Recommendation Model Based on Social Homogeneity Factor and Social Influence Factor

Weizhi Ying[1], Qing Yu[1], and Zuohua Wang[2(✉)]

[1] Tianjin Key Laboratory of Intelligence Computing and Network Security, Tianjin University of Technology, Tianjin 300384, China
[2] China Everbright Bank CO. LTD., Beijing 100035, China

Abstract. In recent years, more and more recommendation algorithms incorporate social information. However, most social recommendation algorithms often only consider the social homogeneity factor between users and do not consider the social influence factor. To make the recommendation model more in line with the real-life situation, this paper proposes a novel graph attention network to model the homogeneity effect and the influence effect in the user domain. Besides, we also extended this idea to the item domain, using information from similar items to alleviate the problem of data sparsity. Also, considering that there will be interactions between the user domain and the item domain, which together affect the user's preference for the item, we use a contextual multi-armed bandit to weigh the interaction between the two domains. We have conducted extensive comparative experiments and ablation experiments on two real public datasets. The experimental results show that the performance of our proposed model in the rating prediction task is better than other social recommendation model.

Keywords: Graph attention network · Social recommendation · Multi-armed bandit

1 Introduction

With the rapid development of the Internet, people have entered an information overload era [1]. Since the traditional collaborative filtering recommendation methods usually have problems with the data-sparse and cold start. Therefore, many scholars propose to use social networks to establish a novel recommendation system (i.e., social recommendation) to alleviate the problems mentioned above.

In the social recommendation, the user's decision is often affected by various factors. As shown in Fig. 1, users are affected by the user homogeneity factor and user influence factor when making decisions in the user domain. User homogeneity factor means that users and social friends often have similar preferences [2]. The user influence factor implies that the user may recommend the item to his friends after they purchase (or click) an item [3]. Besides, the item homogeneity factor and item influence factor also exist in the item domain. Item homogeneity factor means that similar items have similar attractiveness. Also, the item influence factor means that the user clicks on

H. Gao and X. Wang (Eds.): CollaborateCom 2021, LNICST 406, pp. 83–98, 2021.
https://doi.org/10.1007/978-3-030-92635-9_6

an item, and its similar items are more likely to be clicked by the other user. Finally, the four factors will affect the user's decisions together.

Previous research on social recommendation has adopted many ways to simulate social effects [4–6]. Converts social information into the regularization term [7, 8]. Uses network embedding methods to map each user into a low-dimensional vector [9, 10, 19]. Uses graph neural networks to obtain the social diffusion process.

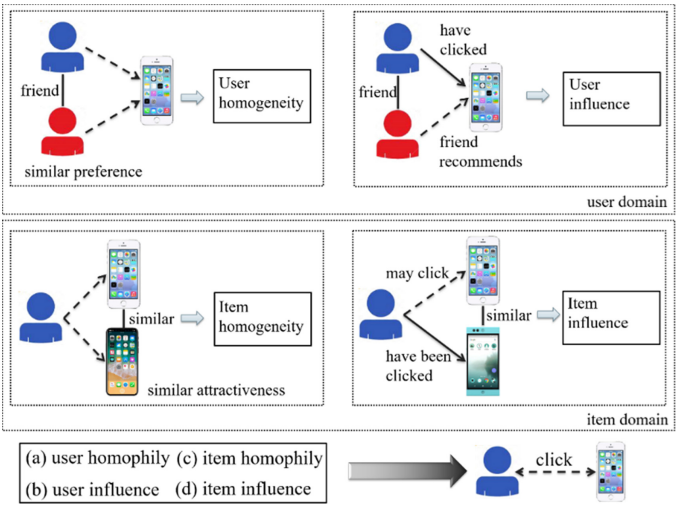

Fig. 1. Social homogeneity factor and social influence factor.

However, the studies mentioned above have the following shortcomings: First, most research on social recommendation only considers the effect of social homogeneity among users. There is not only a social homogeneity but also exists social influence among users. Secondly, many social recommendation studies assume that restricts social friends' influence through constant weights. This way of restraint does not conform to the social situation in real life.

Therefore, in this paper, we focus on the rating prediction task and propose RMHFIF (**R**ecommendation **M**odel Based on Social **H**omogeneity **F**actor and Social **I**nfluence **F**actor), a novel social recommendation model based on the **N**ovel **G**raph **A**ttention Net**w**ork (**NGAT**). Specifically, the RMHFIF model consists of four NGAT modules. In the user domain, an NGAT is used to aggregate the embedding vectors of neighboring users that reflect user preferences to obtain user homogeneity factor; an NGAT is used to convolve the context-aware preferences of neighboring users to obtain user influence factor. Similarly, in the item domain, we also use NGAT to model item homogeneity factor and item influence factor. To verify the effectiveness of the model proposed in this paper, we conducted experiments on two public datasets. In summary, the main contributions of this paper are as follows:

- We propose a novel graph attention weight calculation method. And we apply it to model the social homogeneity factor and social influence factor.
- We propose the RMHFIF model, which models the four factors in the user domain and the item domain. It enables the model to weigh the homogeneity factor and influence factor according to the environment.
- We conducted extensive experiments on public datasets. The experimental results showed that our proposed model is better than the other recommendation model.

