Computer Science and Information Systems 20(2):609–629 https://doi.org/10.2298/CSIS220705012W

Content-only attention Network for Social Recommendation

Bin Wu[1], Tao Zhang[2] and Yeh-Cheng Chen[3]

[1] School of Internet of Things Engineering, Jiangnan University,
Wuxi, 214122, China
wubin@jiangnan.edu.cn
[2] China Ship Scientific Research Center,
Wuxi 214122, China
taozhang@jiangnan.edu.cn
[3] Department of computer science, University of California,
Davis, CA, USA
ycch@ucdavis.edu

Abstract. With the rapid growth of social Internet technology, social recommender has emerged as a major research hotspot in the recommendation systems. However, traditional graph neural networks does not consider the impact of noise generated by long-distance social relations on recommendation performance. In this work, a content-only multi-relational attention network (CMAN) is proposed for social recommendation. The proposed model owns the following advantages: (i) the comprehensive trust based on the historical interaction records of users and items are integrated into the recursive social dynamic modeling to obtain the comprehensive trust of different users; (ii) social trust information is captured based on the attention network mechanism, so as to solve the problem of weight distribution in the same level domain; (iii) two levels of attention mechanisms are merged into a unified framework to enhance each other. Experiments conducted on two representative datasets demonstrate that the proposed algorithm outperforms previous methods substantially.

Keywords: recommender system, social network, content-only multi-relational attention network.

1. Introduction

With the rapid development of computer technology and the maturity of Internet economy, recommendation systems have become a hot topic for researchers. The traditional recommendation systems mainly face two problems: one is the sparsity problem with the users' rating data. In practical applications, the number of users and items is vast, but the historical behavioral records between users and items are rare. When the number of users and items in the recommendation system increases, with the extremely sparse historical behavioral matrix, the user preferences cannot be accurately learned. Therefore, the accuracy of recommendation is significantly reduced. The other problem is the cold start problem. For users or items newly added to the recommendation system, there is no historical behavioral record, so it is difficult to provide fine personalized recommendations.

How to get effective information from massive amounts of data is a huge challenge for ordinary users. Researchers have proposed to use the traditional matrix decomposition methods to improve the neural network recommendation models in recent years. For example, some works use deep learning schemes to model the deeper data relationship between the user eigenvector and the item eigenvector, which proactively provides them with products that meet their potential preferences [1]. In addition, the graph-based neural network is used to depict the user preference generation process, and the recommendation system can help them quickly find satisfactory information in large amounts of information, and for merchants, it can not only help to decide to push to specific users, but also can increase user loyalty through more satisfactory services [2, 3]. Although the recommendation algorithm in various applications have achieved great success, however, sparse data problem is still one of the important bottlenecks affecting algorithm performance recommendation algorithm is usually based on the user's historical data to model user preferences.

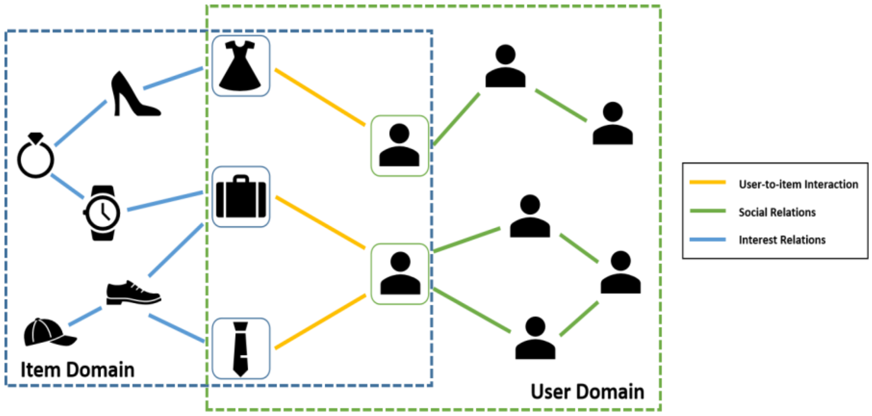

Fig. 1. The graph structure in social recommendation, which includes three graphs: the user-user graph (left part), the user-item graph (middle part) and the item-item graph (right part).

This paper mainly concern the recommendation algorithms combined with social information. In order to improve the quality of recommendation algorithms, especially in dealing with the problem of sparse data, a content-only multi-relational attention network (CMAN) is proposed, which jointly models the three graph structures for social recommendation, including the user-user graph, the item-item graph and the user-item graph (Fig. 1.). The contribution points of this paper mainly include:

First, a novel framework called Content-based Multi-Relational Attention Network (CMAN) is proposed for social recommendation, which jointly captures the influence and interest diffusion in multi-relational context space;

Second, this paper try to add homogeneous information between items to solve the data sparsity problem and high-order influence diffusion process is further exploited to extract multi-relational contexts;

Third, a two-level attention mechanism is proposed to comprehensively consider the influence of score similarity and implicit trust relationship on the trust between each group of users, and obtain a more accurate trust beard and recommendation model.

The remainder of this paper is organized as follows. In Section 2, the social recommendation problem is defined to be solved and review some related works. The proposed framework is formally described in Section 3. Experimental results and discussion are presented in Section 4. Finally, conclusion and future research directions are given in Section 5.

2. Problem Definition and Related Work

This part mainly introduces two classical algorithms directly related to this paper. First, the problem of social recommendation is formally defined and described, and then two existing related work are reviewed respectively, including the recommendation algorithm based on probability matrix decomposition.

