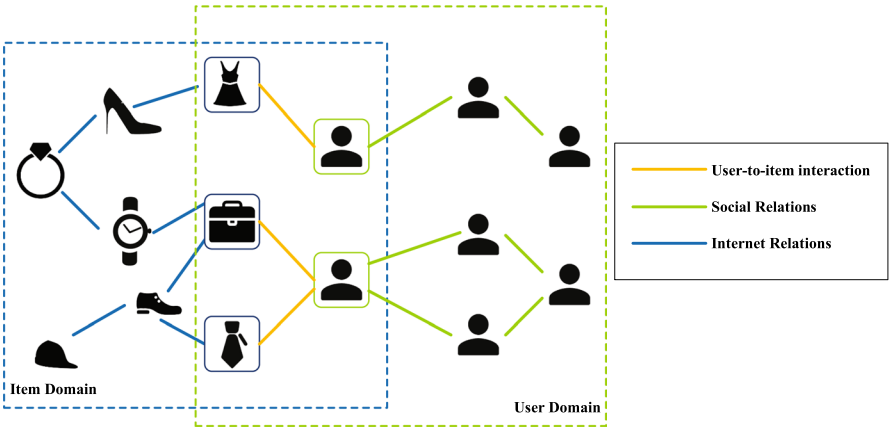

Fig. 1 The graph data in social recommendation. The graph data contain three graphs including the user-user graph (left part), the user-item graph (middle part) and the item-item graph (right part). Note that the edges of user-item graph denote the historical behavioral records

The rest of this paper is organized as follows. In Sect. 2, we reviewed some work related to the social recommendation. We defined the social recommendation problem to be solved and formally described the proposed framework in Sect. 3. The experimental results and discussion are presented in Sect. 4. And then we conclude this paper in Sect. 5 and pointed out the future research directions. Finally, we discuss the advantages and disadvantages of our proposed method in Sect. 6.

2 Related work

In this subsection, we briefly review the related works about the social recommendation in three categories, i.e., classical collaborative filtering recommendation models, matrix factorization-based social recommendation models, and the recent graph neural network-based recommendation models.

2.1 Classical CF recommendation models

There are two main types of collaborative filtering methods [4], i.e., (i) memory-based collaborative filtering, which calculates the similarity between users and items through users' rating history, and then new items are recommended for users according to the similarity. Typical examples of this approach are neighborhood-based CF and item-based/user-based top-N recommendations [5, 6]; and (ii) model-based collaborative filtering models, which are developed using different machine learning algorithms to predict users' rating of unrated items [7]. Once the model training phase is finished, the model-based CF can predict the ratings of users very quickly.

There are many model-based collaborative filtering methods, among which the most common method is the matrix factorization-based model. Known as the latent factor model, it compresses the user-item matrix into a low-dimensional representation in terms of latent factors (LFM) [8]. It can alleviate data sparsity using dimensionality reduction techniques and usually produce more accurate recommendations than the memory-based CF approaches [9]. Assuming that the user-item rating matrix is $R \in \mathbb{R}^{n*m}$ (where each row represents a user and each column represents an item), the matrix factorization algorithm usually learns two low-rank matrices $U \in \mathbb{R}^{n*k}$ and $V \in \mathbb{R}^{m*k}$, so that

$$R \approx \hat{R} = UV^T \tag{1}$$

where $\hat{R}$ represents the approximation matrix of R, U represents the user's latent feature matrix (each row represents a user's feature vector), and V represents the item's latent feature matrix (each row represents an item's feature vector). Generally speaking, the rank k of two characteristic matrices U and V is far less than n and m, so the above matrix factorization is also called low rank matrix factorization. After learning U and V, the user a rating for the item i can be predicted according to the following formula [10–12]:

$$\hat{r}_{ai} = u_a v_i^T \tag{2}$$

where u_a is the $a-th$ row of the user embedding matrix U, which is the embedding of user a. Similarly, v_i denotes the latent embedding of item i in $i-th$ row of item embedding matrix V. In order to learn the optimal matrix representation $U \in \mathbb{R}^{n*k}$ and $V \in \mathbb{R}^{m*k}$,, additional L2-norm regularization terms are incorporated in the overall optimization function as [8]:

$$L = \sum_{a=1}^{n} \sum_{i=1}^{m} (r_{ai} - u_a v_i^T)^2 + \mu \|U\|_F^2 + \mu \|V\|_F^2 \tag{3}$$

where the first term is the approximation error of matrix factorization, the second and third terms are regularization terms, used to prevent overfitting of the model, and μ is the regularization coefficient.

Since model-based recommendation models significantly reduce the memory requirement and computation complexity, SVD [13], matrix factorization (MF) [14, 15]and non-negative matrix factorization (NFM) [16] are widely used, which all take advantage of LFM.

2.2 Matrix factorization-based social recommendation models

As more and more social behaviors happen on the Internet, people have realized that social information is of great use for recommendation systems and therefore social recommendations have emerged as one of the hottest research topics these days [17–19]. Traditional recommendation systems assume that users are independent and identically distributed (i.i.d), which subconsciously ignores the social interaction between users. Therefore, in the context of social networks, recommender systems not only need to focus on the relationship between users and items, but also on the relationship between users.

Considering the interaction between users, social information is introduced to improve the traditional recommendation models, and a series of social recommendation models are proposed. We classify matrix factorization-based social recommendation methods into two major categories according to the representation methods of the user characteristic matrix.

The first category is based on the shared representation of the user characteristic matrix, which means that a user characteristic matrix is used to model with the user-item rating matrix and user-user social matrix separately. By assuming the user characteristic matrix is hidden in both rating information and social information, the objective function of SoRec [19] can be written as:

$$F_{SoRec} = \sum_{r_{ai} \neq 0} (r_{ai} - g(u_a v_i^T))^2 + \lambda_u \sum_{S_{ab} \neq 0} (S_{ab} - g(u_a z_b^T))^2 + \lambda_r (\|U\|_F^2 + \|V\|_F^2 + \|Z\|_F^2) \tag{4}$$

where $g(x) = \frac{1}{1+exp(-x)}$ is a logistic function, $z_b \in \mathbb{R}^d$ is the social attribute representation of user b, which is the b-th row of the social attribute matrix $Z \in \mathbb{R}^{n*d}$ and $u_a z_b^T$ denotes the predicted social relationship between user a and user b, which is fitted by the user feature vector u_a and social feature vector z_b. Different from SoRec, TrustMF [20] is a social recommendation model that divides the trust information

into trust and trusted relationship, which maps each user to two different d-dimensional feature vectors according to the directivity of trust relationship. Father more, Fang et al. [21] modeled the trust information from four dimensions, i.e., benevolence, integrity, competence and predictability. In addition, Tang et al. [22] proposed a social recommendation model LOCABAL, which integrated local and global social information. Social relations from different perspectives are applied to recommendation systems to improve recommendation performance.

The second category enhances the representation of the user characteristic matrix by taking social relationship into account [23]. Guo et al. introduced social information into the SVD++ model, and proposed TrustSVD model [24]. In this model, both user-user social relationship and user-item rating information are regarded as the implicit feedback information, and social feedback information is added to the original SVD++ model to reconstruct the objective function.

2.3 Graph neural network-based recommendation models

Graph Neural Networks (GNNs) [25], as a generalization of deep neural networks on graph data, are able to extract and represent data characteristics in graph field. GNN has derived many powerful variants, such as GCN [26], GAT [27], GraphSAGE [28] and so on. Compared with the traditional deep learning methods, GNN can characterize entities and their relationships through the graph structure. By assigning attributes to the nodes of the generated graph and constantly updating the state of the nodes continuously, the graph neural network obtains the state containing the information of adjacent nodes and the topological characteristics of the graph. These nodes are finally output through specific methods to obtain the final node embedding.