2 Related Work

In recent years, the integration of social relationships into recommendation algorithms has received a lot of attention [18, 20]. The social recommendation model assumes that the preferences of users and his/her friends are similar. Many scholars have proposed many social recommendation models based on this assumption. Ma H et al. proposed the SoRec method, which is a social recommendation algorithm based on feature sharing [21]. In this method, the user's feature vector is learned by simultaneously decomposing the rating matrix and the social relationship matrix, so that the learned user feature vector can take into account the user's rating habits and social characteristics. Jamali et al. proposed the SocialMF method, which assumes that the user's behavioral preferences should be similar to the average preferences of the user's social neighbors [6]. This method adds the trust propagation mechanism to the model, which can improve the accuracy of the recommendation for cold-start users. Lin et al. proposed the CSR method [22]. This method models the characteristics of social relationships and models the influence characteristics of social relations. Rafailidis D proposed the SDPL method, which is a ranking model that performs social deep pairwise learning with users' trust and distrust relationships [23].

Graph neural networks can learn graph structure data. The model related to our work is GraphRec [18]. GraphRec uses the graph neural network to learn the hidden factors of users and items in the user-user social graph and user-item feedback graph, and finally completes the task of rating prediction. Although the previous work has achieved good results, the previous work only modeled the homogeneity in social relationships and did not model the influence effects in social relationships. In this paper, we propose the RMHFIF model to fill this gap.

3 Preliminary and Problem Definition

3.1 Novel Graph Attention Network

The setting of the attention weight in the traditional graph attention network [11] is shown in Eq. (1). In Eq. (1), $\overrightarrow{h}_i$ represents the characteristics of the input node, W represents the weight matrix, N_i is the neighborhood of node i in the graph, and $\overrightarrow{a}$ is the weight vector.

$$\alpha_{ij} = \frac{\exp(Leaky\mathrm{Re}LU(\overrightarrow{a}^T[W\overrightarrow{h_i}||W\overrightarrow{h_j}]))}{\sum_{k\in N_i}\exp(Leaky\mathrm{Re}LU(\overrightarrow{a}^T[W\overrightarrow{h_i}||W\overrightarrow{h_k}]))} \tag{1}$$

Inspired by [24], the graph attention network uses the attention mechanism to aggregate the information of neighbor nodes. Applying it to the social network graph can be understood as each user node in the social network graph aggregates the information of neighboring user nodes. However, the traditional graph attention network weight calculation formula does not add a bias term. In real life, everyone's interests are different, and people with different interests can also become friends, which affects the user's interests. Therefore, in this section, a novel calculation method for the attention weight of the graph attention network is proposed. The bias term is added to the original weight calculation formula. Using this calculation method can provide a better social graph representation, the specific expression is as follows:

$$\alpha_{ij} = \frac{\exp(\tanh(\overrightarrow{a}^T[W\overrightarrow{h_i}||W\overrightarrow{h_j}] + b_{ij}))}{\sum_{k\in N_i}\exp(\tanh(\overrightarrow{a}^T[W\overrightarrow{h_i}||W\overrightarrow{h_k}] + b_{ik}))} \tag{2}$$

3.2 Notations

Table 1 shows the mathematical notations and their definitions used in this paper.

Table 1. Notation.

Symbols	Definitions
R	user-item rating matrix
r_{ui}	user u's rating for item i
M	the number of users
N	the number of items
$C_I(u)$	the set of items rated by user u
$C_U(i)$	the set of users who have rated item i
V_U	the set of users
E_U	the set of edges connecting users
SN_U	the social network between users, $SN_U = (V_U, E_U)$
e_{uv}	the frequency of interaction between user u and user v
$F_U(u)$	the set of the user u's social friends
$F_I(i)$	the set of item i's similar items

3.3 Problem Definition

The definition of the social recommendation problem for this paper is as follows: Given a rating matrix R and the social network SN_U, predict the user's rating for items that have not yet been rated.

4 The Proposed Model

The architecture of the proposed model is shown in Fig. 2. In this section, we will introduce in detail each module of the model and the method of model training.

4.1 Model Details

4.1.1 Original Input and Similar Item Network

First, we require rating matrix R and social network SN_U as input. The similarity coefficient between items Sim_{ij} is defined as the number of users who rate item i and item j at the same time. Specifically, given a threshold τ, if $Sim_{ij} > \tau$ then item i is similar to item j. We define the similar item set as V_I and connect similar items to form an edge set E_I. V_I and E_I together form a similar items network $SN_I = (V_I, E_I)$.

