electronics

Article

A Graph Neural Network Social Recommendation Algorithm Integrating the Multi-Head Attention Mechanism

Huawei Yi *, Jingtong Liu, Wenqian Xu, Xiaohui Li and Huihui Qian

School of Electronics and Information Engineering, Liaoning University of Technology, Jinzhou 121001, China
* Correspondence: dxxyyihuawei@lnut.edu.cn

Abstract: Collaborative filtering recommendation systems are facing the data sparsity problem associated with interaction data, and social recommendations introduce user social information to alleviate this problem. Existing social recommendation methods cannot express the user interaction interest and social influence deeply, which limits the recommendation performance of the system. To address this problem, in this paper we propose a graph neural network social recommendation algorithm integrating multi-head attention mechanism. First, based on the user-item interaction graph and social network graph, the graph neural network is used to learn the high-order relationship between users and items and deeply extract the latent features of users and items. In the process of learning user embedding vector representation based on the social network graph, the multi-head attention mechanism is introduced to increase the importance of friends with high influence. Then, we make rating predictions for the target users according to the learned user embedding vector representation and item embedding vector. The experimental results on the Epinions dataset show that the proposed method outperforms the existing methods in terms of both Recall and Normalized Discounted Cumulative Gain.

Keywords: recommendation system; data sparsity; social information; graph neural network; multi-head attention mechanism

Citation: Yi, H.; Liu, J.; Xu, W.; Li, X.; Qian, H. A Graph Neural Network Social Recommendation Algorithm Integrating the Multi-Head Attention Mechanism. *Electronics* **2023**, *12*, 1477. https://doi.org/10.3390/electronics12061477

Academic Editor: Gemma Piella

Received: 17 February 2023
Revised: 16 March 2023
Accepted: 16 March 2023
Published: 21 March 2023

1. Introduction

The rapid development of the Internet has greatly facilitated people's communication, resulting in an exponential increase in the amount of information on the network. The problem of information overload is becoming more and more serious. It is difficult for users to directly obtain relevant or interesting content from the massive source of information in a timely and accurate manner. As an important means of information filtering, collaborative recommendation systems provide an effective way to solve the problem of "information overload" on the internet and have been widely used in various fields (e.g., e-commerce, social, and education) [1–3]. The traditional recommendation system based on collaborative filtering uses historical interaction data between users and items to build a user preference model to predict user preference for items [4], but the data sparsity problem of interaction data limits the performance of the recommendation method [5–8].

With the development of social media, the interaction between users becomes more and more frequent, which makes the social data of users more and more abundant. The social information is interpretable and can well reflect the preferences of users and other users. Therefore, social recommendations are proposed, which use the direct social relationship between users as auxiliary information to alleviate the sparsity of the recommendation system [9–15]. However, most of the existing algorithms use traditional methods, such as matrix decomposition to model, and these models are not deep enough to express users' collaborative interests and social impact [16,17]. In addition, social information is directly incorporated into the model, which ignores the high-order connectivity between users and items and does not carry out in-depth mining on social information, thus limiting the system's ability to predict user preferences and failing to extract the potential feature representation of users and items implied in the data.

In recent years, graph neural network (GNN) [18] has shown a strong modeling ability for data with a graphic structure. It has excellent feature extraction abilities and can better capture the implicit representation in the data. Since most information in the recommendation system has a graph structure, the graph neural network is applied to the recommendation systems [19–22]. To solve the shortcomings of the above social recommendation, the graph neural network is used to model the collaborative interests of user interaction networks and the social impact of social networks in this paper. At the same time, considering the different influences of different trusted friends on target users could further improve the accuracy of the recommendation's results.

The main contributions of this paper are as follows:

- Based on user interaction information and social information, the graph neural network is introduced to extract the latent feature representation of users and items.
- In the process of learning user embedding vector representation, the multi-head attention mechanism is introduced to assign different weights to the trusted friends of the target user, which can increase the importance of friends with high influence, so as to obtain the interest preference of the target user more accurately.
- We design a social recommendation algorithm based on the graph neural network and multi-head attention mechanism and conduct experiments on the Epinions dataset to demonstrate the effectiveness of the proposed algorithm. Experimental results show that the proposed algorithm is better than similar algorithms.

2. Related Work

2.1. Recommendations Based on Social Network

Ma et al. [23] proposed a social recommendation algorithm using probabilistic matrix factorization, which combines the social relationship matrix and user-item interaction matrix to mine the impact of users' social relationship on target users through feature sharing. Targeting the problem of data sparsity, Zhu et al. [24] used a new location classification algorithm to identify location categories considering the user's location sign-in information and semantic location information in the social network. They constructed the user model for each user from four aspects. Experimental results showed that the proposed algorithm can achieve a better recommendation performance in comparison with other methods. Bin et al. [25] used the information propagation method to divide the community structure of multi-relational social network and combined with the collaborative filtering algorithm to recommend. Jia et al. [26] found that the matrix factorization model fusing social information ignored latent relationships between users and items. Therefore, they constructed the user similarity network and item similarity network and improved the matrix factorization model with the help of the community division. Guo et al. [27] proposed a share-based representation learning method called Trust Singular Value Decomposition (SVD), which is based on the recommendation algorithm SVD++. It not only models the user's explicit rating data, but also considers the user's implicit social information and user trust information. The experiment showed that the user trust information and rating data can complement each other, so that the model is more consistent with real-life application scenarios. Yang et al. [28] divided the user trust network into trust behavior and trusted behavior, and mixed recommendations were made by integrating the interaction information between users and items and the trust network between users so as to improve the performance of recommendations.