The development of graph neural networks provides a better way for people to further analyze the entities of recommendation systems and the relationship between them. In recent years, the related research based on graph neural networks in recommender systems has attracted more and more attention from scholars and has achieved good results. GC-MC [29] proposed a graph auto-encoder framework, which extracts the latent representation of users and items from a bipartite graph between the user and item nodes and solves the problem of rating prediction in recommendation systems from the perspective of link prediction. STAR-GCN [3] adopted a stack of GCN encoder-decoders introduced the reconstruction mechanism to extract latent factors of users and items and proposed a new training strategy to tackle the problems of label leakage and cold start in GC-MC. The main point of RS in this phase is to model the equal influence of first-order neighbors from a user-item bipartite graph and the last layer of node representation is used for rating prediction.

Seeing that the classical CF-based methods only use the embedding of user and item for collaborative recall, the latent relationship (cooperative signal) is not encoded in the user-item interaction data. This may cause the embedding to be not powerful enough to capture the effect of collaborative filtering. NGCF [2] generalized GC-MC by integrating user-item interaction (i.e., user-item bipartite graph) into the embedding process and taking the higher-order collaborative signals between

users and items into account during the users/items embedding process. Similarly, PinSage [30] is a random-walk-based GCN that explicitly models the high-order connectivity in the web-scale bipartite graph, so that the cooperative signal is injected into the embedded process in an explicit way.

GNN-based methods can be divided into GCN (graph evolutionary network) based and GAT (graph attention network) based methods. They often use the neighbor knowledge of entities to encode the structure of the knowledge map. Most of the neighbors and the input features as embedded modules. Because there is an assumption that aligned entities will have similar neighbors. Most GNN-based methods only use entities as alignment seeds in training, rather than relationships as alignment seeds. The disadvantage of GCN lies in its poor flexibility, transitivity, and scalability. In addition, we can use verification sets to help improve performance, which is a little contrary to its original intention of semi-supervised learning. The training is full batch, which is difficult to expand to large-scale networks and converges slowly. GraphSAGE aims to improve the scalability of GCN and improve the defects of training methods. It aims to learn an aggregator rather than a representation for each node, which can improve the flexibility and generalization of the model. In addition, thanks to flexibility, it can train in batches to improve the convergence speed. The main advantages of GAT are: (i) By assigning different weights to nodes in the same neighborhood, the model scale can be expanded. (ii) The model weight value is shared, which can well handle the unseen nodes in the sub-graph and can also execute the transitive and inductive tasks. (iii) Compared with GraphSAGE, it does not need to fix the sampling size, and the algorithm handles the entire neighborhood. (iv) Sampling node characteristics calculate the similarity, not the structural characteristics of nodes, so as to pre-calculate without knowing the graph structure. (v) Multi head self-attention mechanism is used, which is convenient for parallel and efficient. The main disadvantages of GAT are: (i) It is too smooth to handle high-order features. (ii) The maximum size of the receptive field is affected by the depth of the model. (iii) Due to the high overlap of domain nodes, redundant computing occurs. The above methods apply GNNs over the user-item interaction data without considering social connection information. For comparison, GraphRec [31] proposed a unified framework for jointly modeling user/item embeddings in user-user social network and user-item bipartite graph separately. The key points of the paper include integrating the user-user graph and user-item graph, better capturing the connection between user, items and the user's rating of items, and using Attention network to distinguish the importance of social relations. They utilized attention mechanism [27]and concatenation operation to distinguish essential elements and each user's local neighbors' preferences effectively to alleviate the data sparsity for optimizing the recommendation task. However, the social influence and interest preference might not be fully extracted since GraphRec only harvests one-order information among neighborhood nodes in graph structure. Moreover, DiffNet [32] assumed that users' preferences are recursively influenced by their trusted social neighbors and modeled the high-order social influence diffusion process with a recursive influence propagation structure.

Learn the representation of the user's social graph and the user-item bipartite graph, such as DiffNet, GraphRec, and MHCN. First, learn the user representation

from the social graph and user-item graph respectively, and then combine the representation with the sum pool. MLP or attention mechanism. DiffNet++ [33] further recursively modeled interest diffusion process for the item nodes, and employed the attention mechanism in two deep diffusion processes respectively to aggregate the different order neighbors' feature vectors for each node. DiffNet++ uses a typical method to learn a unified user representation diagram, including social graph and user-item bipartite graph. It first uses the GAT mechanism to aggregate the information in the user-item sub graph and the social sub graph, and then combines the representation with the designed multi-level attention network at each layer. MHCN uses GCN to propagate on the constructed hypergraph to obtain high-order social relations. Methods with attention mechanisms, such as GraphRec and DiffNet++, assume that the social impact of different neighbors on the social graph is different, and give different weights to the social impact of different friends. In social recommendation, user representation is learned from two different perspectives, namely social impact and user interaction. In combination with user representation, there are two strategies: (i) learn the representation of user social graph and user item bipartite graph; (ii) Learn the unified user representation diagram, including social graph and user item bipartite graph. Using the first strategy, such as DiffNet, GraphRec, and MHCN, first learn the user representation from the social graph and user item graph respectively, and then combine the representation with the sum pool. MLP or attention mechanism. DiffNet++is a typical method using the second strategy. It first aggregates the information in the user-item sub graph and social sub graph by using the GAT mechanism, and then combines the representation with the designed multi-level attention network at each layer.

Traditional social recommendation systems usually directly use this relationship as a regularizer to constrain users' final representation or as input to enhance users' original embedding. This approach only considers the influence of users' first-order neighbors and ignores the recursive diffusion of higher-order influences in social networks. Our work summarized the existing relationships recommended by the society into four aspects, and multi-level GNN captured the multi-level impact information in these four aspects. In real life, users may be influenced by their friends. Graph neural network can simulate how users are affected by recursive social diffusion process. The construction methods of graphs are divided into two categories: stacked graphs and Hypergraphs to capture high-order relationships in social graph. In addition, attention mechanism and connection operation are introduced for personalized recommendation.

3 The proposed model

In this section, we will first elaborate the notation convention used in this paper in Table 1. Then, we formulate the problem to be solved. What's more, we will give an overview about the architecture of our proposed model MRAN, and then detail each component of the model. Finally, the training process of MRAN are discussed.

Table 1 Notations used in this paper

Symbols	Definitions and descriptions
P_a	The free latent embedding of user a
q_i	The free latent embedding of item i
x_a	The real-valued attributes for user a
y_i	The real-valued attributes for item i
d	The length of the embedding vector
S	The user-user social network
S_a	The set of social friends whom user a follows
F	The item-item homogeneous network
F_i	The set of items that item i similar with
R	The user-item interaction matrix
$R_u(i)$	The set of users who have interacted with item i
$R_I(a)$	The set of items with which user a has interacted
τ	A fixed threshold limiting the number of users who liked both items in F
r_{ai}	The observed preference of item i by user a
$\hat{(r_{ai})}$	The predicted preference of item i by user a
$\oplus$	The concatenation operator of two vectors
G_s	The user-user social graph
G_I	The user-item interest graph
G_F	The item-item influence graph