Table 1. Variables used of this paper

Variables	Definitions
p_a	Latent embedding of user a
q_i	Latent embedding of item i
x_a	Real-valued attributes of a
y_i	Real-valued attributes of i
d	Length of the embedding vector
S	User-user social network
S_a	The set of social friends that a follows
R	User-item interaction matrix
$R_U(i)$	The set of users that interacted with item i
$R_I(a)$	The set of items that user a interacts with
F	Item-item influence network
F_i	The set of item friends that item i connects
τ	A fixed threshold who links both items in F
r_{ai}	The observed preference of user a in item i
$\hat{r}_{ai}$	The predicted preference of user a in item i
$\oplus$	Concatenation operator
G_S	User-user social graph
G_I	User-item interest graph
G_F	Item-item influence graph

2.1. Problem Definition

This paper assumes that $U = \{u_1, u_2, \ldots, u_n\}$ denotes the sets of users, and $V = \{v_1, v_2, \ldots, v_m\}$ is the sets of items, where *N* and *M* are numbers of users and items. A user-item interaction matrix $R \in \mathbb{R}^{n \times m}$ is used to depict users' implicit preference and interests to items (In proposed model, rating values range from 1 to 5). This paper assume that $r_{ai} = 1$ if u_a is interested in v_i, and $r_{ai} = 0$ if u_a do not rate v_i. In addition, this paper uses $R_U(i)$ and $R_I(u)$ *K*-dimensional potential vector of user u and product, respectively. Moreover, the user-user directed graph is denoted as $G = [U, S \in \mathbb{R}^{n*n}]$, where U denotes the set of users *u* and *S* is the connections relationship between users of a social network. This paper denotes $s_{ab} = 1$ if u_a trusts u_b and zero otherwise. In addition, this paper uses S_a to denote the set of users, the purpose of the probability matrix factorization is to learn these vectors from the score, i.e., $S_a = [b \mid s_{ab} = 1]$. Moreover, this paper introduces an embedding vector $x_a \in \mathbb{R}^d$ to predict how the user *u* scores on the unrated items for u_a and an embedding vector $y_i \in \mathbb{R}^d$ to predict how the user *v* scores on the unrated items for v_i, where d denotes the dimension of embedding vector, given the evaluation matrix R and the trust relationship T, it can be shown that recommendation algorithms combined with social information can effectively alleviate the problem of sparse data. The used notations of this paper are summarized in Table 1. The social recommendation problem can be described as:

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

Output: $\hat{R} = f(U, V, R, S, X, Y)$, where $\hat{R} \in \mathbb{R}^{n*m}$ denotes the unobserved interactions between users and items.

2.2. Related Work

Classical CF Recommendation Models. There are two main types of collaborative filtering methods [4], i.e., 1) memory-based collaborative filtering, which compute the similarity between users and items through users' rating history, and then new items are recommended for users based on the similarity. Typical examples of this approach are neighborhood-based CF and item-based/user-based top-N recommendations [5, 6]; and 2) model-based collaborative filtering models, which are developed to predict users' rating of unrated items [7]. The focus of classical CF recommendation models is on how to integrate social information and evaluation information more effectively.

Most social recommendation algorithms focus on solving two problems: how to effectively integrate social information into recommendation algorithms? How to estimate the trust between users to improve the algorithm accuracy? For the first problem, many work attempts to model social information from different perspectives [8]. For example, Jamah has proposed a TrustWalker algorithm for random walk in scoring networks and social networks. The algorithm obtains the results by running midstream in the historical data without pretraining, but when the data volume is relatively large, the inquiry time is often too large [9]. Assuming that the user-item rating matrix is $R \in \mathbb{R}^{n\times m}$ (The purpose of the probability matrix factorization is to learn these vectors from the score), the matrix factorization algorithm usually learns two low-rank matrices $U \in \mathbb{R}^{n*k}$ and $V \in \mathbb{R}^{m*k}$, therefore, it can be formulated as:

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

where $\hat{R}$ denotes the approximation matrix of R, U denotes the user's latent feature matrix, and V represents the item's latent feature matrix. Generally speaking, the rank k of two characteristic matrices U and V is very small, so the above matrix factorization is also called low rank matrix factorization. After achieving U and V, the user a that corresponds to item i can be predicted according to the following criterion [10-12]:

$$\hat{r}_{\mathrm{ai}} = U_{\mathrm{a}} V_{\mathrm{i}}^T \tag{2}$$

where u_a is the a-th row of the user embedding matrix U, which is a normal distribution as the mean. Similarly, v_i denotes the latent embedding of item i in i-th row of item embedding matrix V. To get the optimal matrix representation $U \in \mathbb{R}^{n*k}$ and $V \in \mathbb{R}^{m*k}$, additional L2-norm regularization terms [8] are used to solve this problem:

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

where the first term is the approximation error of matrix decomposition, the second and third terms are regularization terms, and μ is the regularization coefficient. Higher accuracy was achieved with their algorithm compared to previous work.

While 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 recommendation methods, and the implied similarity of the trust relationship is considered.

Matrix Factorization-Based Social Recommendation Models. Social-based recommendation has gradually become an indispensable part of recommendation algorithms. The focus of social recommendation algorithm is on how to integrate social information and evaluation information more effectively [17-19]. Traditional recommendation systems assume that users are independent and identically distributed, which subconsciously ignores the social interaction between users. However, these algorithms always face the problem of very sparse user history data, resulting in decreased recommendation quality. To solve this problem, it is effective to introduce auxiliary data or mine more laws in the data into the algorithm.

The first category is based on the shared representation of the user feature matrix, which means that the dynamic combination with the user and product similarity to improve the algorithm to process sparse data. By assuming the user evaluation is determined by personal preferences and friend influence, the objective function of SoRec [19] can be described as:

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

where $g(x) = \frac{1}{1+\exp(-x)}$, $z_b \in \mathbb{R}^d$ denotes 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 the SoRec algorithm, assuming that users share the same preference vector in the evaluation network and social network, and using the probability matrix decomposition to finally obtain the recommendation structure considering the user's friend factors. Higher accuracy was achieved with their algorithm compared to previous work. Followed it, Fang et al. [21] proposed RSTE, a recommendation algorithm that integrates evaluation information and social networks. In addition, Tang et al. [22] proposed a social recommendation model LOCABAL, The algorithm assumes that user evaluation is determined by personal preferences and friend influence, and has a balance.