2.2. Recommendations Based on Graph Neural Network

Considering the influence of data sparsity on the learning node feature of graph neural network, Chen et al. [29] proposed a model combining a graph neural network and heterogeneous information network to jointly decode multi-source heterogeneous information. Bi et al. [30] designed a hierarchical social collaborative filtering framework (GHSCF) based on a GNN-fused bidirectional Long Short-Term Memory (LSTM) network, which effectively captures the information of neighbors and extracts user-item interaction sequences. Zhou et al. [31] used user interaction data and movie information to build a three-part graph. Based on the graph neural network and diffusion algorithm, they calculated the correlation between tags to obtain overlapping communities and combined them with the attribution degree and matching degree to accurately recommend personalized content to users. Chang et al. [32] proposed the short for SeqUential Recommendation with Graph neural nEtworks (SURGE) model for capturing user preferences in sequential recommendations, which reconstructs item sequences into a compact item–item interest graph. They applied cluster-aware and query-aware graph convolution propagation on the newly constructed graph to extract implicit preference signals and mined dynamic preferences through dynamic graph pooling for recommendations.

2.3. Recommendations Integrating Attention Mechanism

Yu et al. [33] introduced an attention mechanism to filter user behavior sequences when modeling the short-term preferences of users and adaptively fused the long-term and short-term preferences of users based on the attention framework. Peng et al. [34] proposed a hybrid model, HARSAM, which combines the attention mechanism with a deep neural network and uses the self-attention mechanism to extract the correlation between items in different time intervals. To extract more hidden information from the sparse check-in data of users, Pang et al. [35] used a hierarchical attention mechanism with a "local-global" structure to mine the contribution of different features for recommendations and used this information with more contributions to make recommendations. Zhuo et al. [36] proposed an attention-gated recurrent unit-based algorithm considering the dynamic variability of user preferences for items. Tao et al. [37] used user interaction data to build multimodal interaction graphs and introduced a gated attention mechanism based on GNN to distinguish the importance of different modalities to user preferences, thereby capturing hidden information and providing more accurate recommendations.

Based on the above, we compare our proposed algorithm with the works mentioned in Sections 2.2 and 2.3. Our work aims to use graph neural network and multi-head attention mechanism to jointly model and make recommendations for users. Based on user interaction information and social information, the graph neural network is introduced to extract the latent feature representation of users and items. In the process of learning user embedding vector representation, the multi-head attention mechanism is introduced to assign different weights to the trusted friends of the target user, which can increase the importance of friends with high influence so as to obtain the interest preference of the target user more accurately.

3. Background

3.1. Recommendation Model of Graph Neural Network

For the association between users and items, Figure 1a shows the first-order connectivity that exists between the two. In process of recommending items to the user u_1, the existing algorithms only learn the association between u_1 and i_1, i_2, i_3, while ignoring the possible impact of i_4 and i_5 on u_1 preferences. Figure 1b shows the association between user u_1 and other users who have common interaction records. According to the different paths from each node to u_1, the nodes related to u_1 are divided into three layers, and the algorithm can learn the influence of i_4 and i_5 on u_1 preferences. Therefore, the embedding representation of u_1 can be better obtained.

Based on the above analysis, in order to fully explore the high-order connectivity between users and items, we use graph neural network to model the recommendation process. The process is shown in Figure 2 (Take the link process of u_1 and i_4 in Figure 1 as an example). The process is mainly divided into three layers: the embedding layer, propagation layer, and rating prediction layer.

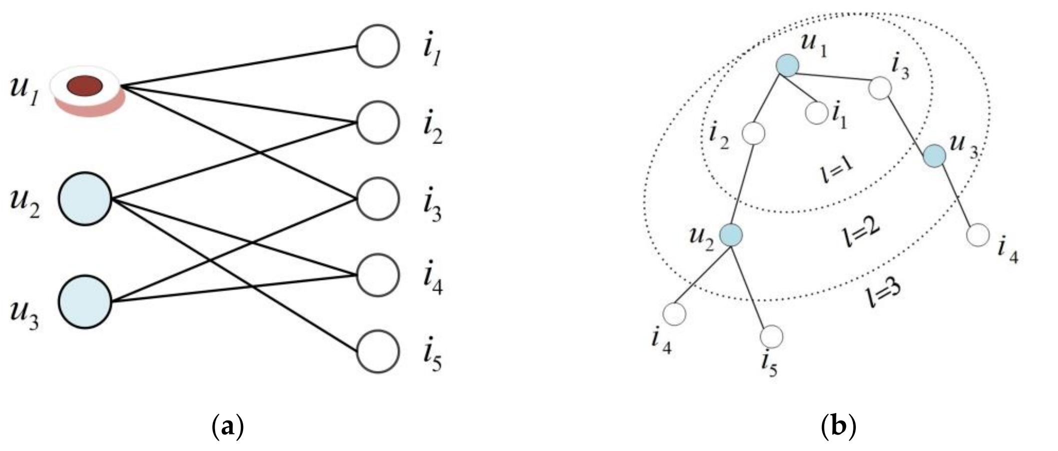


Figure 1. First-order connectivity versus higher-order connectivity. (**a**) User-item interaction graph; (**b**) Higher-order connectivity graph of u_1.