3.1 Problem definition

Let $U = \{u_1, u_2, \ldots, u_n\}$ and $V = \{v_1, v_2, \ldots, v_m\}$ be the sets of users and items, where N and M are numbers of users and items respectively. We consider a user-item interaction matrix $R \in \mathbb{R}^{n*m}$ denoting users' implicit preference and interests to items. We denote $r_{ai} = 1$ if u_a is interested in v_i, otherwise $r_{ai} = 0$. In addition, we use $R_U(i)$ and $R_I(u)$ to respectively denote the set of users who have interacted with v_i and the set of items which u_a has interacted with. Moreover, we assume the existence of a user-user directed graph $G = [U, \mathbb{R}^{n*n}]$, where U is the set of users and S represents the connections between users of a social network. We denote $s_{ab} = 1$ if u_a follows or trusts u_b and zero otherwise. Also, we use S_a to denote the set of users with whom u_a directly connects, i.e., $S_a = [b|s_{ab} = 1]$. Similarly, we define an item-item directed graph $G = [U, F \in \mathbb{R}^{m*m}]$, where F denotes the similarity relationship among items, and we use F_i to denote the set of items that item i similar with. Following[33], we use an embedding vector $x_a \in \mathbb{R}^d$ to denote the real-valued attributes(e.g., user profile) for u_a and an embedding vector $y_i \in \mathbb{R}^d$ to denote the real-valued attributes(e.g., item text representation) for v_i, where d is the length of embedding vector, i.e., there are two matrixes of entities: a user attribute matrix $X \in \mathbb{R}^{n*d}$, and an item attribute matrix $Y \in \mathbb{R}^{m*d}$. The mathematical notations we use are summarized in Table 1. We now formally formulate the social recommendation problem as follows:

Input: a user set U, an item set V, the user-item interaction matrix R, the user-user social network S, the item-item homogeneous network F and the real-valued attribute matrices X and Y of users and items.

Output: a preference predicting function that maps a user-item pair to a real value, i.e., $\hat{R} = f(U, V, R, S, F, X, Y)$, where $\hat{R} \in \mathbb{R}^{n*m}$ denotes the unobserved interactions of users to items.

3.2 An overview of the proposed model

The overall architecture of the proposed model is shown in Fig. 2. Generally speaking, the proposed MRAN model consists of three components, i.e., user modeling, item modeling, and rating prediction. At the beginning of user modeling and item modeling, we first integrate free embedding and feature embedding to get the initial low-dimension user/item representation. Compared with the traditional social recommendation methods, our model not only leverages user-item interaction graph G_I and social network G_S, but also introduces item-item relation graph G_F to enhance the item presentations. So far, we simultaneously consider both influence diffusion and interest diffusion in the process of user modeling and item modeling. Besides, four aggregations are introduced to handle these two different diffusion processes in user/item modeling respectively. Moreover, multi-layer GNNs with two different attention mechanism capture the multi-order influence information among these four aspects. Specifically, at each layer k, by taking embedding u_a^{k-1} of user a and embedding v_i^{k-1} of item i as input, these layers recursively output the updated embeddings of v_i^k and u_a^k through the diffusion operations. This iteration step starts at k=0 and stops when the recursive process reaches a pre-defined depth K. Finally, in the process of rating prediction, each user's preference for items will be predicted by

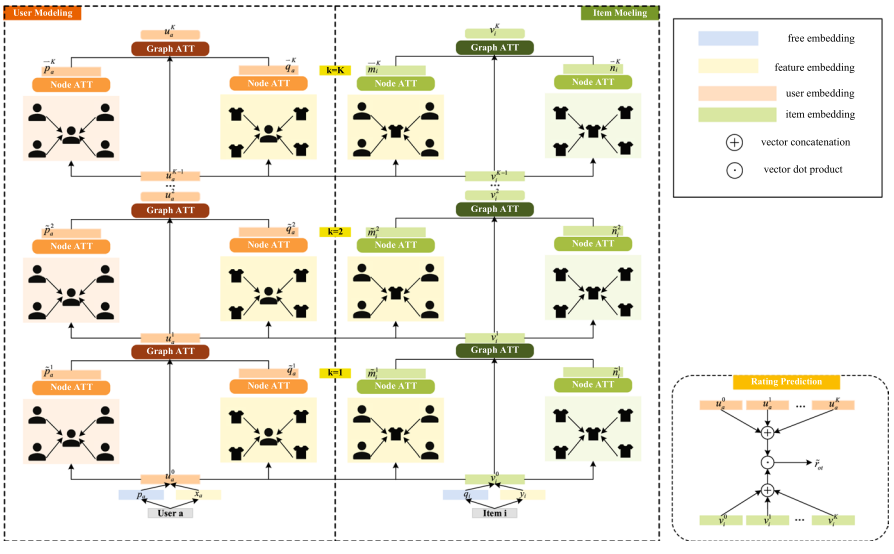

Fig. 2 The overall architecture of the proposed MRAN model

integrating user and item modeling components and calculating their dot product. Next, we describe the details of each component in order.

3.3 User modeling

We define $P \in \mathbb{R}^{n*d}$ as the free embedding matrices of users, where d is the embedding size and p_a denotes the free latent embedding for user a, i.e., the $a-th$ row of matrix P. By feeding p_a and the associated feature vector x_a into the fusion layer, the initial latent preference of user a can be captured as:

$$\mu_a^0 = \sigma(\mathbf{W}_1 * [\mathbf{p}_a, \mathbf{x}_a]) \tag{5}$$

where $\sigma(\cdot)$ is the activation function, $\mathbf{W}_1$ is a trainable transformation matrix and we omit the bias term for symbolic simplification.

General GNN-based social recommendation methods leverage two different graphs, i.e., a user-user social graph G_S and a user-item interest graph G_I as input. Inevitably, two aggregations are introduced to process these two different graphs in user modeling. We define $\tilde{p_a^k}$ as the aggregated embedding of influence diffusion from the trusted social neighbors in G_S and $\tilde{q_a^k}$ as the embedding of interest diffusion from the interested item neighbors in G_I at the $k-th$ layer. Therefore, it is convenient for us to be able to model user $a's$ updated embedding u_a^k from different perspectives as:

$$\mathbf{u}_a^k = (\gamma_{a1}^k \mathbf{u}_a^{k-1} + \gamma_{a2}^k \tilde{\mathbf{p}}_a^k + \gamma_{a3}^k \tilde{\mathbf{q}}_a^k) \tag{6}$$

$$\tilde{\mathbf{p}}_a^k = \sum_{b \in S_a} \alpha_{ab}^k \mathbf{u}_b^{k-1} \tag{7}$$

$$\tilde{\mathbf{q}}_a^k = \sum_{i \in R_I(a)} \beta_{ai}^k \mathbf{v}_i^{k-1} \tag{8}$$

where u_a^{k-1} denotes the latent embedding of user a at the $(k-1)-th$, $\tilde{p}_a^k$denotes the user-based social influence diffusion process and $\tilde{q}_a^k$ denotes the item-based interest influence diffusion process from two graphs respectively. Specifically, α_{ab}^kdenotes the social influence of user b to a at the $k-th$ layer in G_S, β_{ai}^k denotes the interest influence of item i to user a at the $k-th$ layer in G_I, and γ_{al}^k denotes the graph level weight that learns to fuse and aggregate information from different aspects. A naive aggregation function is the mean operator, which sets all weights to equal values, i.e., $\gamma_{a1}^k = \gamma_{a2}^k = \gamma_{a3}^k$, $\alpha_{ab}^k = \frac{1}{|S_a|}$, and $\beta_{ai}^k = \frac{1}{|R_a|}$. It assumes that all interactions contribute equally in the aggregation process. However, this may not be optimal as the influence of interactions on users can be very different.