Guo et al. [23] introduced the method of trust communication into the recommendation algorithm (SocialMF) [24]. This model spreads the trust relationship by restricting the average preferences of users and their friends that are similar, so as to get more accurate results, therefore, our objective function could be rewritten as:

$$\sum_{r_{ai} \neq 0} \left(r_{ai} - \mu - b_a - b_i - \left(u_a + |R(a)|^{-\frac{1}{2}} \sum_{p \in R(a)} I_p + |S(a)|^{-\frac{1}{2}} \sum_{k \in S(a)} W_k \right)^T v_i \right)^2 \quad (5)$$

where μ denotes the mean value of global ratings, b_a denotes the difference between the average rating of u_a and μ, b_i denotes the difference between the average rating of v_i and μ, I_p indicates the influence factor of v_p on user feature vector, W_k indicates the influence factor of u_k on user feature vector, and $R(a)$ denotes the collection of items rated by u_a, $S(a)$ denotes the social association user set of u_a.

Graph Neural Network-Based Social Recommendation Models. Graph Neural Networks (GNNs), as a generalization of deep neural networks on graph data[25], can better extract and represent data characteristics in graph field. GNN has developed a lot of different forms, such as GCN [26], GAT[27], GraphSAGE [28] and so on. In this regard, many works considered the impact of user score similarity on trust strength, using social networks as a regular constraint term to learn user preferences, and obtained more accurate results. Guo Lei et al. used the potential vectors obtained from the probability matrix decomposition of the scoring matrix to calculate the similarity between users and friends, which improves the accuracy of the algorithm. They also proposed to make full use of the relationship between objects to improve the recommendation accuracy.

The development of graph neural networks assumes that each friend has the same impact on the user. In recent years, the related research based on graph neural networks in recommender systems has attracted more and more attention of scholars. GC-MC [29] considered the different cases of users as trusted people and trusted people, and calculates the trust similarity of the two cases respectively, so as to restrict the learning of user preference vectors, and then weight the user vectors in these two cases and influence the scoring results.

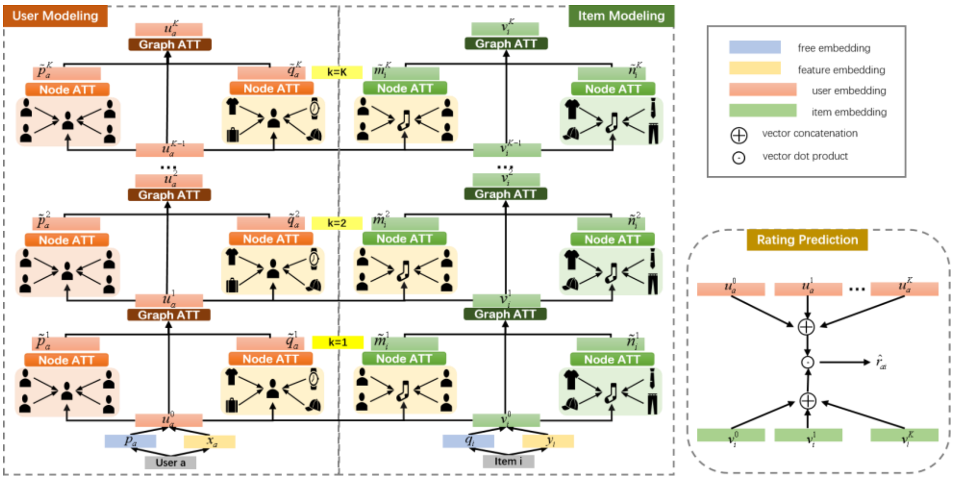

Fig. 2. The overall architecture of the proposed CMAN model.

Considered the trust strength between users from the perspective of scoring and trust data, the latent relationship (cooperative signal) is proposed to improve user's ability measure function and reliability measure function, respectively, and obtained the estimated users' preferences by combining them with the score similarity function. A more accurate recommendation model is also constructed. Similarly, PinSage [30] is a random-walk-based GCN that uses probabilistic matrix factorization to obtain the potential vectors of the user as trusted and trusted person, and use this to calculate the similarity of the user as trusted and trusted, respectively.

The above methods directly use GNNs over the user-item interaction data without considering cold start and sparse information. For comparison, GraphRec [31] proposed a unified framework for jointly modeling user/item, In fact, the trust relationship among users is influenced by a variety of factors, some by similar hobbies, some by the same social circle, and some just out of politeness. A simple binary trust network does not reflect the size of the influence between users, nor does it fully exploit the implied user preferences information in social networks, which leads to the second question, namely, how to estimate the trust between users [27]. Moreover, DiffNet [32] used the potential vectors obtained from the probability matrix decomposition of the scoring matrix to calculate the similarity between users and friends, which improves the accuracy of the algorithm. They also proposed to make full use of the relationship between objects to improve the recommendation accuracy. DiffNet++ [33] further established the relationship between user scoring preferences and social preferences through the linear mapping method to aggregate the different order neighbors' feature vectors for each channel.

This paper uses probabilistic matrix factorization to obtain the potential vectors of the user as trusted and trusted person, and uses this to calculate the similarity of the user as trusted and trusted, respectively. Besides that, this paper comprehensively considered the effect of score and trust similarity on the strength of trust among users.

3. The Proposed Model

In this section, we will first give an overview about the architecture of our proposed model CMAN, and then detail each component of the model. Finally, we give the training process of CMAN.

3.1. An Overview of the Proposed Model

The overall architecture of the proposed model is shown in **Fig. 2**. Generally speaking, the proposed CMAN 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 to the previous work of social recommendation, our trust similarity calculation does not depend on the common friend collection or the number of user friends between the users. Stable and reliable calculation results are obtained even when the trust data is sparse or the set of common friends is zero. Moreover, multi-layer GNNs with two different attention mechanism captures the multi-order influence information at different scales. Meanwhile, by mapping user social behavior to low-dimensional subspaces, it shows that user vectors not only contain direct associations between users, but also imply indirect connections between users. Therefore, a more accurate trust-most model is obtained. Last, we comprehensively considered the effect of score and trust similarity on the strength of trust among users.

3.2. User Modeling

This paper defines $P \in \mathbb{R}^{n*d}$ as the embedding matrices of users, where d is the embedding size and p_a denotes the free latent embedding for user a. By feeding p_a and the associated feature vector x_a into the fusion layer, the initial latent preference of user a is defined as:

$$u_a^0 = \sigma\left(\mathrm{W}_1 \times \left[\mathrm{p}_a, \mathrm{x}_a\right]\right) \tag{6}$$

where $\sigma(\cdot)$ denotes the activation function, W_1 denotes a trainable transformation matrix, based on this, the information in the scoring and trust data is also considered in close proximity to our work.