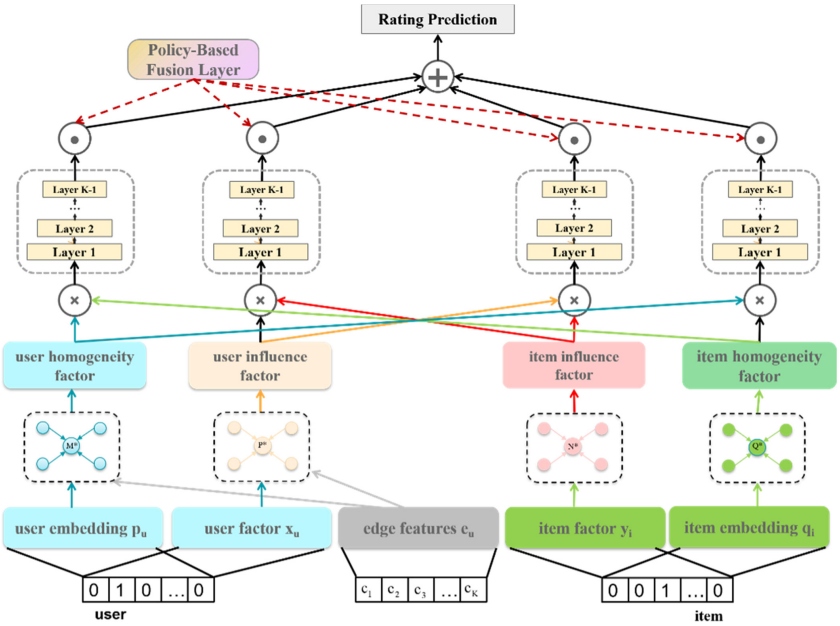

Fig. 2. The architecture of the proposed model.

4.1.2 Embedding Layer

Because each user or item's original input is a high-dimensional and sparse one-hot vector. We first need to map each user vector or item vector to a low-dimensional dense vector representation through the embedding operation. Inspired by [12], each user vector can be mapped into a specific user embedding $P = \{p_u\}_{D\times M}$ and a rating-based user embedding factor $X = \{x_u\}_{D\times N}$. The specific user embedding reflects the user's interest, and the rating-based user embedding factor reflects the implicit influence of the user's past rating on the current decision. Similarly, each item can be mapped to a

specific item embedding $Q = \{q_i\}_{D\times N}$ and rating-based item embedding factor $Y = \{y_i\}_{D\times M}$. The specific item embedding reflects the item's attributes, and the rating-based item embedding factor reflects the influence of the item's past rating on the user's decision-making.

4.1.3 NGAT Layer

1) Using NGAT to obtain user homogeneity factor

After the embedding layer's operation, we have a specific user embedding P. We use Eq. (3) to calculate and output the user homogeneity factor M^*:

$$M^* = \sigma(Att_M(SN_U)PW_M^T + b_M) \tag{3}$$

In Eq. (3), σ, W_M and b_M are the activation function, weight matrix, and bias vector, respectively. $Att_M(SN_U) = \{\alpha_{uv}^M\}_{M\times M}$ represents the attention weight between user u and user v obtained from SN_U. α_{uv}^M is calculated by Eq. (4)− Eq. (6).

$$\alpha_{uv}^M = \text{softmax}(\alpha_{uv}^{M*}) = \frac{\exp(\alpha_{uv}^{M*})}{\sum\limits_{w\in\Gamma_U(u)} \exp(\alpha_{uw}^{M*})} \tag{4}$$

$$\alpha_{uv}^{M*} = attn_U(W_M p_u, W_M p_v, W_E e_{uv}) \tag{5}$$

$$attn_U(x, y, z) = \tanh(W_U^T z \otimes (x||y) + b_U) \tag{6}$$

In Eq. (4), $\Gamma_U(u) = \{u\} \cup F_U(u)$. In Eq. (5), W_E represent the weight matrix of edge features e_{uv}. In Eq. (6), W_U and b_U are the weight matrix and the bias vector, respectively. $\otimes$ represent the element-wise product and $||$ represent concatenation.

2) Using NGAT to obtain item homogeneity factor

Like the user homogeneity factor, after the embedding layer's operation, we have specific items embedding Q. We use Eq. (7) to combine similar item information, calculate and output the item homogeneity factor Q^*.

$$Q* = \sigma(Att_Q(SN_I)QW_N^T + b_N) \tag{7}$$

In Eq. (7), $Att_Q(SN_I) = \{\beta_{ij}^Q\}_{N\times N}$ represents the attention weight between item i and item j obtained from SN_I. β_{ij}^Q is calculated by Eq. (8)−Eq. (10).

$$\beta_{ij}^Q = \text{softmax}(\beta_{ij}^{Q*}) = \frac{\exp(\beta_{ij}^{Q*})}{\sum_{k\in\Gamma_I(i)} \exp(\beta_{ij}^{Q*})} \tag{8}$$

$$\beta_{ij}^{Q*} = attn_I(W_Q q_i, W_Q q_j), \quad j \in \Gamma_I(i) \tag{9}$$

$$\mathrm{attn_I}(x, y) = \tanh(w_I^T(x||y) + b_I) \tag{10}$$

In Eq. (9), $\Gamma_I(i) = \{i\} \cup F_I(i)$. It's worth noting that the attention weight calculated by Eq. (7) remains unchanged after item i and item j are determined, which means Q∗ is fixed for the item.