Here we employ two different attention mechanisms to select the most discriminative features and the most important neighbors. Specifically, the node-level weights, i.e., the social influence strengths α_{ab}^k and the interest influence strengths β_{ai}^k, concretely point out the strength of each connection with user a in two graphs while graph-level

weights focus on how user a balances the social influence and interest influence for user embedding iterative updating. The situation of different users is different, because some users are more easily influenced by people they trust, while the interests of other users may be quite stable. Therefore, it is necessary for each user to build personalized weight. In the user-space, we argue that similar users have similar feature representation, so we take the related two users' embeddings at the $(k-1)-th$ layer as input, and cosine similarity denotes the social influence strength α_{ab}^{k} as:

$$\alpha_{ab}^{k} = \frac{\mu_a^{k-1} \cdot \mu_b^{k-1}}{|\mu_a^{k-1}| * |\mu_b^{k-1}|} \tag{9}$$

$$\alpha_{ab}^{k} = softmax(\alpha_{ab}^{k}) = \frac{exp(\alpha_{ab}^{k})}{\sum_{b \in S_a} exp(\alpha_{ab}^{k})} \tag{10}$$

We use a softmax function that transforms each value into range (0,1). Similarly, we calculate the interest influence score β_{ai}^{k} by taking related user embedding and item embedding as input. We feed it through a dot product layer to calculate the probability that user a is interested in the given item i as:

$$\beta_{ai}^{k} = \sigma(\mu_a^{k-1} \odot v_i^{k-1}) \tag{11}$$

$$\beta_{ai}^{k} = softmax(\beta_{ai}^{k}) = \frac{exp(\beta_{ai}^{k})}{\sum_{i \in R_I(a) exp(\beta_{ai}^{k})}} \tag{12}$$

where σ denotes the sigmoid function, and the weight β_{ai}^{k} is normalized by a softmax function. This product operation can help to focus on the candidate item and model dynamic interest influence under a specific context which is related to the final rating prediction operation. In contrast to the node attention layer, we can model the graph attention weights of $\gamma_{al}^{k}(l = 1, 2, 3)$ as:

$$\gamma_{a1}^{k} = MLP_1^{k}(\mu_a^{k-1}) \tag{13}$$

$$\gamma_{a2}^{k} = MLP_2^{k}([\mu_a^{k-1}, \tilde{p}_a^{k}]) \tag{14}$$

$$\gamma_{a3}^{k} = MLP_3^{k}([\mu_a^{k-1}, \tilde{q}_a^{k}]) \tag{15}$$

where three MultiLayer Perceptrons (MLPs) are used to learn the graph attention weights with the related user embedding at the $(k-1)-th$ layer (u_a^{k-1}) and node attention representations at the $k-th$ layer [$\tilde{p}_a^{k}$ and $\tilde{q}_a^{k}$]. Without confusion, we omit the normalization operation of all attention modeling in the following expressions, as all of them share the similar form as shown in Eq. (10). In addition, considering $\gamma_{a1}^{k} + \gamma_{a2}^{k} + \gamma_{a3}^{k} = 1$, if the value of γ_{a2}^{k} is larger than that of γ_{a3}^{k}, the effect of influence diffusion is greater than that of interest diffusion, and larger $\gamma_{a2}^{k} + \gamma_{a3}^{k}$ denotes

that user embedding at layer k will be more affected by the two influence diffusion effects. As a result, the user embedding at the $(k-1)$ layer is less retained during the user's embedding updating process.

3.4 Item modeling

We define $Q \in \mathbb{R}^{m*d}$ as the free embedding matrices of items, where d is the embedding size and q_i denotes the free latent embedding for item i, i.e., the $i-th$ row of matrix $\mathbf{Q}$. By feeding q_i and the associated feature vector y_i into the fusion layer, the initial item embedding is defined as:

$$v_i^0 = \sigma(\mathbf{W_2} * [q_i, y_i]) \tag{16}$$

Since there is no prior information that explicitly captures the relationship between items, most existing methods simply leverage user' rating history to process item embedding or do not process it. In this paper, we attempt to build an item-item homogeneous network F which is similar to the user-user social network S. We calculate the similarity between the two items by the number of common users who liked them [34, 35]. For any item i and item j, we define their similarity coefficients s_{ij} as the number of users who liked both items, and item i is related to item j if $s_{ij} > \tau$ with τ a fixed threshold. We define the item implicit network as the graph $G_F = [V, F \in \mathbb{R}^{m*m}]$, where V is the set of items and F represents the connections between the two related items of an item-item homogeneous network.

Likewise, we use a similar method as shown in user modeling. For each item i, we need to aggregate user-space information from the set of users who have interacted with item i, denoted as $R_U(i)$, and aggregate item-space information from the set of item friends that have related with item i, denoted as F_i. For each item i, given its $(k-1)-th$ layer embedding u_a^{k-1} and v_i^{k-1}, we model the updated item embedding v_i^k at the $k-th$ layer from G_F and G_I as:

$$v_i^k = (\eta_{i1}^k v_i^{k-1} + \eta_{i2}^k \tilde{m}_i^k + \eta_{i3}^k \tilde{n}_i^k) \tag{17}$$

$$\tilde{m}_i^k = \sum_{j \in F_j} \mu_{ij}^k v_j^{k-1} \tag{18}$$

$$\tilde{n}_i^k = \sum_{a \in R_U(i)} v_{ia}^k u_a^{k-1} \tag{19}$$

where $R_U(i) = [a|r_{ia} = 1]$ is the user set that rates item i, $F_i = [j|f_{ij} = 1]$ is the item set that is related to item i, $\tilde{m}_i^k$ is the item $i's$ aggregated embedding from its neighbor items in the item-item influence graph G_F, $\tilde{n}_i^k$ is the item $i's$ aggregated embedding from its neighbor users in the user-item interest graph G_I, and $\eta_{il}^k (l = 1, 2, 3)$ denotes the aggregation weight. The design of item modeling is very similar to user modeling, which aims to acquire a global view of both user aggregation and item aggregation.

By taking the item node i as the central node, we calculate the interest attention weights $\tilde{n}_i^k$ between node i and its user node neighbors, and the influence attention weights $\tilde{m}_i^k$ between node i and its related item node neighbors. Officially, the influence scores between the target item's node representation v_i^{k-1} and all of its selected neighbors are defined as:

$$\mu_{ij}^k = \frac{v_i^{k-1} \cdot v_j^{k-1}}{|v_i^{k-1}| * |v_j^{k-1}|} \tag{20}$$

$$v_{ia}^k = \sigma(v_i^{k-1} \odot u_a^{k-1}) \tag{21}$$

The graph attention weights $\gamma_{al}^k (l = 1, 2, 3)$ in user-space could be found in Eq. (6), here we use an attention network to learn the item graph attention weight $\eta_{il}^k (l = 1, 2, 3)$ of Eq. (17) by:

$$\eta_{i1}^k = MLP_4^k(v_i^{k-1}) \tag{22}$$

$$\eta_{i2}^k = MLP_5^k(v_i^{k-1}, \tilde{m}_i^k) \tag{23}$$

$$\eta_{i3}^k = MLP_6^k(v_i^{k-1}, \tilde{n}_i^k) \tag{24}$$

where three other MLPs are used to learn the graph attention weights with the related item embedding at the $(k-1)-th$ layer (v_i^{k-1}) and node attention representations at the $k-th$ layer($\tilde{m}_i^k$ and $\tilde{n}_i^k$).