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. Two aspects of information are combined, and the implied similarity of the trust relationship is considered, thus obtaining a more accurate trust. This paper defines $\tilde{p}_a^k$ as the aggregated embedding of influence diffusion from the trusted social neighbors of G_S and $\tilde{q}_a^k$ as the embedding of interest diffusion from the interested item neighbors of G_I at the k-th layer. Finally, we obtain a recommendation method model that comprehensively considers the scoring and trust similarity, which can be defined as:

$$\mathrm{u}_a^k = \left(\gamma_{a1}^k \mathrm{u}_a^{\mathrm{k\text{-}1}} + \gamma_{a2}^k \mathrm{p}_a^{\sim\mathrm{k}} + \gamma_{a3}^k \mathrm{q}_a^{\sim\mathrm{k}} \right), \tag{7}$$

$$\mathrm{p}_a^{\sim k} = \sum_{b \in S_a} \alpha_{ab}^k \mathrm{u}_b^{k-1}, \tag{8}$$

$$\mathrm{q}_a^{\sim k} = \sum_{i \in R_I(a)} \beta_{ai}^k \mathrm{v}_i^{k-1}, \tag{9}$$

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. In social networks, α_{ab}^k denotes 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.

The user's potential preference information is usually implied in the scoring matrix, which is the main data adopted by the recommendation algorithm. Users' social relations usually can only obtain binary data, but not all friends have the same impact on users.The basic goal of the social recommendation algorithm is to predict how the user u scores on the unrated items *i*, given the evaluation matrix *R* and the trust relationship T. Specifically, the node-level weights, i.e., the social influence strengths α_{ab}^k and the interest influence strengths β_{ai}^k, concretely show that recommendation algorithms combined with social information can effectively alleviate the problem of sparse data. At present, how to efficiently mine the hidden user preference information in social relationships is the focus of social recommendation algorithm. Traditional collaborative filtering recommendation algorithms usually use users only user-history evaluation data to model them, and then predict users 'future evaluation and selection, including memory-based and model-based 3' women. Therefore, it is necessary for each user to build personalized weight. Thus, this step can quickly generate the recommended results, and cosine similarity functions α_{ab}^k are defined:

$$\alpha_{ab}^k = \frac{u_a^{k-1} \cdot u_b^{k-1}}{|u_a^{k-1}| \times |u_b^{k-1}|}, \tag{10}$$

$$\alpha_{ab}^{k} = soft\max\left(\alpha_{ab}^{k}\right) = \frac{\exp\left(\alpha_{ab}^{k}\right)}{\sum_{b\in S_a}\exp\left(\alpha_{ab}^{k}\right)}, \tag{11}$$

Where we use a softmax function and each value is quantified into (0,1). Similarly, we calculate the interest influence score β_{ai}^{k} by corresponding two vector product and item embedding as input. Then, the conditional probability of the scoring matrix R in the given item i is defined as:

$$\beta_{ai}^{k} = \sigma\left(u_a^{k-1} \odot v_i^{k-1}\right), \tag{12}$$

$$\beta_{ai}^{k} = soft\max\left(\beta_{ai}^{k}\right) = \frac{\exp\left(\beta_{ai}^{k}\right)}{\sum_{i\in R_I(a)}\exp\left(\beta_{ai}^{k}\right)}, \tag{13}$$

where σ is the sigmoid function. Meanwhile, in order to prevent overfitting, the algorithm assumes that the potential vectors of the user and the product satisfy the Gaussian distribution. Inspired by the scheme of tackling the node attention layer, we can model the graph attention weights of $\gamma_{al}^{k}(l = 1,2,3)$ as:

$$\gamma_{a1}^{k} = MLP_1^{k}\left(u_a^{k-1}\right), \tag{14}$$

$$\gamma_{a2}^{k} = MLP_2^{k}([u_a^{k-1}, p_a^{\sim k}]), \tag{15}$$

$$\gamma_{a3}^{k} = MLP_3^{k}([u_a^{k-1}, q_a^{\sim k}]), \tag{16}$$

where MultiLayer Perceptrons (MLPs) approaches are used to learn the graph attention weights at the $(k-1)$-th layer (u_a^{k-1}) and node attention representations at the k-th layer($p_a^{\sim k}$ and $q_a^{\sim k}$). Suppose that U- -RKXN and V6RK money represent the potential matrix of user and product and product, respectively, where U " and K represent the K-dimensional potential vector of user U and product, respectively. They model both from the perspective of trusted and scored close neighbor sets, however, these two sets are not equal, using only one aspect of the information for close neighbors present in only one of the sets. In addition, considering $\gamma_{a1}^{k} + \gamma_{a2}^{k} + \gamma_{a3}^{k} = 1$, if the value of γ_{a2}^{k} is bigger 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.

3.3. Item Modeling

$Q \in$ ¡ ^{m*d} indicates the free embedding matrices of items, where d denotes the embedding size and q_i represents the free latent embedding for item *i*. By incorporating q_i and the associated feature vector y_i into the fusion layer, we can get the initial item embedding:

$$v_i^0 = \sigma(W_2 \times [q_i, y_i,]) , \tag{17}$$

In this paper, we attempt to construct an item-item influence network F . In order to more effectively model the social relationship between users, and fully explore the influence of social interaction and friends on users' preferences, a lot of work has made a beneficial exploration of this [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. This paper defines the item implicit network as the graph $G_F = [V, F \in \mathbb{R}^{m*m}]$, where V denotes the set of items and F denotes the connections between the two related items.