3) Using NGAT to obtain user influence factor

Unlike the user homogeneity factor, the user influence factor is often context-aware. User influence factor is different when facing different items and different friends. Through the embedding layer, we can get the item embedding X_u, which is clicked by user u. Use Eq. (11) to make the item that clicked by the user u interact with the item i^+ that has not yet been clicked.

$$X_u^{i^+} = \{x_j \otimes x_{i^+} | j \in C_I(u)\} \tag{11}$$

Equation (9)'s product operation can be helpful for modeling in a dynamic user influence environment. Define the rating-based user embedding $P_{i^+} = \{p_u^{i^+}\}_{D\times M}$. Next, we use the maximum pooling (MP) operation to select the most important D-dimensional features. The specific process is shown as follows:

$$p_{ud}^{i^+} = \underset{j\in C_I(u)}{\mathrm{MP}}\{x_{jd} \cdot x_{i^+ d}\},\ d = 1, \ldots, D \tag{12}$$

In Eq. (12), MP stands for maximum pooling operation. We use Eq. (13) to calculate the user influence factor $P_{i^+}^*$.

$$P_{i^+}^* = \sigma(\mathrm{Att_P}(\mathrm{SN_U})PW_P^T + b_P) \tag{13}$$

In Eq. (13), $\mathrm{Att_P}(\mathrm{SN_U}) = \{\alpha_{uv,i^+}^P\}_{M\times M}$, $\alpha_{uv,i^+}^{P^*}$ is calculated by Eq. (14)−Eq. (15).

$$\alpha_{uv,i^+}^P = \mathrm{softmax}(\alpha_{uv,i^+}^{P^*}) = \frac{\exp(\alpha_{uv,i^+}^{P^*})}{\sum_{w\in\Gamma_U(u)}\exp(\alpha_{uw,i^+}^{P^*})} \tag{14}$$

$$\alpha_{uv,i^+}^{P^*} = \mathrm{attn_U}(W_P p_u^{i^+}, W_P p_v^{i^+}, W_E e_{uv}), v \in \Gamma_U(u) \tag{15}$$

The $\mathrm{attn_U}(\bullet)$ used in Eq. (15) is the same as Eq. (6). It's worth noting that the attention weight α_{uv,j^+}^P depends on the user's rating history and the specific candidate item i^+. It means that $P_{i^+}^*$ will continue to change with the context. Such a design conforms to the real situation of social influence among users.

4) Using NGAT to obtain item influence factor

Like user influence, the calculation of the rating-based item represents $N_{u^+} = \{n_i^{u^+}\}_{D\times N}$ is shown in Eq. (16)−Eq. (17).

$$Y_i^{u^+} = \{y_v \otimes y_{u^+} | v \in C_I(i)\} \tag{16}$$

$$n_{id}^{u^+} = \underset{v \in C_U(i)}{MP} \{y_{vd} \cdot y_{u^+ d}\}, \forall d = 1, ..., D \tag{17}$$

In Eq. (17), $n_{id}^{u^+}, y_{u^+ d}, y_{vd}$ are the d-th features of $n_i^{u^+}, y_{u^+}, y_v$ respectively. Next, we calculate the item influence by Eq. (18).

$$N_{u^+}^* = \sigma(Att_N(SN_U)N_{u^+} W_N^T + b_N) \tag{18}$$

In Eq. (18), $Att_N(SN_U) = \{\beta_{ij,u^+}^N\}_{N \times N}$, and β_{ij,u^+}^N is calculated by Eq. (19)−Eq. (20).

$$\beta_{ij,u^+}^N = \text{softmax}(\beta_{ij,u^+}^{n^*}) = \frac{\exp(\beta_{ij,u^+}^{n^*})}{\sum_{k \in \Gamma_I(i)} \exp(\beta_{ik,u^+}^{n^*})} \tag{19}$$

$$\beta_{ij,u^+}^{N^*} = attn_I(W_N n_i^{u^+}, W_N n_j^{u^+}), \quad j \in \Gamma_I(i) \tag{20}$$

4.1.4 Pairwise Neural Interaction Layer

After using NGAT to obtain homogeneity factor and influence factor, inspired by [13], we make homogeneity factors and influence factors interact with each other in pairs. Input the four factors into different neural networks.

$$z_0 = [m_u^* \otimes q_i^*, m_u^* \otimes n_i^*, p_u^* \otimes q_i^*, \quad p_u^* \otimes n_i^*] \tag{21}$$

$$g_k^a(z_{k-1}) = \tanh(W_k^a z_{k-1}^a + b_k^a), k \in [1, K-1] \tag{22}$$

$$h_a = g_K^a(...g_2^a(g_1^a(z_0[a]))), a \in \{1, 2, 3, 4\} \tag{23}$$

We use the tower structure. The higher layer has fewer neurons [13].