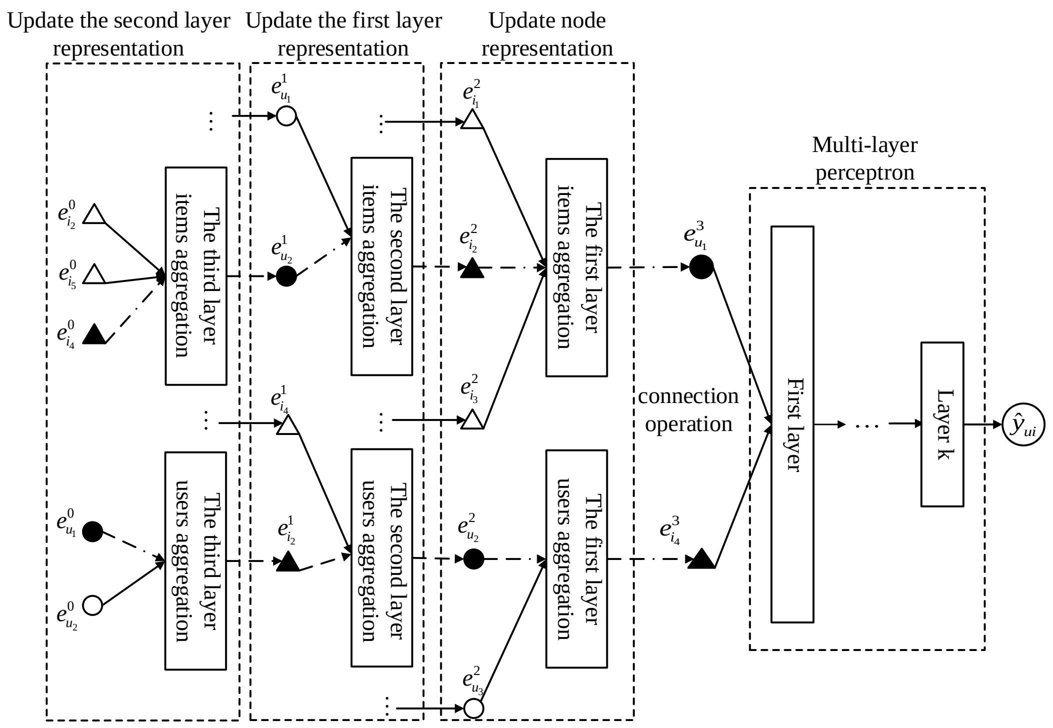

Update the second layer representation
Update the first layer representation
Update node representation
Multi-layer perceptron
The third layer items aggregation
The third layer users aggregation
The second layer items aggregation
The second layer users aggregation
The first layer items aggregation
The first layer users aggregation
connection operation
First layer
Layer k
$\hat{y}_{ui}$

Figure 2. The process of the graph neural network recommendation algorithm.

3.2. Multi-Head Attention Mechanism

The introduction of the attention mechanism [38,39] enables the model to focus on more important parts of numerous data sources, so it is widely used in natural language, machine translation, and other fields. With the publication of ‘Attention is all you need’ in the literature [40], researchers began to use the attention mechanism to improve social recommendation algorithms. During the process of the graph neural network aggregating the information of neighboring nodes, we introduced attention mechanism to learn the weight of each neighbor node and distinguish neighbor nodes depending on different weights. We only focused on neighbor nodes that had a greater impact on the target user node. Figure 3 shows the structure of the attention network layer, which is used to calculate the attention weight between node *i* and node *j*.

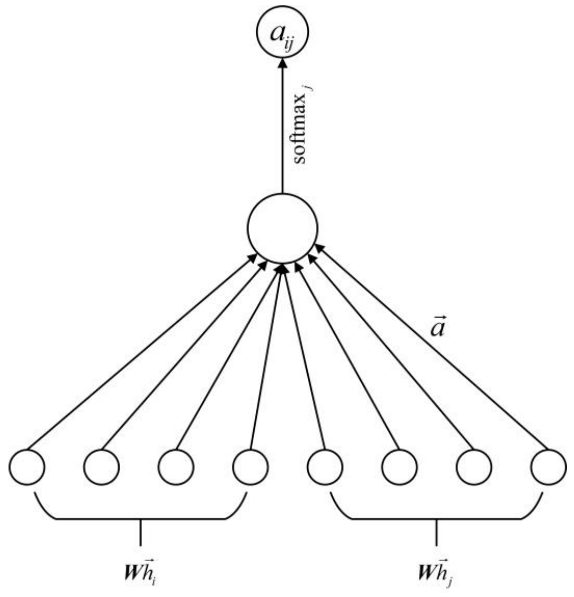


Figure 3. Representation of the attention layer.