Based on the assumption that users and friends have similar preferences, for each item i , what is needed is to aggregate user-space information from the set of users who have interacted with item i , denoted as $R_U(i)$, including the recommendation algorithm based on probability matrix decomposition 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 as:

$$v_i^k = (\eta_{i1}^k v_i^{k-1} + \eta_{i2}^k\, m_i^{\sim k} + \eta_{i3}^k\, n_i^{\sim k}) , \tag{18}$$

$$m_i^{\sim k} = \sum_{j \in F_i} \mu_{ij}^k v_j^{k-1} , \tag{19}$$

$$n_i^{\sim k} = \sum_{a \in R_U(i)} v_{ia}^k u_a^{k-1} , \tag{20}$$

where $R_U(i) = [a \mid r_{ia} = 1]$ denotes the user set that rates item i , $F_i = [\, j \mid f_{ij} = 1]$ denotes the item set that is related to item i , $\tilde{m}_i^k$ is the item i 's aggregated embedding in the item-item influence graph G_F , $\tilde{n}_i^k$ represents the item i 's aggregated embedding in the user-item interest graph G_I , and $\eta_{il}^k (l = 1, 2, 3)$ is the aggregation weight. Considering that the trust intensity of users to each friend is different, 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 node neighbors. The similarity can be measured from the perspective of scoring or trust data, given item's node representation v_i^{k-1} and all of its selected neighbors are described as:

$$\mu_{ij}^k = \frac{v_i^{k-1} \cdot v_j^{k-1}}{| v_i^{k-1} | \times | v_j^{k-1} |} , \tag{21}$$

$$v_{ia}^k = \sigma(v_i^{k-1} \odot u_a^{k-1}) , \tag{22}$$

Because the graph attention weights $\gamma_{al}^{k}(l=1,2,3)$ in user-space is achieved, here an attention network is used to learn the item graph attention weight $\eta_{il}^{k}(l=1,2,3)$:

$$\eta_{i1}^{k} = MLP_4^k(v_i^{k-1}), \tag{23}$$

$$\eta_{i2}^{k} = MLP_5^k([v_i^{k-1}, m_i^{\sim k}]), \tag{24}$$

$$\eta_{i3}^{k} = MLP_6^k([v_i^{k-1}, n_i^{\sim k}]), \tag{25}$$

where 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($\overset{\circ}{m}_i^{k}$ and $\overset{\circ}{n}_i^{k}$).

3.4. Rating Prediction

This paper comprehensively considered the effect of score and trust similarity on the strength of trust among users. With the latent embedding of user a and item i at layer k (i.e., u_a^k and v_i^k) for $k=[0,1,2,\ ...,\ K]$, we can first concatenate them at each layer to get the final user embedding $u_a^* = \left[u_a^0 \,\|\, u_a^1 \,\|\, ... \,\|\, u_a^K\right]$ and the final item embedding $v_i^* = \left[v_i^0 \,\|\, v_i^1 \,\|\, ... \,\|\, v_i^K\right]$. Then, the user's potential preference information is usually implied in the scoring matrix, which is the main data adopted by the recommendation algorithm:

$$\hat{r}_{ai} = [u_a^0 \,\|\, u_a^1 ... \,\|\, u_a^K]^T [v_i^0 \,\|\, v_i^1 \,\|\, ... \,\|\, v_i^K], \tag{26}$$

3.5. Model Training

To give the parameters of CMAN, Bayesian Personalized Ranking loss (BPR) function[12] is used for training, the purpose of ranking task is to learn these vectors from the score[33, 36, 37]. The loss function is defined as:

$$L = \min_{\Theta} \sum_{(a,i^+,i^-)\in R} -\ln\sigma(\hat{r}_{ai^+} - \hat{r}_{ai^-}) + \lambda \,\|\, \Theta \,\|_2^2 \tag{27}$$

where $R = \left\{\left(a,i^+,i^-\right) | (a,i+) \in R^+, \left(a,i^-\right) \in R^-\right\}$ denotes the training set, R^+ represents the set of positive samples and R^- represents the set of negative samples (the user scoring matrix can learn the potential vectors of users and products to make predictions). $\sigma(x)$ is sigmoid function and Θ is regularization parameters set, i.e., $\Theta = [P, Q, W_1, W_2, [MLP_i^k]_{i=1,2,3,4,5,6}]$.

4. Experimental Results and Analysis

Yelp. Users in Yelp can rate local services and follow others that they like. 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 (Since scoring and trust data are usually sparse, the values of 0 and K shown in Table 2 are relatively small). Similar to previous works, this paper considers the ratings larger than 3 as "My Likes" of this user.

Flickr. Flickr is an online photo sharing website. Users follow other users and share funny images to friends, family and social media followers based on their preferences. The original dataset provides a great deal of preference information and social information.

4.1. Experiments Settings

This paper evaluated our proposed model on two representative data sets, Yelp and Flickr, from the authors of [32,33]. As they did in the study, this paper kept only users with at least two ratings and two social links, and filters with fewer than two interactions. In addition, this paper performs additional preprocessing steps to extract at least 2 similar item pairs that users like and use them as the edge information of our model. Note that item pairs (item links) are very sparse, so let's further consider the available links. The statistical results of the final data set are summarized in **Table 2**. This paper randomly selected 85% of the data for training, 5% for validation, and the remaining 10% for testing.

Table 2. The statistics of the two datasets

Dataset	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%

To evaluate the top-K recommendation performance of the model, this paper used the recall based measure HR@K(hit rate) and the rank-based measure NDCG@K(normalized discount cumulative gain), which are widely used in top-K recommendation tasks [33,38]. Specifically, HR@K measures the percentage of test items that are successfully recommended in the top-K recommendation list, and NDCG@K further considers the ranking of test items in the top-K recommendation list. For these two indicators, the higher the value, the better the recommendation result. In our experiment, for many recommended tasks [32,39], this paper randomly selected 1000 unrated items for each user as negative items. We repeated each experiment 10 times and reported the average score of optimal performance for both indicators.

To evaluate the performance, this paper compare our CMAN against ten SOA baselines including traditional CF methods, social based recommender approaches and graph neural network based models. The baselines are detailed as below.

BPR [40]: 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 decomposition method which considers pairwise feature interactions.

SocialMF [41]: 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 [42]: 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.

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.

NGCF [2]: A deep neural network based framework leveraging high-order signals in user-item bipartite graph.

DiffNet[32]: A graph neural network based model that simulates social influence propagation.

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

LCELS [26]:A low-dimensional space Diffusion Network for social recommendation.