4.1.5 Policy-Based Fusion Layer

Here, the four interactive features h_a are further merged into a synthetic feature. The homogeneity factor and the influence factor can work together, but these factors are different for different users and items. Thus, we modeled the weight distribution of the homogeneity factor and influence factor as a contextual multi-armed bandit problem. The action is represented by $\gamma \in \{1, 2, 3, 4\}$, indicates which factor is selected, and the environment is a user-item pair. The random strategy can be represented by conditional probability $p(\gamma | p_u, q_i)$, representing the probability of choosing different social effects given a specific user-item pair (u, i).

We use Eq. (24)−Eq. (25) to calculate the conditional probability $p(\gamma|p_u, q_i)$:

$$e_\gamma = w_\gamma[p_u||q_i||(p_u \otimes q_i)] + b_\gamma \tag{24}$$

$$p(\gamma|p_u, q_i) = \frac{\exp(e_\gamma)}{\sum_{s=1}^{4} \exp(e_s)} \tag{25}$$

Next, comprehensive representation can be expressed as follows:

$$h = E_{\gamma \sim p(\gamma|p_u,q_i)}(h_\gamma) = \sum_{\gamma=1}^{4} p(\gamma|p_u,q_i) \times h_\gamma \tag{26}$$

4.1.6 Output Layer and Loss Function

The model final outputs the user u's rating for unrated item i, which is specifically calculated by the following equation:

$$\widehat{r}_{ui} = w_o h + b_o \tag{27}$$

We follow the research of [18] and also use the mean square error loss function:

$$H_1 = \sum_{(u,i)\in O} (\widehat{r}_{ui} - r_{ui})^2 \tag{28}$$

In Eq. (28), O represents the set of user-item ratings in which user u has rated item i, r_{ui} is the actual rating of user u on item i and $\widehat{r}_{ui}$ represents the predicted rating.

4.2 Model Training

4.2.1 Mini-Batch Training

In each epoch, since the number of friends of each user is different, we performed a sampling operation: First, set a threshold F, and then perform sampling operations. If the number of friends exceeds F, then F friends are randomly sampled as input data. If the number of friends is less than F, 0 is filled so that the vector with the number of dimensions F is reached.

4.2.2 Alleviate Overfitting

To alleviate the over-fitting problem, we adopted L1 regularization. The following equation represents the regularization loss term:

$$H_2 = \sum_u (||p_u|| + ||y_u||) + \sum_i (||q_i|| + ||x_i||) \tag{29}$$

In summary, the final loss function of our proposed model is as follows:

$$L_{final} = H_1 + \lambda H_2 \tag{30}$$

In the experimental part, we will further discuss the influence of the choice of regularization parameter λ on the model's performance. Besides, we adopt the dropout strategy to alleviate overfitting.

5 Experiments

To comprehensively evaluate the performance of our proposed model, we conducted extensive experiments to answer the following questions:

RQ1 Compared with other recommendation models, how does our social recommendation model perform?
RQ2 The proposed model uses both the homogeneity factor and the influence factor. Only use one of them, how will the performance of the model change?
RQ3 The proposed model modifies the traditional attention weight calculation method in the graph attention network. If only the attention weight calculation method in the traditional graph attention network is used, how will the performance of the model change?
RQ4 How does the change of hyperparameters affect the performance of the model?

5.1 Dataset Introduction

We choose two public data sets for the experiment. The statistics of the datasets are shown in Table 2.

- **Ciao**: This dataset was obtained by Tang Jiliang [14] crawled from the article review website Ciao in May 2011.

Table 2. Dataset statistics.

Statistics	Ciao	Epinions
Users	7,375	49,290
Items	105,114	139,738
Ratings	284,086	664,824
Rating range	[1,5]	[1,5]
Relations	111,781	487,181

- **Epinions**: On the Epinions website, visitors can read various item reviews to determine their purchase behavior and write reviews to get rewards. This dataset is Paolo Massa crawled from the Epinions.com website [15].

5.2 Experimental Setup

5.2.1 Experimental Environment Setting

To prove the effectiveness of our proposed model, we select 80% of the user-item interaction data as the training set and the remaining 20% as the test set. All experimental environments are based on Python 3.7 and use TensorFlow 1.15. The training and testing of the model are based on the NIVIDA TESLA T4 GPU.