The expression a_{ij} for the attention weight of node j to node i is Equation (1). W is a mapping matrix which can perform feature transformations on node i and node j. a^T is the weight parameter. $||$ is the stitching of vectors. *LeakyReLU* is an activation function.

$$a_{ij} = \frac{exp\left(LeakyReLU\left(a^T\left[Wh_i \parallel Wh_j\right]\right)\right)}{\sum_{k\in N_i} exp\left(LeakyReLU\left(a^T\left[Wh \parallel Wh_j\right]\right)\right)} \tag{1}$$

To highlight the influence of attention weights, we introduce the Multi-head Attention Mechanism (MAM) to improve the instability of the model learning process. We use P different functions to jointly compute the attention of the model. Each attention mechanism obtains a set of attention parameters, which can be spliced to get:

$$h_i'(P) = \mathop{\Big\|}_{p=1}^{P} \sigma\left(\sum_{j\in N_i} a_{ij}^p W^p h_j\right) \tag{2}$$

where a_{ij}^p denotes the attention weight between node i and node j learned by the pth attentional function.

4. Proposed Algorithm

4.1. Preliminaries

$U = \{u_1, u_2, \ldots, u_m\}$ and $I = \{i_1, i_2, \ldots, i_n\}$ are the sets of users and items, respectively, where m is the number of users and n is the number of items. $\boldsymbol{R}^{m\times n}$ is the rating matrix, also known as the user-item graph, and we can construct the user-item interaction graph according to the records of user-item interactions. $N(i)$ is the set of users directly connected by u_i, and $C(i)$ is the set of items that has interacted with u_i. In addition, there are also social relationships between users, and the user–user social relationship graph is denoted by T. If there is a relationship between u_i and u_j, $T_{ij} = 1$, otherwise $T_{ij} = 0$. Given the user-item interaction graph and social network graph, the core aim of this paper is to train a model that can learn user and latent item embedding representations based on user-item interactions and social trust relationships and then perform rating predictions.

4.2. User Embedding Vector Representation Integrating Interaction Information and Social Information

The embedding layer in the recommendation model of the graph neural network given in Section 3.1 mainly includes user embedding and item embedding. The learning process of user embedding vector representation is divided into two parts. One is based on the user-item graph and the other is based on the social network graph combined with the multi-head attention mechanism. We fuse the user embedding representation learned from the two parts to obtain the final user embedding representation.

The initial embedding representations of users, items, and friends of users are obtained based on the user-item interaction graph and social network graph. The equations are as follows:

$$e_u = [e_{u_1}, e_{u_2}, \ldots, e_{u_m}] \tag{3}$$

$$e_i = [e_{i_1}, e_{i_2}, \ldots, e_{i_n}] \tag{4}$$

$$e_{u_{(i,j)}} = \left[e_{u_{(i,1)}}, e_{u_{(i,2)}}, \ldots, e_{u_{(i,b)}}\right] \tag{5}$$

where e_u denotes the initial embedding vector representation of the target user, e_i denotes the initial embedding vector representation of the item, and $e_{u_{(i,j)}}$ represents the initial embedding vector representation of the trusted friend of the target user.

Based on the above three initial embedding vector representations, the final user embedding vector representation is shown below.

(1) User embedding vector representation based on the user-item interaction graph

We learn the user embedding representation by aggregating their neighbors' information. On the one hand, we learn the influence of items on user preferences based on the rating information. On the other hand, according to the transmissibility of information, the user embedding representation can be refined at the propagation layer through a multi-layer neural network which can more effectively reflect the potential association between users and items. The learning process is described below.

The higher the user's rating for the item, the better the rating can represent the user's characteristics. In other words, the item is more in line with the user's preferences. r_{ui} represents the rating of user u on item i, and $r_{ui} \in [0, 5]$. $r_{ui} = 5$ shows that item i exactly matches the preferences of user u, and the vector representation of item i can be directly used to characterize the vector representation of user u. $r_{ui} = 0$ shows that item i is opposite of the preferences of user u, and the vector representation of item i cannot be used to characterize the vector representation of user u. Based on the above ideas, we define rating weights to measure the extent to which the embedding representation of user u is characterized by the embedding representation of item i, as shown in Equation (6).

$$a_{ui} = \frac{r_{ui}}{5} \tag{6}$$

where a_{ui} represents the rating weight, and the range of value is $0 \sim 1$.

According to Equations (4) and (6) mentioned above, the user's neighbor node representation is aggregated according to the user-item interaction graph. The user embedding vector representation is shown below.

$$e_{u \leftarrow i} = \sum_{i \in N_u} p_{ui} a_{ui} e_i \tag{7}$$

$$p_{ui} = \frac{1}{\sqrt{|N_u||N_i|}} \tag{8}$$

where N_u and N_i represent the neighbor nodes' set of users and items, respectively. $|N_u|$ and $|N_i|$ represent the number of neighbor nodes of users and items, respectively. $i \in N_u$ represents the neighbor node information of the user, that is, the item node representation. p_{ui} represents the weight coefficient of users and items, that is, the normalization coefficient. Because the number of neighbor nodes of each user is different, if the neighbor node information is aggregated directly, the data will be unstable. Therefore, we need to normalize the neighbor node.