Table 3. Comparison of the baselines

Model	Model Input		Model Embedding Ability			
	F	S	UU	UI	IU	II
BPR[40]	×	×	×	√	×	×
FM[10]	√	×	×	√	×	×
SocialMF[41]	×	√	√	√	×	×
TrustSVD[24]	×	√	√	√	×	×
ContextMF[42]	√	√	√	√	×	×
GraphRec[31]	×	√	√	√	×	×
PinSage[30]	√	×	×	√	√	×
NGCF[2]	×	×	×	√	√	×
DiffNet[32]	√	√	√	√	×	×
DiffNet++[33]	√	√	√	√	√	×
CMAN	√	√	√	√	√	√
CMAN-nf	×	√	√	√	√	√
CMAN-ns	√	×	×	√	√	√
CMAN-nii	√	√	√	√	√	×

Table 3 shows all the baselines and the key features of our models, showing what information each model leverages. Specifically, we use "F" for feature input and "S" for social network input. In the modeling process, "UU" and "UI" were used to represent social information and interest information used for user embedded learning, and "IU" and "II" were used to represent interest information and project homogeneity information used for project embedded learning. Note that our proposed CMAN is the only one of these models that considers project homogeneity information. Because our proposed model, CMAN, is flexible and can be reduced to a simpler version, we also constructed several variants of CMAN as ablation studies. We use CMAN-NF, cman-NS, and Cman-NII to represent a simplified version of CMAN when deleting user and item characteristics, deleting social network input, and deleting item homogeneity information.

We implemented our proposed model using the Tensorflow framework, which optimized all models using the Adam optimizer with a batch size of 512. ,32,64 [16] and [0.0005, 0.001, 0.005, 0.01, 0.05, 0.1] search is embedded in size and more. We randomly initialize the user/item free embedding parameters and weight parameters with a Gaussian distribution, where the mean and standard deviation of all models are set to 0 and 0.1, respectively. SRAN in our proposed model, we in [0.0001, 0.0003, 0.001, 0.003, 0.01] in search of regularization parameter, and the proposed model reaches the best performance for Yelp dataset and Filckr dataset respectively when $\lambda_1 = 0.001, \lambda_2 = 0.003$. In addition, we have empirically set the hidden layer size to be the same as the embedded size and the activation function to be the same as Leaky ReLU. We carefully adjust the parameters of all baselines to ensure optimal performance for fair comparison.

4.2. Performance of Our Model and Baselines (RQ1)

We start by comparing CMAN's Top-10 recommendation performance with other baselines. Table 3 presents the overall rating prediction accuracy w.r.t. HR and NDCG for the recommendation methods with different embedding sizes *D* on the Yelp and Flickr data sets. We observe the following points. Firstly, the model based on graph neural network usually has better performance than the traditional model, including the classic CF model (such as BPR [12], FM[10]) and the society-based recommendation method (such as SocialMF [41], TrustSVD[24], ContextMF[42]). This observation makes sense because traditional models fail to capture important nonlinear relationships between users and objects. However, graph neural network-based models take into account higher-order social networks or higher-order user-item interaction information. The second observation is that, compared with other methods (such as PinSage[30], NGCF[2]), models with attention mechanism (such as GraphRec[31], diffnet++[33]) obtain better performance. This is not surprising, as this attention mechanism helps to better understand the implicit relationships between different nodes and aspects, improving recommendation performance. Thirdly, both social information and interest information play an important role in improving recommendation results. The performance of diffnet++ was significantly better than other baselines, making it the strongest baseline model, and our SRAN provided the best performance across all data sets (shown in Table 4).

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

Model	Yelp				Flickr			
	HR		NDCG		HR		NDCG	
	D=32	D=64	D=32	D=64	D=32	D=64	D=16	D=64
BPR[12]	0.261	0.263	0.157	0.155	0.081	0.079	0.061	0.063
FM[10]	0.283	0.286	0.172	0.172	0.121	0.123	0.087	0.095
SocialMF [41]	0.271	0.279	0.169	0.168	0.106	0.117	0.086	0.096
TrustSVD [24]	0.285	0.294	0.171	0.175	0.134	0.140	0.106	0.108
ContextMF [42]	0.301	0.304	0.181	0.182	0.138	0.143	0.109	0.110
GraphRec [31]	0.291	0.291	0.168	0.181	0.121	0.123	0.090	0.093
PinSage[30]	0.296	0.305	0.179	0.186	0.123	0.126	0.094	0.099
NGCF[2]	0.307	0.304	0.184	0.188	0.115	0.119	0.088	0.094
DiffNet[32]	0.344	0.346	0.209	0.212	0.159	0.166	0.112	0.127
DiffNet++ [33]	0.355	0.369	0.216	0.226	0.168	0.183	0.121	0.142
LCELS [26]	0.339	0.342	0.199	0.209	0.147	0.151	0.107	0.115
CMAN	**0.363**	**0.384**	**0.223**	**0.238**	**0.175**	**0.197**	**0.127**	**0.154**
CMAN-ns	0.357	0.376	0.223	0.234	0.171	0.189	0.121	0.148
CMAN-nii	0.361	0.380	0.222	0.237	0.173	0.195	0.126	0.152

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=10	N=15	N=10	N=15	N=10	N=15	N=10	N=15
BPR[12]	0.263	0.325	0.155	0.175	0.079	0.103	0.062	0.073
FM[10]	0.282	0.344	0.171	0.187	0.123	0.147	0.095	0.106
SocialMF [41]	0.278	0.336	0.167	0.184	0.117	0.130	0.096	0.106
TrustSVD [24]	0.293	0.369	0.174	0.198	0.140	0.173	0.108	0.120
ContextMF [42]	0.304	0.383	0.182	0.208	0.143	0.176	0.110	0.113
GraphRec [31]	0.291	0.362	0.181	0.195	0.123	0.148	0.093	0.099
PinSage[30]	0.305	0.386	0.185	0.214	0.126	0.150	0.099	0.105
NGCF[2]	0.304	0.375	0.183	0.204	0.119	0.139	0.094	0.099
DiffNet[32]	0.346	0.422	0.212	0.231	0.166	0.185	0.127	0.130
DiffNet++ [33]	0.369	0.449	0.226	0.249	0.183	0.220	0.142	0.154
LCELS [26]	0.323	0.417	0.199	0.218	0.173	0.213	0.139	0.156
CMAN	**0.384**	**0.461**	**0.238**	**0.261**	**0.197**	**0.233**	**0.154**	**0.165**
CMAN-ns	0.376	0.448	0.234	0.256	0.189	0.222	0.148	0.158
CMAN-nii	0.380	0.457	0.237	0.260	0.195	0.232	0.152	0.164