5.2.2 Evaluation Metrics

We use **M**ean **A**bsolute **E**rror (**MAE**) and **R**oot **M**ean **S**quare **E**rror (**RMSE**) to evaluate our proposed model's rating prediction quality.

The definition of MAE and RMSE are as follows:

$$\mathrm{MAE} = \frac{1}{\mathrm{T}}\sum_{\mathrm{i,j}} |\mathrm{R_{ij}} - \hat{R}_{\mathrm{ij}}| \tag{31}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{T}}\sum_{\mathrm{i,j}} (\mathrm{R_{ij}} - \hat{R}_{\mathrm{ij}})^2} \tag{32}$$

In Eq. (31), R_{ij} represents the rating of user i to item j, R_{ij} represents the rating of user i to item j predicted by the model, and T represents the number of ratings in the test set. We can find that the smaller MAE and RMSE, the better the model's performance.

5.2.3 Compare Models

We choose the following comparison models for comparison to verify the effectiveness of the model proposed in this paper:

- **SVD++** [12]**:** This is a recommendation model that considers explicit feedback information and implicit feedback information.
- **SoReg** [4]**:** This is a classic social recommendation model that transforms social information into social regularization terms.
- **TrustSVD** [16]**:** This model integrates rating implicit feedback and social implicit feedback information.
- **CUNE** [17]**:** This is a recommendation model that alleviates the sparsity problem of social information by generating top-k potential friend information.
- **GraphRec** [18]**:** This is a social recommendation model using a graph neural network. This model uses the social graph and user-item interaction graph.

- **SREPS** [8]**:** This is a collaborative algorithm based on the essential preference space, which combines explicit feedback, implicit feedback, and social relationships.

5.3 Comparative Experiments: RQ1

We report the comparative experiment results in Table 3. The percentages in Table 3 are improvements to our RMHFIF model over other models. It can be seen from Table 3 that our proposed RMHFIF model is always better than other models. The

Table 3. Performance comparison with other models

Model	Ciao		Epinions	
	RMSE	MAE	RMSE	MAE
SVD++	1.034	0.761	1.077	0.832
SoReg	0.995	0.752	1.072	0.824
TrustSVD	1.033	0.759	1.049	0.814
CUNE	1.028	0.766	1.063	0.819
GraphRec	0.979	0.759	1.057	0.817
SREPS	0.955	0.722	1.039	0.801
RMHFIF	**0.941**	**0.706**	**1.027**	**0.775**
Improvement	1.47%	2.22%	1.15%	3.35%

experimental results verify the effectiveness of our proposed model in the social recommendation. We have some findings from the experimental results. First, in most cases, the traditional recommendation model's performance based on matrix factorization is not as good as the recommendation model that considers social information. This result is consistent with previous studies' results. Secondly, the performance of GraphRec is significantly better than SoReg, TrustSVD, and CUNE. This result shows that graph neural network applies to social recommendations can dramatically improve the model's recommendation performance.

5.4 Ablation Experiments: RQ2

To verify whether each component in the model is effective, we conducted an ablation experiment. The experimental results are shown in Figs. 3 and 4. The model we proposed uses both the homogeneity factor and the influence factor. Therefore, we removed the homogeneity factor and the influence factor in the model to generate two variants, and we named them RMHFIF-α and RMHFIF-β, respectively. In addition, since this paper modifies the attention weight calculation method in the traditional graph attention network, to verify whether the modification is effective, we will use the traditional graph attention network to replace it, and name the variant model RMHFIF-γ. The definitions are as follows:

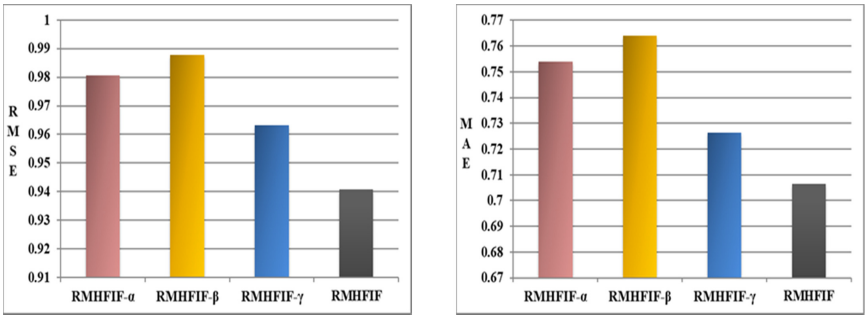

Fig. 3. Performance comparison of different variants in the Ciao dataset.