However, Equation (7) only uses the item aggregation of the user's first-order neighbors, and the obtained user representation is not comprehensive enough. We also need to incorporate the influence of users who have interacted with this user on its representation to utilize the higher-order connectivity of users proposed in Figure 1b. Therefore, according to the learning process of the algorithm in Figure 2, we obtain the information representation of the user's third-order neighbors through the 3-layer information embedding and propagation process, and thus learn user embedding representation. Then, suppose that after the information propagation of the l layer, the user embedding of the l layer is expressed as:

$$e^{l}_{u\leftarrow i} = \sum_{i\in N_u} p_{ui} a_{ui} e^{l-1}_{i} \tag{9}$$

where $e^{l}_{u\leftarrow i}$ denotes user embedding vector representation aggregated by neighbor items at layer l, e^{l-1}_{i} denotes item embedding vector representation at layer $l-1$, and p_{ui} denotes the weighting coefficient. In the process of information propagation, the corresponding user embedding vector representation $e^{0}_{u\leftarrow i}, e^{1}_{u\leftarrow i}, \ldots, e^{l}_{u\leftarrow i}$ will be learned at each layer. Since the state updates of each layer contain rich information and are different, we cannot simply apply the highest-order vector to represent the user embedding. We need to aggregate the representation of status updates obtained at each layer to get the user embedding vector representation.

$$e_{u\leftarrow i} = f\left(e^{0}_{u\leftarrow i}, e^{1}_{u\leftarrow i}, \ldots, e^{l}_{u\leftarrow i}\right) \tag{10}$$

where $f(\cdot)$ denotes the aggregation function and we use the mean aggregation function to aggregate neighbor information representation. This not only ensures that the model learns more data information, but also makes it simpler and more efficient. The final user embedding representation vector is show as:

$$e_{u\leftarrow i} = avg\left(e^{0}_{u\leftarrow i}, e^{1}_{u\leftarrow i}, \ldots, e^{l}_{u\leftarrow i}\right) \tag{11}$$

According to the above ideas, the user embedding vector representation based on user-item interaction graph is described in Algorithm 1.

Algorithm 1 User embedding vector representation based on the user-item interaction graph

```
Input: user-item interaction graph $\boldsymbol{R} : U \times I$, user rating $r_{ui} \in \boldsymbol{R}$
Output: user embedded representation $e_{u\leftarrow i}(u \in U)$
Begin
1 For $l = 1, \ldots, L$ do
2     $a_{ui} \leftarrow \frac{r_{ui}}{5}$
3     For $u \in U$ do
4             $e^{l}_{u\leftarrow i} \leftarrow \sum_{i\in N_u} p_{ui} a_{ui} e^{l-1}_{i}$
5     End
6 End
7 $e_{u\leftarrow i} \leftarrow avg\left\{e^{0}_{u\leftarrow i}, e^{1}_{u\leftarrow i}, \ldots, e^{l}_{u\leftarrow i}\right\}$
End
```

Line 1 sets the number of information aggregation layers of the graph neural network model. Line 2 computes the rating weights based on user ratings. Lines 3 to 5 take into account the weight coefficients of the number of neighbors and the rating weights and then aggregate the neighbor item information of the user to learn the user embedding vector representation of the layer l. Line 7 uses the mean aggregation function to obtain the final user embedding vector representation.

(2) User embedding vector representation with MAM based on the social network graph

From the perspective of the social network graph, this section uses graph neural network to deeply study user embedding representation. In the learning process, the multi-head attention mechanism is introduced into the information aggregation process to model the trust relationship between users and their friends and increase the importance of high-impact friends.

u_i denotes the user waiting for the recommended items, that is, the target user. According to Equations (3) and (5), the initial embedding representation of the user u_i and his trusted friends is obtained as e_{u_i} and $e_{u_{(i,j)}}$. We fuse e_{u_i} and $e_{u_{(i,j)}}$ to obtain the joint embedding vector s of the user u_i and his friend as shown below.

$$s = s_1 * e_{u_i} + (1 - s_1) * e_{u_{(i,j)}} \tag{12}$$

$$s_1 = tanh(w_1 * e_{u_i} + w_2 * e_{u_{(i,j)}} + b) \tag{13}$$

where w_1, w_2, and b are the weight parameters and bias respectively. Then, the multi-head attention vector a is calculated as follows:

$$a_t = \frac{exp(a_t^*)}{\sum_k exp(a_k^*)} \tag{14}$$

$$a_t^* = s^T K_t \tag{15}$$

where a_t^* is an element of the multi-head attention vector a, which indicates the degree of influence of the users' friends on the user in a certain aspect. The attention vector a is learned from the key matrix K. $K_t \in \mathbb{R}^d$, where d is the dimension of user and item embedding representation. s^T is the transpose of the joint embedding vector.

Next, the multi-head attention parameters are fused into the embedding vector representation of each layer, and the weighted sum is used to obtain the relationship vector representation of the user's friends.

$$f_{(i,j)} = \sum_t a_t \Big(e_{u_{(i,j)}} \odot M_t \Big) \tag{16}$$

$f_{(i,j)}$ denotes the vector that shows the j-th friend of user i influences user preferences. We take $f_{(i,j)}$ as an input and embed it into different header attention layers, as shown in Equation (17). The importance of friends is distinguished by comparing the weights. The greater the weight, the greater the influence on the user.

$$F_{(i.j)} = Multi - head - attention\Big(f_{(i,j)}\Big) \tag{17}$$

Then, Equation (17) and the user's preferences are fused to obtain the user embedding vector representation $e_{u \leftarrow s}$.

$$e_{u \leftarrow s} = G * e_{u_i} + (1 - G) * F_{(i,j)} \tag{18}$$

$$G = sigmoid\Big(w_1 e_{u_i} + w_2 F_{(i,j)} + b\Big) \tag{19}$$

where w_1 and w_2 are weight parameters, $e_{u \leftarrow s}$ denotes user embedding vector representation, and G denotes a gating layer.