In this experiment, the validity of proposed model is measured with different top-N values in Table 5, and the overall trend is similar to the previous analysis. For example, slan-ns implements 0.2608HR@5 and 0.1928NDCG@5 in Yelp, while slan-nii implements 0.2609HR@5 and 0.1940NDCG@5. Both variants of SRAN outperformed all baselines in Yelp, and RAN-NII was even more competitive than RAN-NS. The same experimental results were reflected in the Flickr dataset, confirming that both the social network and the project-project graph contributed positively to the performance of our CMAN. Therefore, we can conclude that CMAN can improve recommendation performance by capturing high order heterogeneous information between user-user, item-item, and user-item in aggregation operations through two attention mechanisms.

4.3. Effectiveness of Our Attention Mechanisms (RQ2) and Diffusion Depth K (RQ3)

In this paper, two attention mechanisms are proposed, namely :(1) node attention block in the process of influencing diffusion; (2) Graphic attention block in the process of information aggregation. To investigate the role of these two different attention mechanisms, we compared the CMAN model with a number of model variables. We use AVG to denote an attention mechanism that degrades to equal rights of attention without any learning process. We have done some ablation studies and the results of different attention modeling combinations are shown in Table VI. In particular, we ran each submodule of SRAN with/without the corresponding attention mechanism (i.e., ATT or AVG) and found that the best performance was achieved by combining node-level attention and graph-level attention. The experimental results show that both nodes and graph attention blocks can improve the performance of the model by distinguishing important weights.

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

Graph mode	Node mode	Yelp				Flickr			
		HR	Improve	NDCG	Improve (%)	HR	Improve (%)	NDCG	Improve (%)
AVG	AVG	0.374	-	0.232	-	0.181	-	0.141	-
AVG	ATT	0.374	+0.16%	0.233	+0.56	0.181	+0.06	0.141	+0.28
ATT	AVG	0.381	+1.9%	0.237	+2.07	0.195	+7.80	0.151	+7.53
ATT	ATT	0.384	+2.7%	0.238	+2.63	0.197	+8.96	0.154	+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	0.263	-	0.155	-	0.079	-	0.063	-
		31.40%		34.76%		59.64%		40.81%
K = 1	0.375	-	0.233	-	0.181	-	0.142	-
		2.32%		2.14%		8.22%		7.86%
K = 2	0.384	-	0.238	-	0.197	-	0.154	-
K = 3	0.388	+1.20%	0.241	+1.13%	0.203	+3.10%	0.157	+2.01%
K=4	0.392	+2.11%	0.244	+2.48%	0.207	+5.23%	0.162	+5.39%

Next, we analyze the sensitivity of our model to the diffusion depth K and which depth value yields the best recommended results. In Table 7, we report the experimental results of SRAN in two data sets with different K values. It is worth noting that many related studies have achieved the best performance at K=2 [30,33], and performance declines as the depth of the graph continues to increase. The "Improvement" column shows the performance change compared to the SRAN setting, that is, K=2. We found that the performance improved rapidly as K increased from 0 to 1, while the performance still improved slightly as the diffusion depth continued to increase. We conclude that the application of these two newly proposed attention mechanisms alleviates the over-smoothing problem in graph neural network training and preserves the differences in the representation of the hidden layer of each node.

5. Conclusions

In this paper, a novel framework CMAN is proposed, which effectively recommends relevant items to users. Compared with the existing algorithm, this paper first to trust matrix decomposition, avoid Pearson correlation or cosine distance must have a common object, at the same time, fully consider the trust relationship implicit similarity, improve the accuracy of sparse data, finally, this paper jointly consider the score similarity and trust relationship implicit similarity on user similarity, further improve the recommendation accuracy. This paper focuses on the improvement of trust relationship implicit similarity to social recommendation algorithms. Other related issues, such as directed trust delivery, time drift, and product characteristics, will be further explored in future work.

Acknowledgment. This work is supported by the 111 Project (B12018); Innovative Research Foundation of Ship General Performance (14422102).

References

1. Chen, X., et al.: DNNOff: Offloading DNN-based Intelligent IoT Applications in Mobile Edge Computing. IEEE Transactions on Industrial Informatics, 612-623. (2021)
2. Wang, X., et al.: Neural graph collaborative filtering. in Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval. 1208-1313. (2019)
3. Zhang, J., et al. STAR-GCN: Stacked and Reconstructed Graph Convolutional Networks for Recommender Systems. in IJCAI, 612-623. (2019)
4. Chen, X., Li, M., Hsu, C. H.: A Reinforcement Learning Empowered Feedback Control System for Industrial Internet of Things. IEEE Transactions on Industrial Informatics, 1109-1124. (2021)
5. Lin, B., Huang, Y., Zhang, J. S., Hu, J., Li, J.: Cost-Driven Offloading for DNN-based Applications over Cloud, Edge and End Devices. IEEE Transactions on Industrial Informatics, Vol. 16, No. 8, 5456-5466. (2020)
6. Lin, B., Zhu, F., Zhang, J. S., Chen, J., Mauri, J. L.: A Time-driven Data Placement Strategy for a Scientific Workflow Combining Edge Computing and Cloud Computing. IEEE Transactions on Industrial Informatics, Vol. 15, No. 7, 4254-4265. (2019)
7. Do, M.P.T., Nguyen, D. V. and Nguyen, L.: Model-based approach for Collaborative Filtering. in Model-based Approach for Collaborative Filtering. 390-407. (2010)
8. Jenatton, R., et al.: A latent factor model for highly multi-relational data. in International Conference on Neural Information Processing Systems. 249-275.(2012)