3.5 Rating prediction

We have obtained a series of user and item latent factors through K times of iterative diffusion process. With the latent embedding of user a and item i at layer k (i.e., u_a^k and v_i^k) for $k = [0, 1, 2, \ldots, K]$, we can first concatenate them at each layer to get the final user embedding $u_a^* = [u_a^0 \| u_a^1 \| \ldots \| u_a^K]$ and the final item embedding $v_i^* = [v_i^0 \| v_i^1 \| \ldots \| v_i^K]$. Then, the rating of user a towards item i could be predicted as the inner product between the final user and item embeddings:

$$\tilde{r}_{ai} = [\mathbf{u}_a^0 \| \mathbf{u}_a^1 \| \ldots \| \mathbf{u}_a^K]^T [\mathbf{v}_i^0 \| \mathbf{v}_i^1 \| \ldots \| \mathbf{v}_i^K] \tag{25}$$

3.6 Model training

To estimate model parameters of MRAN, we adopt Bayesian Personalized Ranking loss (BPR) loss [12] for training, which is widely used for ranking task [33, 36, 37]. The loss function is formulated as:

$$L = min_{\Theta} \sum_{(a,i^+,i^-)\in R} -\ln \sigma(\hat{r}_{ai^+} - \hat{r}_{ai^-}) + \lambda \|\Theta\|_2^2 \tag{26}$$

where $R = \{(a, i^+, i^-)|(a, i+) \in R^+, (a, i^-) \in R^-\}$ is the training set, R^+denotes the set of positive samples (observed user-item interactions) and R^- denotes the set of negative samples (unobserved user-item interactions that follow a random sampling strategy). $\sigma(x)$is sigmoid function and Θ is regularization parameters set, i.e., $\Theta = [P, Q, W_1, W_2, [MLP_i^k]_{i=1,2,3,4,5,6}]$. Since all the parameters are differentiable, we use back propagation algorithm to optimize our model, and more detailed descriptions about the parameter setting will be given at experiment part.

4 Experiments

To comprehensively evaluate the effectiveness of our proposed model MRAN, we conduct experiments on two real-world datasets aiming to answer the following research questions:

RQ1: How does the performance of our model compared with baselines?
RQ2: How do the attention mechanisms affect model performance?
RQ3: How does the model performance benefit from the diffusion depth?

4.1 Experiment setup

4.1.1 Datasets

To avoid experimental bias, we selected two independent datasets, for algorithm validation, Yelp and Flickr.

Yelp. Users in Yelp (http://www.yelp.com) can rate local services and follow others that they want to follow. The original dataset contains two parts of information, i.e., the directed interactive relationships among users, as well as the users' ratings to locations. There are five levels of ratings from 1 to 5 (higher is better). Similar to many works [32], we regard the ratings larger than 3 as "My Likes" of this user.

Flickr. Flickr is an online photo-sharing website (http://www.flickr.com). Users follow other users and share interesting images based on their preference to their friends, family and social media followers. The original dataset provides a large amount of preference information and social information.

We evaluate our proposed model on two representative datasets Yelp and Flickr. Same as what they have done in their study, we only keep users who have at least 2 rating records and 2 social links and filter items which have been interacted less than 2 times. In addition, we conduct additional preprocessing step by extracting similar pairs of items that are preferred by at least 2 users and accept this as side information of our model. Note that item-pairs (Item Connections) are very sparse, we further take available links into consideration. Statistics of the final datasets are summarized in Table 2. We randomly select 85% of the data for training, 5% for validation, and the remaining 10% for testing.

Table 2 The statistics of the two datasets

Dataset 1	Yelp	Flickr
# of Users	17,237	8,358
# of Items	38,342	82,120
# of Ratings	204,448	327,815
# of Density(Ratings)	0.03%	0.05%
# of Social Connections	143,765	187,273
# of Density (Social Relations)	0.05%	0.27%
# of Item Connections	79,876	498,664
# of Density (Item Relations)	0.011%	0.015%

4.1.2 Evaluation metrics

In order to evaluate the top-K recommendation performance of the models, we adopt a recall-based metric HR@K (Hit Ratio) and a ranking-based metric NDCG@K (Normalized Discounted Cumulative Gain), which are widely used in top-K recommendation tasks [33, 38]. Specifically, HR@K measures the percentage of the testing items being successfully recommended in the top-K recommendation lists and NDCG@K further takes the ranking position of testing items within the top-K recommendation list into account. For both metrics, bigger values indicate better recommendation results. As many recommendation tasks [32, 39], in our experiments, for each user we randomly sample 1000 unrated items as negative items. We repeat each experiment 10 times and report the average score of the best performance for both metrics.

4.1.3 Baselines

To evaluate the performance, we compare our MRAN against ten state-of-the-art baselines including traditional CF methods, social based recommender approaches and graph neural network based models. For each group, we select representative baselines detailed as below.

BPR [12]: A typical pair-wise algorithm that is derived from the maximum posterior estimator, only using the interaction data between users and items.

FM [10]: A powerful matrix factorization method which considers pairwise feature interactions.

SocialMF [40]: A matrix factorization technique with trust propagation for recommendation in social networks.

TrustSVD [24]: A social recommendation method that incorporates first order social relations into modeling process.

ContextMF [41]: A fast and context-aware embedding learning method for social recommendation.

GraphRec [31]: A network embedding approach that employs attention mechanism to encode social network.GraphRec's model mainly includes three components:

user modeling, item modeling, and rating prediction. Next, we will look at these core contents in detail. The proposed GraphRec method is always superior to all baseline methods. Compared with DeepSoR and GCMC+SN, GraphRec provides advanced model components to integrate ratings and social network information.

PinSage [30]: A random-walk Graph Convolutional Network that is highly-scalable and capable of learning embeddings for nodes in web-scale graphs containing billions of objects.Compared with the deep learning baseline method, PinSage can generate higher quality recommendations. For recommended tasks, the click through rate of PinSage is 150% higher than that of the best baseline method, and the MRR is 60% higher.

NGCF [2]: A deep neural network based framework leveraging high-order signals in user-item bipartite graph.NGCF is proposed to solve the problem that the cooperative signal in user-item interactions cannot be expressed in the embedded layer.

DiffNet [32]: A graph neural network based model that simulates social influence propagation.DiffNet can use GNN to capture the deeper social diffusion process. However, the model also has limitations: (i) The assumption of the same impact is not suitable for real scenarios; (ii) The model can also be enhanced by interactive users if the item representation is ignored.

DiffNet++ [33]: A Neural Influence and Interest Diffusion Network for social recommendation.

Table 3 presents the main characteristics of all baselines and our model, showing what information each model utilizes. Specifically, we use "F" represents feature input and "S" denotes the social network input. For the modeling process, we use "UU" and "UI" denote the social information and interest information for user

Table 3 Comparison of the baselines

Model	Model Input		Model Embedding Ability			
	F	S	UU	UI	IU	II
BPR [12]	×	×	×	√	×	×
FM [10]	√	×	×	√	×	×
SocialMF [40]	×	√	√	√	×	×
TrustSVD [24]	×	√	√	√	×	×
ContextMF [41]	√	√	√	√	×	×
GraphRec [31]	×	√	√	√	×	×
PinSage [30]	√	×	×	√	√	×
NGCF [2]	×	×	×	√	√	×
DiffNet [32]	√	√	√	√	×	×
DiffNet++ [33]	√	√	√	√	√	×
MRAN	√	√	√	√	√	√
MRAN-nf	×	√	√	√	√	√
MRAN-ns	√	×	×	√	√	√
MRAN-nii	√	√	√	√	√	×

embedding learning, and use "IU" and "II" to denote the interest information and item homophily information for item embedding learning. Note that our proposed MRAN is the only one that considers item homophily information among all these models. Since our proposed model MRAN is flexible and can be reduced to a simpler version, we also construct several variants of MRAN as ablation study. We use MRAN-nf, MRAN-ns and MRAN-nii to represent the reduced versions of MRAN when removing user and item features, removing social network input and removing item homophily information.

4.1.4 Parameter setting

We implement our proposed model by using Tensorflow framework which optimizes all models with the Adam optimizer, where the batch size is 512. The embedding size d and learning rate is searched in [16, 32, 64] and [0.0005, 0.001, 0.005, 0.01, 0.05, 0.1]. We randomly initialize user/item free embedding and weight parameters with a Gaussian distribution, where the mean and standard deviation is set as 0 and 0.1 for all models. In our proposed MRAN model, we search the regularization parameter λ in [0.0001, 0.0003, 0.001, 0.003, 0.01], and find that $\lambda = 0.001, \lambda = 0.003$ reaches the best performance for Yelp dataset and Filckr dataset respectively. Moreover, we empirically set the size of the hidden layer the same as the embedding size and the activation function as Leaky ReLU. We carefully tune the parameter for all baselines to ensure the optimal performance for fair comparison.