(3) User embedding vector representation UEV_IS

We fuse the user embedding vector representation learned from the above to obtain User Embedding Vector Representation Integrating Interaction Information and Social

Information (UEV_IS), which integrates interaction information and social information. The calculation equation is as follows:

$$e_u = e_{u \leftarrow i} \oplus e_{u \leftarrow s} \tag{20}$$

According to the above ideas, the final user embedding vector representation UEV_IS is described in Algorithm 2.

Algorithm 2 User embedding vector representation UEV_IS

```
Inputs: user-item interaction graph $\boldsymbol{R} : U \times I$, user social network $T$
Output: user-embedded vector representation $e_u$
Begin
1 $e^0_{u\leftarrow i}, e^0_{u\leftarrow s}\ \forall u \in U$
2 For $l = 1, 2, \ldots, L$ do
3     For $u \in U$ do
4         $e^l_{u\leftarrow i} \leftarrow \sum_{i \in N_u} p_{ui} a_{ui} e^{l-1}_i$
5     End
6     For $u_i \in T$ do
7         $s \leftarrow s_1 * e_{u_i} + (1 - s_1) * e_{u_{(i,j)}}$
8         $a^*_t = s^T K_t$
9         $a_t \leftarrow \frac{exp(a^*_t)}{\sum_k exp(a^*_k)}$
10    End
11 End
12 $e_{u\leftarrow i} \leftarrow avg\left\{e^0_{u\leftarrow i}, e^1_{u\leftarrow i}, \ldots, e^l_{u\leftarrow i}\right\}$
13 $e_{u\leftarrow s} \leftarrow G * e_{u_i} + (1 - G) * F_{(i,j)}$
14 $e_u \leftarrow e_{u\leftarrow i} \oplus e_{u\leftarrow s}$
End
```

Line 1 randomly initializes to obtain the initial user embedding representation. Lines 2 to 5 obtain the user embedding representation of the l layer according to the user-item interaction graph. Lines 6–10 consider the importance of user's friends influence on them, introduce the multi-head attention mechanism, and learn user feature representation. Line 11 obtains the user embedding representation of aggregated item information. Line 12 gets the user embedding representation based on the influence of the user's friends in the social network. Line 13 fuses the user embedding vector obtained from the user-item graph and social relationship network to get the final user embedding vector representation.

4.3. Item Embedding Vector Representation Based on the User-Item Interaction Graph

The item embedding vector representation based on the user-item interaction graph is proposed. Based on the user-item interaction graph, the user information that interacts with the item is aggregated to obtain the embedding vector representation of the item. Figure 4 shows the specific item embedding model.

The item embedding model is the same as the user embedding model representation based on the user-item interactions in Section 4.2, which mainly learns to obtain item embedding vector representation by aggregating information from an item's neighbor nodes. Accounting for the differences in ratings of the same item by different users, we introduce rating weights to get the item embedding vector representation in Equation (21).

$$e_{i \leftarrow u} = \sum_{u \in N_i} p_{ui} a_{ui} e_u \tag{21}$$

where $u \in N_i$ denotes item neighbor node information, that is, user node representation. p_{ui} denotes the weighting coefficient of the user and item, and a_{ui} denotes the rating weight.

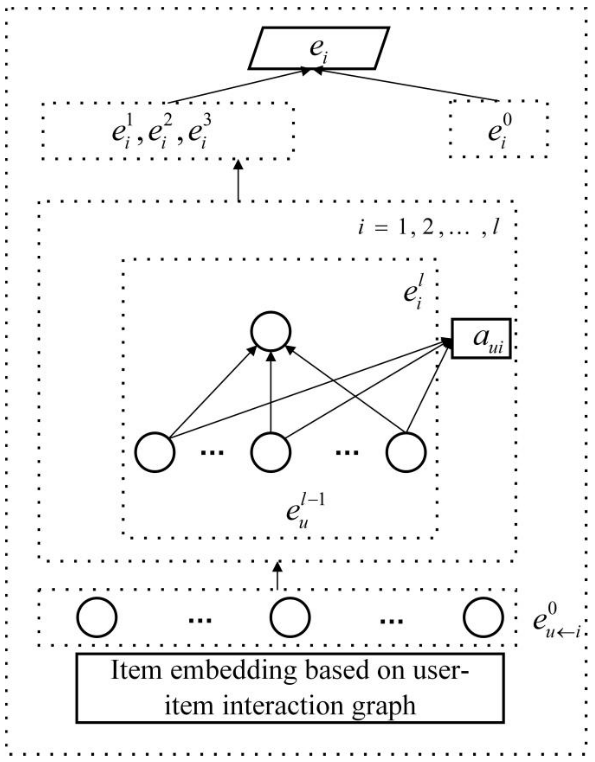


Figure 4. Item embedding model based on the user-item interaction graph.