9. Chen, X., et al.: Resource Allocation for Cloud-based Software Services Using Prediction-Enabled Feedback Control with Reinforcement Learning. IEEE Transactions on Cloud Computing, 111-125. (2022)
10. Rendle, S.: Factorization Machines. in ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17, December (2010)
11. Rendle, S., et al.: BPR: Bayesian personalized ranking from implicit feedback. in UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June, 18-21. (2009)
12. Rendle, S.: Factorization Machines with libFM. Acm Transactions on Intelligent Systems & Technology, 34-46, December (2012)
13. Zhou, X., et al.: SVD-based incremental approaches for recommender systems. Journal of Computer and System Sciences. Vol. 81, No. 4, 717-733. (2015)
14. Koren, Y., Bell, R. and Volinsky, C.: Matrix factorization techniques for recommender systems. Computer. Vol. 42, No. 8, 30-37. (2009)
15. Sarwar, B., et al.: Application of dimensionality reduction in recommender system-a case study. 2000, Minnesota Univ Minneapolis Dept of Computer Science,Montreal (2000)
16. Fan, M.: Non-negative matrix factorization and clustering methods application research in personalized recommendation system. East China Jiaotong University (2012).
17. Wang, X., et al.: Social recommendation with strong and weak ties. in Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 25-37. (2016)
18. Tang, J., Hu, X. and Liu, H.: Social recommendation: a review. Social Network Analysis and Mining. Vol. 34, No. 4, 1113-1133. (2013)
19. Ma, H., et al.: social recommendation using probabilistic matrix factorization. in Proceedings of the 17th ACM conference on Information and knowledge management. 41-49. (2008).
20. Yang, B., et al.: Social collaborative filtering by trust. IEEE transactions on pattern analysis and machine intelligence. Vol. 39, No. 8, 1633-1647. (2016)
21. Fang, H., Bao, Y., and Zhang, J.: Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation. in Proceedings of the AAAI Conference on Artificial Intelligence. 11-18. (2014)
22. Tang, J., et al.: Exploiting local and global social context for recommendation. in IJCAI. 2013. Citeseer (2013)
23. Koren, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. in Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. 61-73. (2008)
24. Guo, G., Zhang, J., Yorke-Smith, T., Trustsvd, N.: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. in Proceedings of the AAAI Conference on Artificial Intelligence. 458-465. (2015)
25. Zhou, J., et al.: Graph neural networks: A review of methods and applications. AI Open. Vol. 1, 57-81. (2020)
26. Chen, S., et al.: Learning Contrastive Embedding in Low-Dimensional Space. Annual Conference on Neural Information Processing Systems. 324-332. (2022)
27. Velickovic, P., et al.: GRAPH ATTENTION NETWORKS. stat. Vol. 10, No. 50, 41-57. (2018)
28. Hamilton, W.L., Ying, R., and Leskovec, J.: Inductive representation learning on large graphs. in Proceedings of the 31st International Conference on Neural Information Processing Systems. 1231-1238. (2017)
29. Abualigah, L., Diabat, A., Mirjalili, S., et al.: The Arithmetic Optimization Algorithm. Computer Methods in Applied Mechanics and Engineering, 376-389. (2021)
30. Ying, R., et al.: Graph convolutional neural networks for web-scale recommender systems. in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 136-145. (2018)

31. Fan, W., et al.: Graph neural networks for social recommendation. in The World Wide Web Conference. 458-468. (2019)
32. Wu, L., et al.: A neural influence diffusion model for social recommendation. in Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval. 1109-1129. (2019)
33. Wu, L., et al.: Diffnet++: A neural influence and interest diffusion network for social recommendation. IEEE Transactions on Knowledge and Data Engineering, 36-45. (2020)
34. Sarwar, B., et al.: Item-based Collaborative Filtering Recommendation Algorithms. ACM, 121-137. (2001)
35. Wu, Q., et al.: Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. in The World Wide Web Conference. 21-27. (2019)
36. Tong, Z., Mcauley, J. and King, I.: Leveraging Social Connections to Improve Personalized Ranking for Collaborative Filtering. in the 23rd ACM International Conference. 1-8. (2014)
37. Chen, J., et al.: Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention. in International Acm Sigir Conference. 511-523. (2017)
38. You, D., et al.: Attributed Multi-Relational Attention Network for Fact-checking URL Recommendation. in The 28th ACM International Conference on Information and Knowledge Management (CIKM). 1358-1371. (2019)
39. He, X., et al.: Neural collaborative filtering. in Proceedings of the 26th international conference on world wide web. 590-602. (2017)
40. Rendle, S., et al.: Bayesian personalized ranking from implicit feedback. in Proc. of Uncertainty in Artificial Intelligence. 1246-1258. (2014)
41. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks, in Proceedings of the fourth ACM conference on Recommender systems. 2010, Association for Computing Machinery: Barcelona, Spain. 135–142. (2010)
42. Jiang, M., et al.: Scalable Recommendation with Social Contextual Information. IEEE Transactions on Knowledge & Data Engineering. Vol. 26, No. 11, 2789-2802. (2014)

Bin Wu received the B.Sc. degree from Jiangnan University, Wuxi, China, in 1996, and the M.S. degree from Jiangnan University, Wuxi, China, in 2005. He is currently a lecturer with the School of Internet of Things Engineering, Jiangnan University. His major research interests include visual surveillance, object detection, integrated circuit design and application of embedded system.

Tao Zhang received the bachelor's degree from Henan Polytechnic University, Jiaozuo, China, in 2008, and the Ph.D degree from the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China, in 2016. He is currently an Associate Professor with the Jiangsu Provincial Engineering Laboratory for Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi, China. He has led many research projects (e.g., the National Science Foundation and the National Joint Fund), He has authored over thirty quality journal articles and conference papers. His current research interests include data mining, information systems, wireless network, artificial intelligence, IoT and security, medical data analysis, visual surveillance, scene understanding, behavior analysis, object detection, and pattern analysis.visual monitoring, scene understanding, behavior analysis, target detection and pattern analysis.

Yeh-Cheng Chen is a Ph.D. at the Department of Computer Science, University of California, Davis, CA, USA. His research interests are radio frequency identification (RFID), data mining, social network, information systems, wireless network artificial intelligence, IoT and security.

Received: July 12, 2022; Accepted: February 10, 2023.