4.2 Experiment details

4.2.1 Performance of our model and baselines (RQ1)

We first compare the Top-10 recommendation performance of MRAN and other baselines. Table 4 shows the overall rating prediction precision w.r.t. HR and NDCG with different embedding size D among the recommendation methods on Yelp and Flickr datasets, we had the following observations. First, graph neural network-based models usually obtains better performance than traditional models, including classical CF models (e.g., BPR [12], FM [10]) and social-based recommender approaches (e.g., SocialMF [40], TrustSVD [24], ContextMF [41]). This observation makes sense because traditional models failed to capture the important nonlinear relationship between users and items. However, the graph neural network-based models take higher-order social network or higher-order user-item interaction information into account. The second observation is that models with attention mechanism (e.g., GraphRec [31], DiffNet++ [33]) achieve better performance compared with other methods (e.g., PinSage [30], NGCF [2]). It is not surprising since that attention mechanism helps to better understand the implicit relationship between different nodes and aspects and improves recommendation performance. Third, since both social information and interest information play important roles in improving recommendation results. Table 4 shows the overall rating prediction accuracy w.r.t.

Table 4 Overall comparison of HR@10 and NDCG@10 with different dimension size D

Model	Yelp						Flickr					
	HR			NDCG			HR			NDCG		
	D=16	D=32	D=64	D=16	D=32	D=64	D=16	D=32	D=64	D=16	D=32	D=64
BPR [12]	0.2435	0.2616	0.2632	0.1468	0.1573	0.1554	0.0773	0.0812	0.0795	0.0611	0.0652	0.0628
FM [10]	0.2768	0.2835	0.2825	0.1698	0.1720	0.1717	0.1115	0.1212	0.1233	0.0872	0.0968	0.0954
SocialMF [40]	0.2571	0.2709	0.2785	0.1655	0.1695	0.1677	0.1001	0.1056	0.1174	0.0862	0.0910	0.0964
TrustSVD [24]	0.2826	0.2854	0.2939	0.1683	0.1710	0.1749	0.1352	0.1341	0.1404	0.1056	0.1039	0.1083
ContextMF [41]	0.2985	0.3011	0.3043	0.1758	0.1808	0.1818	0.1405	0.1382	0.1433	0.1085	0.1079	0.1102
GraphRec [31]	0.2873	0.2910	0.2912	0.1663	0.1677	0.1812	0.1195	0.1211	0.1231	0.0910	0.0924	0.0930
PinSage [30]	0.2944	0.2966	0.3049	0.1753	0.1786	0.1855	0.1192	0.1234	0.1257	0.0937	0.0986	0.0998
NGCF [2]	0.3050	0.3068	0.3042	0.1826	0.1844	0.1882	0.1110	0.1150	0.1189	0.0880	0.0895	0.0945
DiffNet [32]	0.3293	0.3437	0.3461	0.1982	0.2095	0.2118	0.1476	0.1588	0.1657	0.1121	0.1242	0.1271
DiffNet++ [33]	0.3406	0.3552	0.3694	0.2070	0.2158	0.2263	0.1562	0.1678	0.1832	0.1213	0.1286	0.1420
MRAN	0.3452	0.3628	0.3837	0.2110	0.2227	0.2382	0.1623	0.1746	0.1970	0.1273	0.1361	0.1539
MRAN-nf	0.3421	0.3571	0.3764	0.2095	0.2219	0.2344	0.1536	0.1714	0.1890	0.1211	0.1345	0.1476
MRAN-ns	0.3432	0.3610	0.3802	0.2076	0.2225	0.2366	0.1608	0.1733	0.1950	0.1255	0.1353	0.1522

HR and NDCG of the recommended methods with different embedded sizes D on Yelp and Flickr datasets. It can be found from the table that the larger the D is, the larger the MRAN value is. Traditional models cannot capture the important nonlinear relationship between users and items. The graph neural network model considers the interaction information of high-order social networks or high-order user-item interaction information. The larger D the model with attention mechanism, the better the performance. DiffNet++ significantly outperforms other baselines and becomes the strongest baseline model and our MRAN gives the best performance across all data sets.

In this experiment, we are interested in measuring the effectiveness of our model with different top-N values in Table 5 and the overall trend is similar to the analysis before. Since the data sparsity problem is commonly encountered in real-world scenarios. We also want to know how MRAN performs in the absence of user social information or item homogeneous information, and add the comparative experiment of two MRAN variants. Specifically, MRAN-ns achieves 0.2608HR@5 and 0.1928NDCG@5 in Yelp, whereas MRAN-nii achieves 0.2609HR@5 and 0.1940NDCG@5. Both of the MRAN variants outperform all the baselines in Yelp, and MRAN-nii is even more competitive than MRAN-ns.

The same experimental results are also reflected in Flickr dataset, which confirms that both user-user social network and item-item homogeneous network have positively contributed to our MRAN. In terms of retrieval efficiency, memory efficiency and time efficiency, it is also greatly superior to a wide range of existing retrieval models. Therefore, we can conclude that MRAN can capture high-order heterogeneous information between user-user, item-item and user-item in aggregation operations through two attention mechanisms, which improves the recommendation performance.

In the recommendation system, the algorithm should be considered from two aspects: the accuracy of the algorithm itself and the efficiency of the algorithm. The complexity, uncertainty and emergence of big data itself have also brought many new challenges to the recommendation system. The time efficiency, space efficiency and recommendation accuracy of traditional recommendation systems have encountered serious bottlenecks. Compared with algorithm accuracy, recommendation system engineering pays more attention to algorithm efficiency. In essence, recommendation systems improve the efficiency of information distribution and information acquisition.

4.2.2 Effectiveness of our attention mechanisms (RQ2)

We propose two attention mechanisms, namely (i) influence the node attention block in the influence diffusion process; And (ii) a graph attention block in the information aggregation process. To study the effect of these two different attention mechanisms, we compared MRAN with some model variants. The proposed algorithm can adapt to the changes in network topology and form the shortest path after stabilization. The consistency and accuracy of the optimal path calculated independently by each node are better guaranteed, rather than based on the calculation results of other nodes. We use AVG to represent the attention mechanism that degenerates to usage equal

Table 5 Overall comparison of HR@N and NDCG@N with different top-N values (D=64)