Equation (21) is the same as Equation (7), as both are representations of first-order neighbor nodes. Due to the latent connection between items and their second-order neighborhoods, user aggregation using only the first-order neighborhoods of items cannot fully represent items. Therefore, when learning item embedding representations, we should also consider the higher-order connectivity of items to capture more accurate information. Therefore, Equation (21) can be further improved:

$$e_{i\leftarrow u}^{l} = \sum_{u\in N_i} p_{ui} a_{ui} e_u^{l-1} \tag{22}$$

where $e_{i\leftarrow u}^{l}$ denotes the item embedding vector representation of layer l neighbor user aggregation. e_u^{l-1} denotes user embedding vector representation of layer $l-1$. Similar to the process described in Section 4.2, after the information propagation of layer l, the aggregated item embedding vector representation is shown as:

$$e_i = avg\left(e_{i\leftarrow u}^{0}, e_{i\leftarrow u}^{1}, \ldots e_{i\leftarrow u}^{l}\right) \tag{23}$$

Similar to the user embedding vector representation learned based on the user-item interaction graph, we select a mean aggregation for aggregation operation to obtain the final item embedding representation.

After the above steps, we learn the relationship between users and items based on the operation of information dissemination and aggregation. According to Equation (20) and Equation (23), the embedding representations e_u and e_i of users and items are further obtained and the user's preference for the items is predicted through the rating prediction layer.

$$\hat{y}_{ui} = e_u^T e_i \tag{24}$$

In order to make the model better reflect the characteristics of users, items, and trusted friends, it is necessary to optimize the higher-order feature vectors, weights, and deviations of the model. Considering the relative order of interactions between users and items that can be observed and those that cannot be observed, this paper adopts Bayesian Personalized Ranking (BPR) loss [41], which is widely used in recommendation systems to define the loss calculation method as follows:

$$Loss = \sum_{(u,a,b)\in D} -ln\sigma(P_{ua} - P_{ub}) + \lambda\|\Theta\|_2^2 \tag{25}$$

$$D = \left\{(u,a,b)\middle|(u,a) \in R^+, (u,b) \in R^-\right\} \tag{26}$$

where D expresses paired training data, R^+ denotes observed interaction terms, and R^- denotes unobserved interaction terms. σ denotes the $sigmod()$ function and $\lambda\|\Theta\|_2^2$ denotes the regular term. Θ denotes trainable parameters included in the model, which can control the parameter size and avoid over-fitting. In addition, we use the Adam optimizer [42] to adjust model parameters and minimize the loss function.

4.4. Algorithm GNNSR_MAM

This paper introduces graph neural network to mine latent associations between users and items on the basis of collaborative filtering [28] recommendation algorithms. In the process of user–neighbor aggregation, we introduce a multi-head attention mechanism based on social network relationships to distinguish the importance of the user's trustworthy friends. At the same time, we comprehensively consider the different influences of items and friends on users and propose a Graph Neural Network Social Recommendation Algorithm Integrating Multi-Head Attention Mechanism (GNNSR_MAM). The algorithm consists of three main parts: data preprocessing, graph neural network modeling, and multi-layer perceptron (MLP) prediction rating. The overall framework of the algorithm is shown in Figure 5.

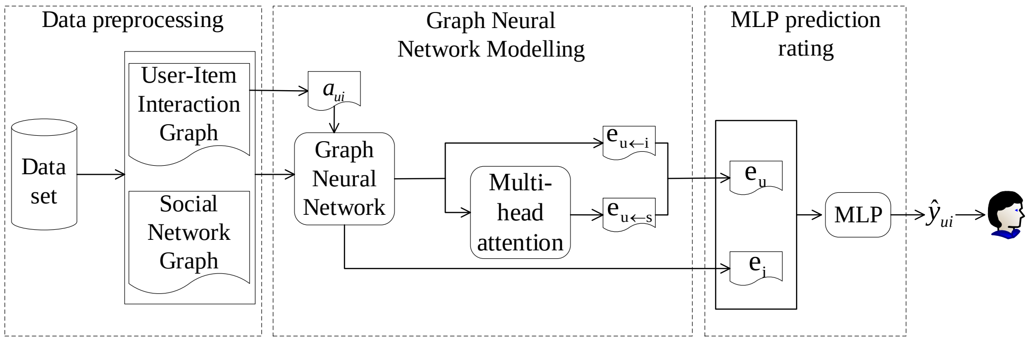


Figure 5. Framework of algorithm GNNSR_MAM.

First, the data is preprocessed to obtain the user-item interaction graph and social network diagram. Then, based on the preprocessed data, the graph neural network is used to learn the embedding vector representation of users and items. In this process, user embedding vector representation $e_{u\leftarrow i}$ and item embedding vector representation e_i are obtained based on user-item interaction diagram learning. Based on the social network graph, the multi-head attention mechanism is introduced into the aggregation process of friends to obtain the embedding vector representation of users $e_{u\leftarrow s}$ from another perspective. The final user embedding vector representation is obtained by combining the user embedding vector representation $e_{u\leftarrow i}$ and $e_{u\leftarrow s}$. Finally, based on the obtained embedding vectors of users and items, e_u and e_i are used to predict the rating.

According to the above ideas, the graph neural network social recommendation algorithm integrating multi-head attention mechanism is described in Algorithm 3.