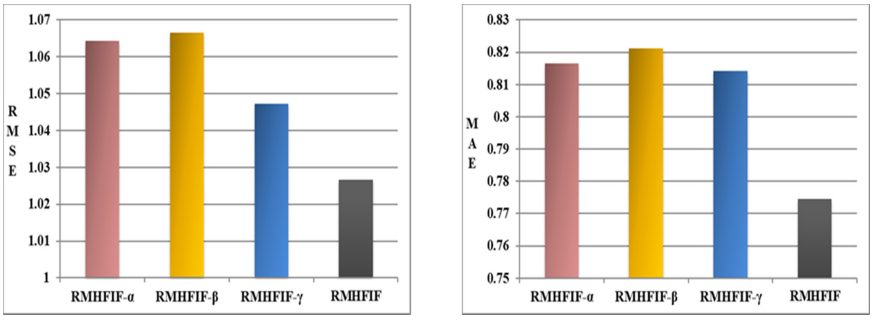

Fig. 4. Performance comparison of different variants in the Epinions dataset.

- **RMHFIF-α**: This variant removes the social homogeneity factor.
- **RMHFIF-β**: This variant removes the social influence factor.
- **RMHFIF-γ**: Modify the original graph attention network attention weight calculation method in the NGAT layer to the traditional graph attention network weight calculation method.

From the experimental results, we can find that: i) the performance of the RMHFIF model is always better than the other variants. It is enough to prove that both the homogeneity factor and the influence factor can play a role in the model, and neither is indispensable. ii) the homogeneity factor and the influence factor have similar effects on the performance of the model. Using the homogeneity factor and the influence factor at the same time can significantly improve the model performance. iii) The performance of RMHFIF is better than RMHFIF-γ, which is enough to prove that the use of a novel attention network attention weight calculation method can significantly improve the performance of the model.

5.5 Parameter Sensitivity Experiments: RQ3

This subsection studies the impact of some hyperparameter changes on performance, including dropout rate ρ, embedding dimension D, and regularization parameters λ.

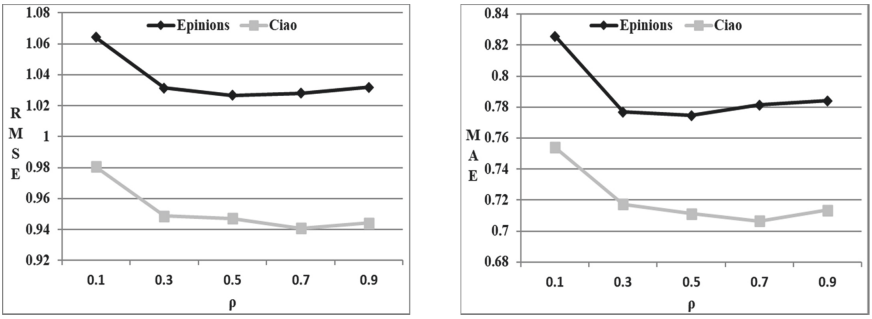

Fig. 5. The effect of dropout rate ρ on model performance.

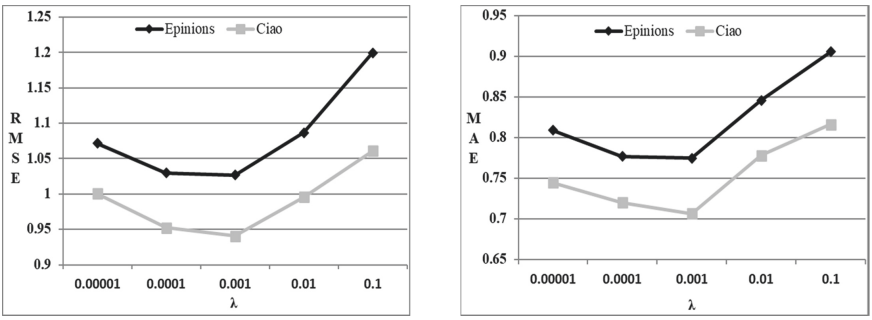

Fig. 6. The effect of regularization parameter λ on model performance.

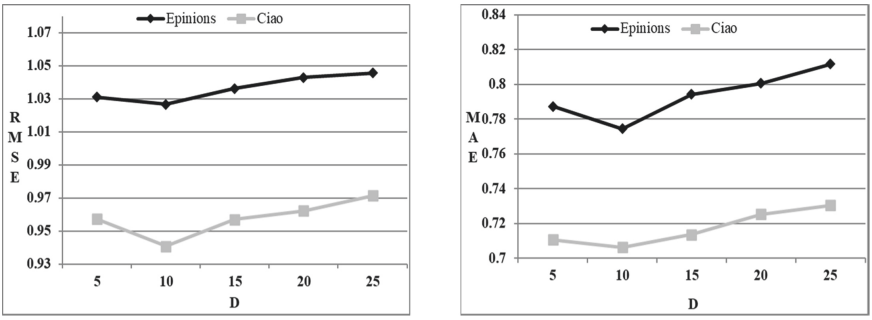

Fig. 7. The effect of embedding dimension D on model performance.