Model	Yelp						Flickr					
	HR			NDCG			HR			NDCG		
	N=5	N=10	N=15	N=5	N=10	N=15	N=5	N=10	N=15	N=5	N=10	N=15
BPR [12]	0.1695	0.2632	0.3252	0.1231	0.1554	0.1758	0.0651	0.0795	0.1037	0.0603	0.0628	0.0732
FM [10]	0.1855	0.2825	0.3440	0.1341	0.1717	0.1876	0.0989	0.1233	0.1473	0.0866	0.0954	0.1062
SocialMF [40]	0.1739	0.2785	0.3365	0.1324	0.1677	0.1841	0.0813	0.1174	0.1300	0.0723	0.0964	0.1061
TrustSVD [24]	0.1882	0.2939	0.3688	0.1368	0.1749	0.1981	0.1089	0.1404	0.1738	0.0978	0.1083	0.1203
ContextMF [41]	0.2985	0.3011	0.3043	0.1758	0.1808	0.1818	0.1405	0.1382	0.1433	0.1085	0.1079	0.1102
GraphRec [31]	0.1915	0.2912	0.3623	0.1279	0.1812	0.1956	0.0931	0.1231	0.1482	0.0784	0.0930	0.0992
PinSage [30]	0.2105	0.3049	0.3863	0.1539	0.1855	0.2137	0.0934	0.1257	0.1502	0.0844	0.0998	0.1046
NGCF [2]	0.1992	0.3042	0.3753	0.1450	0.1828	0.2041	0.0891	0.1189	0.1399	0.0819	0.0945	0.0998
DiffNet [32]	0.2276	0.3461	0.4217	0.1679	0.2118	0.2307	0.1178	0.1657	0.1855	0.1072	0.1271	0.1301
DiffNet++ [33]	0.2503	0.3694	0.4493	0.1841	0.2263	0.2497	0.1412	0.1832	0.2203	0.1269	0.1420	0.1544
MRAN	0.2639	0.3837	0.4611	0.1953	0.2382	0.2614	0.1540	0.1970	0.2329	0.1395	0.1539	0.1653
MRAN-nf	0.2608	0.3764	0.4479	0.1928	0.2344	0.2560	0.1464	0.1890	0.2223	0.1334	0.1476	0.1583
MRAN-ns	0.2609	0.3802	0.4566	0.1940	0.2366	0.2595	0.1504	0.1950	0.2318	0.1374	0.1522	0.1642

attention weight without any learning process. Self-attention model can establish long-distance dependence within a sequence, which can also be achieved through the fully connected neural network, but the problem is that the number of connected edges of a fully connected network is fixed, so it can not deal with sequences with variable length. While the self-attention model can dynamically generate weights of different connections, the number of weights generated and the size of weights are variable. When a longer sequence is input, only more connected edges need to be generated. We have conducted ablation studies, and the results of different attention modeling combinations are shown in Table 6. In particular, we run each sub-module of MRAN with/without the corresponding attention mechanism (i.e., ATT or AVG), and find that it achieves the best performance when combining the node level attention and the graph level attention. The experimental results show confirms that both nodes and graph attention blocks can improve the performance of our model by distinguishing importance weights. Compared with some model variants, MRAN has fast convergence speed and is suitable for large networks. Other algorithms are complex and require large storage space. Since the calculation is conducted after diffusion, the calculation results and process do not affect the diffusion and do not depend on the calculation results of other nodes.

In terms of time efficiency, it is much better than a wide range of existing retrieval models. MRAN can capture advanced heterogeneous information among user-user, item-item and user-item in aggregation operations through two focus mechanisms, thus improving the recommendation performance. In the recommendation system, the algorithm should be considered from two aspects: the accuracy of the algorithm itself and the efficiency of the algorithm. The complexity, uncertainty and emergence of big data itself have also brought many new challenges to time efficiency. Therefore, the improvement in time efficiency confirms that both user-user social networks and item-item homogeneous networks have made positive contributions to our MRAN performance. Compared with algorithm accuracy, MRAN recommendation system engineering pays more attention to algorithm efficiency. In essence, recommendation system improves the efficiency of information distribution and information acquisition.

4.2.3 Effectiveness of diffusion depth K (RQ3)

Now, we analyze how sensitive our model is to the diffusion depth K, and which depth value produces the best recommendation result. We report the experiment results of MRAN with different K values for both datasets in Table 7. Note that, many related studies have achieved the best performance when K=2, and the performance drops when the depth of graph continues to increase. The "improve" column shows the performance change compared to the setting of MRAN, i.e., K=2. We find that the performance improves rapidly when K increases from 0 to 1, the performance will improve rapidly, and when the diffusion depth continues to increase, the performance will still improved slightly. We conclude that the application of our new two attention mechanisms alleviates the problem of over-smoothing in the training of graph neural network training and preserves the difference of hidden layer representation of each node.

Table 6 HR@10 and NDCG@10 performance with different attentional variants (D = 64, K=2)

Graph attention	Node attention	Yelp				Flickr			
		HR	Improve (%)	NDCG	Improve (%)	HR	Improve (%)	NDCG	Improve (%)
AVG	AVG	0.3738	–	0.2321	–	0.1808	–	0.1407	–
AVG	ATT	0.3744	+0.16	0.2334	+0.56	0.1809	+0.06	0.1411	+0.28
ATT	AVG	0.3809	+1.90	0.2369	+2.07	0.1949	+7.80	0.1513	+7.53
ATT	ATT	0.3837	+2.65	0.2382	+2.63	0.1970	+8.96	0.1539	+9.38

Table 7 HR@10 and NDCG@10 performance with different diffusion depth K (D= 64)

Depth K	Yelp				Flickr			
	HR	Improve (%)	NDCG	Improve (%)	HR	Improve (%)	NDCG	Improve (%)
K=0	0.2362	−31.40	0.1554	−34.76	0.0795	−59.64	0.0628	−40.81
K=1	0.3748	−2.32	0.2331	−2.14	0.1808	−8.22	0.1418	−7.86
K=2	0.3837	–	0.2382	–	0.1970	–	0.1539	–
K=3	0.3883	+1.20	0.2409	+1.13	0.2031	+3.10	0.1570	+2.01%
K=4	0.3918	+2.11	0.2441	+2.48	0.2073	+5.23	0.1622	+5.39%

5 Conclusions

In this paper, we propose a new framework MRAN, which can effectively recommend relevant items to users in the social recommendation scenarios. To alleviate the problem of data sparsity problem, we introduce homogeneous information between items as supplementary information. Especially relatively complex neural networks, such as R-CNN. Our idea is to understand the network by deleting some networks and studying its performance. In order to overcome over-smoothing, we build a high-order influence diffusion attention model on the three kinds of graphs to capture important embeddings. The study of ablation is very important for the study of deep learning. Understanding the causality in the system is the most direct way to generate reliable knowledge (the goal of any research). Ablation is a very labor-saving way to study causality. We compared the performance of the model with ten state-of-the-art baselines. The experimental results show that our model achieves up to 5.7% and 6.8% improvement at HR@10 and NDCG@10 compare with the best baseline in both datasets. In addition, our MRAN alleviates the probem of over-smoothing. Increasing the depth K to more than 2 can further improve the performance of the model.

Social recommendation is a challenging research problem.Based on this paper, much work can be done in the future. From the perspective of multi-source social networks, users are not only active in one social network, but also active in multiple social networks. How to combine multiple social networks to learn users is very interesting and promising. From the perspective of multi-type behavior, additional information can be added to the model, such as comment information, comment time, location information, etc.

6 Discussion

Our proposed model alleviates the data sparsity problem by introducing the homogeneous information between items, and constructs a high-order impact diffusion graph attention model using the social information between users, the homogeneous information between items, and the interactive information between users and items. It has designed two kinds of attention mechanisms, which are applied to the diffusion and aggregation levels respectively, so that it can distinguish the importance

weight when building users and embedding items. The inadequacies of the research on social recommended multi relational attention network can be elaborated and analyzed from two aspects: objective factors and subjective factors. Our MRAN research methods have limitations even if they are not affected by external factors. In this study, we designed and evaluated the multi relational social network recommendation algorithms based on collaborative filtering, machine learning and data mining technologies respectively. Finally, the model algorithm is validated with real online social network data to analyze the accuracy and robustness of multi social relationship recommendation mechanism.

In the process of using MRAN, because of various objective factors, these research methods are more likely to lead to biased results. As far as the subjective factors are concerned, the deficiencies of the multi relational attention network recommended by the society due to the subjective reasons of the researchers themselves (whether directly or indirectly) should be mentioned in this part, and specific solutions should be proposed to solve or reduce the research limitations in this area. In addition, the multi relationship attention network recommended by the society is not scientific and typical enough to select samples in the survey stage, and the sampling method is not scientific; The sample size is insufficient to represent the overall situation of the research object.

But in the weaknesses, MRAN cannot find enough samples, which will lead to the accuracy of the final results. MRAN based recommendation systems mainly include collaborative filtering recommendation systems, content-based recommendation systems, etc. Some typical recommendation methods have been effectively applied in practical applications. However, there are some problems with MRAN based recommendation methods, which fail to extract more useful information from social networks, and then conduct entity recommendation from multiple dimensions, which will inevitably result in low accuracy of recommendation results. In view of the shortcomings of the research on MRAN based recommendation systems, we should focus on the research and analysis of user relationship strength in social networks, and propose a multi-dimensional comprehensive recommendation method based on user relationship strength in social networks.

Acknowledgements This work is partly supported by a grant from the Innovative Research Foundation of Ship General Performance (14422102).

Author Contributions YF: Conceptualization, Methodology, Software, Writing Original draft preparation. XX: Investigation, Experiment, Writing Reviewing and Editing. TZ: Supervision, Writing Reviewing and Editing. All authors reviewed the manuscript.

Funding This work is partly supported by a grant from the Innovative Research Foundation of Ship General Performance (14422102).

Data Availibility Not applicable.

Declarations

Conflict of interest The authors declare that they have no competing interests.

Ethical Approval Not applicable.

Consent for publication Not applicable.

References

1. He X, Liao L, Zhang H, Nie L, Hu X, Chua T-S (2017) Neural collaborative filtering. https://doi.org/10.1145/3038912.3052569
2. Wang X, He X, Wang M, Feng F, Chua T-S (2019) Neural graph collaborative filtering, pp 165–174. https://doi.org/10.1145/3331184.3331267
3. Zhang J, Shi X, Zhao S, King I (2019) Star-gcn: stacked and reconstructed graph convolutional networks for recommender systems, pp 4264–4270. https://doi.org/10.24963/ijcai.2019/592
4. Su X, Khoshgoftaar T (2009) A survey of collaborative filtering techniques. Adv Artif Intell. https://doi.org/10.1155/2009/421425
5. Herlocker J, Konstan J, Borchers A, Riedl J (1999) An algorithmic framework for performing collaborative filtering, pp 230–237. https://doi.org/10.1145/312624.312682
6. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of ACM World Wide Web Conference, vol 1. https://doi.org/10.1145/371920.372071
7. Do M-P, Nguyen D, Nguyen L (2010) Model-based approach for collaborative filtering
8. Jenatton R, Roux N, Bordes A, Obozinski G (2012) A latent factor model for highly multi-relational data. vol 4
9. Lü L, Medo M, Yeung CH, Zhang Y-C, Zhang Z-K, Zhou T (2012) Recommender systems. Phys Rep 519(1):1–49. https://doi.org/10.1016/j.physrep.2012.02.006 (**Recommender Systems**)
10. Rendle S (2010) Factorization machines. In: 2010 IEEE International Conference on Data Mining, pp 995–1000. https://doi.org/10.1109/ICDM.2010.127
11. Rendle S (2012) Factorization machines with libfm. ACM Trans Intell Syst Technol (TIST). https://doi.org/10.1145/2168752.2168771
12. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) Bpr: bayesian personalized ranking from implicit feedback, pp 452–461
13. Zhou X, He J, Huang G, Zhang Y (2014) Svd-based incremental approaches for recommender systems. J Comput Syst Sci. https://doi.org/10.1016/j.jcss.2014.11.016
14. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. Computer 42:30–37
15. Sarwar B, Karypis G, Konstan J, Riedl J (2000) Application of dimensionality reduction in recommender system–a case study
16. Fan M (2012) Non-negative matrix factorization and clustering methods application research in personalized recommendation system
17. Wang X, Lu W, Ester M, Wang C, Chen C (2016) Social recommendation with strong and weak ties, pp 5–14. https://doi.org/10.1145/2983323.2983701
18. Tang J, Hu X, Liu H (2013) Social recommendation: a review. Soc Netw Anal Min 3:1113–1133. https://doi.org/10.1007/s13278-013-0141-9
19. Ma H, Yang H, Lyu M, King I (2008) Sorec: Social recommendation using probabilistic matrix factorization, pp 931–940. https://doi.org/10.1145/1458082.1458205
20. Yang B, Lei Y, Liu J, Li W (2016) Social collaborative filtering by trust. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/TPAMI.2016.2605085
21. Fang H, Bao Y, Zhang J (2014) Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation. Procee Nat Conf Artif Intell 1:30–36
22. Tang J, Hu X, Gao H, Liu H (2013) Exploiting local and global social context for recommendation. In: IJCAI International Joint Conference on Artificial Intelligence, pp 2712–2718
23. Koren Y (2008) Factorization meets the neighborhood: a multifaceted collaborative filtering model, pp 426–434. https://doi.org/10.1145/1401890.1401944
24. Guo G, Zhang J, Yorke-Smith N (2015) Trustsvd: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. Procee Twenty-Ninth AAAI Conf Artif Intell 29:123–129. https://doi.org/10.1609/aaai.v29i1.9153
25. Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun M (2020) Graph neural networks: a review of methods and applications. AI Open 1:57–81. https://doi.org/10.1016/j.aiopen.2021.01.001

26. Kipf T, Welling M (2016) Semi-supervised classification with graph convolutional networks
27. Wu N, Wang C (2022) Ensemble graph attention networks. Trans Mach Learn Artif Intell 10, 29–41. https://doi.org/10.14738/tmlai.103.12399
28. Hamilton W, Ying R, Leskovec J (2017) Inductive representation learning on large graphs
29. Berg R, Kipf T, Welling M (2017) Graph convolutional matrix completion
30. Ying R, He R, Chen K, Eksombatchai P, Hamilton W, Leskovec J (2018). Graph convolutional neural networks for web-scale recommender systems. https://doi.org/10.1145/3219819.3219890
31. Fan W, Ma Y, Li Q, He Y, Zhao E, Tang J, Yin D (2019) Graph neural networks for social recommendation, pp 417–426. https://doi.org/10.1145/3308558.3313488
32. Wu L, Sun P, Fu Y, Hong R, Wang X, Wang M (2019) A neural influence diffusion model for social recommendation, pp 235–244. https://doi.org/10.1145/3331184.3331214
33. Wu L, Li J, Sun P, Hong R, Ge Y, Wang M (2020) Diffnet++: a neural influence and interest diffusion network for social recommendation. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2020.3048414
34. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. vol 1. https://doi.org/10.1145/371920.372071
35. Wu Q, Zhang H, Gao X, He P, Weng P, Gao H, Chen G (2019) Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. https://doi.org/10.1145/3308558.3313442
36. Zhao T, McAuley J, King I (2014) Leveraging social connections to improve personalized ranking for collaborative filtering, pp 261–270. https://doi.org/10.1145/2661829.2661998
37. Chen J, Zhang H, He X, Nie L, Liu W, Chua T-S (2017) Attentive collaborative filtering: multimedia recommendation with item- and component-level attention, pp 335–344. https://doi.org/10.1145/3077136.3080797
38. You D, Vo N, Lee K, Liu Q (2020) Attributed multi-relational attention network for fact-checking url recommendation
39. He X, Liao L, Zhang H, Nie L, Hu X, Chua T-S (2017) Neural collaborative filtering, pp 173–182. https://doi.org/10.1145/3038912.3052569
40. Jamali M, Ester M (2010) A matrix factorization technique with trust propagation for recommendation in social networks, pp 135–142. https://doi.org/10.1145/1864708.1864736
41. Jiang M, Cui P, Wang F, Zhu W, Yang S (2014) Scalable recommendation with social contextual information. Knowl Data Eng IEEE Trans 26:2789–2802. https://doi.org/10.1109/TKDE.2014.2300487