The results are shown in Figs. 5, 6 and 7. From Figs. 5, 6 and 7 we can see that: i) The choice of dropout rate ρ impacts the model's performance, and the best ρ value is different for different datasets. If the value of ρ is too large, too many neurons discarded will hinder the model's training process. If the value of ρ is too small, it will lose the ability to alleviate overfitting. ii) The choice of the regularization parameter λ is very important. If λ is too large or too small, the performance of the model will decrease. iii)

The proper embedding dimension is very important. If the embedding dimension is too small, the expressiveness of the model will decrease. If the embedding dimension is too large, the complexity of the model will increase and the representation vector will become sparse. Therefore, we need to find a suitable embedding dimension.

6 Conclusion

In this paper, we propose the RMHFIF model, which uses novel graph attention networks to obtain the homogeneity factor and the influence factor. We also model the four factor's weight distribution as a contextual multi-armed bandit problem for training. The RMHFIF model solves the problem that most of the current social recommendation models only consider social homogeneity factors without considering social influence factors. We conducted extensive comparative experiments and ablation experiments in two public datasets. The experimental results show that the performance of RMHFIF is better than the other recommendation model.

References

1. Huang, L.W., et al.: Survey on deep learning-based recommender systems. Chinese J. Comput. **41**(7), 1619–1647 (2018)
2. Mcpherson, M., Lovin, L.S., Cook, J.M.: Birds of a feather: homophily in social networks. Ann. Rev. Sociol. **27**(1), 415–444 (2001)
3. Marsden, P.V., Friedkin, N.E.: Network studies of social influence. Sociol. Methods Res. **22** (1), 127–151 (1993)
4. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: Proceeding of the 4th ACM International Conference on Web Search Data Mining, pp. 287–296 (2011)
5. Wang, X., He, X.N., Nie, L.Q., Chua, T.S.: Item silk road: Recommending items from information domains to social users. In: Proceeding of the 40th SIGIR Conference, pp. 185–194 (2017)
6. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceeding of the 4th ACM conference on Recommender systems(RecSys), pp.135–142 (2010)
7. Wen, Y.F., Guo, L., Chen, Z.M., Ma, J.: network embedding. based recommendation method in social networks. In: Proceeding of the 27th WWW Conference, pp. 11–12 (2018)
8. Liu, C.Y., Zhou, C., Wu, J., Hu, Y., Guo, L.: Social recommendation with an essential preference space. In: Proceeding of the 32rd AAAI Conference, pp. 346–353 (2018)
9. Wu, L., Sun, P., Hong, R., Fu, Y., Wang, X., Wang, M.: SocialGCN: an efficient graph convolutional network based model for social recommendation (2018). arXiv:1811.02815. https://arxiv.org/abs/1811.02815?context=cs.SI
10. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: Proceeding of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), pp. 974–983 (2018)
11. Velikovi, P., et al.: Graph attention networks (2017). arXiv:1710.10903. https://arxiv.org/abs/1710.10903

12. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceeding of the ACM SIGKDD International Conference on Knowledge Discovery Data Mining, pp. 426–434. ACM, New York, USA (2008)
13. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.: Neural collaborative filtering. In: Proceeding of the 26th WWW Conference, pp. 173–182 (2017)
14. Tang, J., Gao, H., Liu, H.: mTrust: discerning multi-faceted trust in a connected world. In: Proceeding of the 5th ACM International Conference on the Web Search Data Mining, pp. 93–102 (2012)
15. Massa, P., Avesani, P.: Trust-aware recommender systems. In: Proceeding of the 1st ACM Conference on the Recommender System, pp. 17–21. Minneapolis (2007)
16. Guo, G., Zhang, J., Yorke-Smith, N.: TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: Proceeding of the 29th AAAI Conference on Artificial Intelligence, pp. 123–129 (2015)
17. Zhang, C., Yu, L.: Collaborative user network embedding for social recommender systems. In: Proceedings of the 17th SIAM International Conference on Data Mining (SDM), pp. 381–389 (2017)
18. Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., Yin, D.: Graph neural networks for social recommendation. In: Proceeding of the 28th WWW Conference, pp.417–426 (2019)
19. Wu, Q., et al.: Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In: Proceeding of the 28th WWW Conference, pp. 2091–2102 (2019)
20. Yang, L., et al.: ConsisRec: enhancing gnn for social recommendation via consistent neighbor aggregation. In: SIGIR 21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM (2021)
21. Ma, H., Yang, H., Lyu, M.R., King, I.: SoRec: social recommendation using probabilistic matrix factorization. In: Proceeding of the CIKM, pp. 931–940 (2008)
22. Lin, T.-H., Gao, C., Li, Y.: Recommender systems with characterized social regularization. In: Proceeding of the 27th ACM International Conference on Information and Knowledge Management, pp. 1767–1770 (2018)
23. Rafailidis, D.: Leveraging Trust and Distrust in Recommender Systems via Deep Learning (2019). arXiv preprint, arXiv:1905.13612
24. Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate (2014). arXiv preprint, arXiv:1409.0473