Algorithm 3 Algorithm GNNSR_MAM

```
Inputs: user-item interaction graph $\boldsymbol{R} : U \times I$, social network graph $T$ and user rating data $r_{ui}$,
embedding dimension $d$, regularization coefficient $\lambda$, number of propagation layers $l$
Output: predicted rate $\hat{y}_{ui}$
Begin
1 $d \leftarrow 64,\ \lambda \leftarrow 10^{-5},\ l \leftarrow 3$
2 $a_{ui} \leftarrow \frac{r_{ui}}{5}$
3 For $l = 1$ to $3$ do
4     For $i \in I$ do
5         $n^l_{i\leftarrow u} \leftarrow Aggregator\left(e^l_i, e^l_u | u \in N_i\right)$
6         $e^l_{i\leftarrow u} \leftarrow Update\left(e^l_i, n^l_i\right)$
7     End
8     For $u \in U$ do
9         $n^l_{u\leftarrow i} \leftarrow Aggregator\left(e^l_u, e^l_i | i \in N_u\right)$
10        $e^l_{u\leftarrow i} \leftarrow Update\left(e^l_u, n^l_u\right)$
11    End
12    $\left[e^l_{u\leftarrow i}, e^l_{i\leftarrow u}\right] \leftarrow GNN(R, a_{ui})$
13 End
14 $e_i \leftarrow avg\left\{e^l_{i\leftarrow u}, l \in \{1,2,3\}\right\}$
15 $e_{u\leftarrow i} \leftarrow avg\left\{e^l_{u\leftarrow i}, l \in \{1,2,3\}\right\}$
16 For $u_i \in T$ do
17    $F_{(i,j)} \leftarrow Multi - Head - Attention\left(e_{u_i}, e_{u_{(i,j)}}\right)$
18    $e_{u\leftarrow s} \leftarrow GNN\left(T, F_{(i,j)}\right)$
19 End
20 $e_u \leftarrow concat(e_{u\leftarrow i}, e_{u\leftarrow s})$
21 $\hat{y}_{ui} \leftarrow e^T_u e_i$
22 Return $\hat{y}_{ui}$
End
```

Lines 1 to 2 give the embedding vector dimension, number of information propagation layers, regularization coefficient, and rating weight. Lines 3 to 15 use the graph neural network to learn the embedding vector representation of users and items based on the user-item interaction diagram and rating weight. Lines 16 to 19 introduce a multi-head attention mechanism into the aggregation process of users and their friends to learn the embedding vector representation of users based on the social network graph. Line 20 comprehensively considers the influence of items and friends and gets the final embedding vector representation of users. Lines 21 to 22 return the prediction rating based on the embedding vector representation of users and items.

5. Experimental Results and Analysis

5.1. Dataset

In order to evaluate the performance of algorithm GNNSR_MAM, we select the Epinions dataset provided by the site http://www.trustlet.org/downloaded_epinions.html (accessed on 4 July 2022) [43], which is commonly used in social recommendation algorithms. The dataset mainly contains user-item rating information and social trust information among users. All ratings are integer values between 1 and 5 and each item is rated at least once. The rating sparsity of the dataset is 99.9%. We divide the dataset into three groups: 70% are used as the training set, 20% are used as the validation set, and the remaining 10% as the test set. The statistics of the experimental dataset are shown in Table 1.

Table 1. Statistics of the experimental dataset.

Dataset	Epinions
Number of users	49,290
Number of items	139,738
Number of ratings	664,824
Social links	487,181

5.2. Evaluation Metrics

We use Recall and Normalized Discounted Cumulative Gain (NDCG) to evaluate the performance of our proposed algorithm.

Recall is used to calculate the ratio of the number of items recommended to users to the number of movies users as shown in the test set. The higher the value of Recall, the better the recommendation effect of the model. NDCG conducts a comprehensive recommendation quality evaluation on the user recommendation list. The higher the value of NDCG, the better the recommendation effect. The calculation equation is as follows:

$$Recall@K = \frac{\sum_{j=1}^{K} rel_j}{min(K, |y_u^{test}|)} \tag{27}$$

$$NDCG@K = \frac{DCG@K}{IDCG@K} \tag{28}$$

$$DCG@K = \sum_{j=1}^{K} \frac{2^{rel_j} - 1}{log_2(j+1)} \tag{29}$$

where K denotes the number of items in the recommendation list. rel_j denotes whether the jth item in the recommendation list is liked by the user and if it is true, $rel_j = 1$, otherwise, $rel_j = 0$. y_u^{test} denotes the number of items in the test set that have been rated by users. IDCG denotes the best recommendation result obtained by ranking the user recommendation list. In other words, it refers to the maximum Discounted Cumulative Gain achieved through ranking. This paper uses the Top-K recommendation and set $K = \{10, 30, 50\}$ to evaluate the performance of the algorithm.

5.3. Model Parameter Analysis

In the graph neural network model, we consider the high-order connectivity between users and items and capture the association between users and items by aggregating neighbor information. In this process, the selection of the number of the information propagation layer when aggregating higher-order neighbor information is a key factor affecting the model effect. When the number of the information propagation layer are too few, the model cannot effectively capture rich neighbor information in the interaction graph. On the contrary, noise data will be introduced, and over-fitting will occur, which will affect the recommendation performance of the algorithm. Aiming to select the optimal number of the information propagation layer to ensure model effectiveness, we conducted the following experiments on this parameter.

Figure 6 shows the effect of different values on GNNSR_MAM. When $l \leq 3$, with the increase of l, the Recall and NDCG of the algorithm are significantly improved. The optimal value is obtained when $l = 3$, and the performance of the algorithm is the best, which shows that considering the user-item high-order connectivity is effective for the algorithm. Whereas, when $l = 4$, the Recall and NDCG decreased, which indicates over-fitting of the model. In summary, to keep algorithm performance optimal, we set the model propagation layer as $l = 3$. The main parameters of the experiment are set as shown in Table 2.

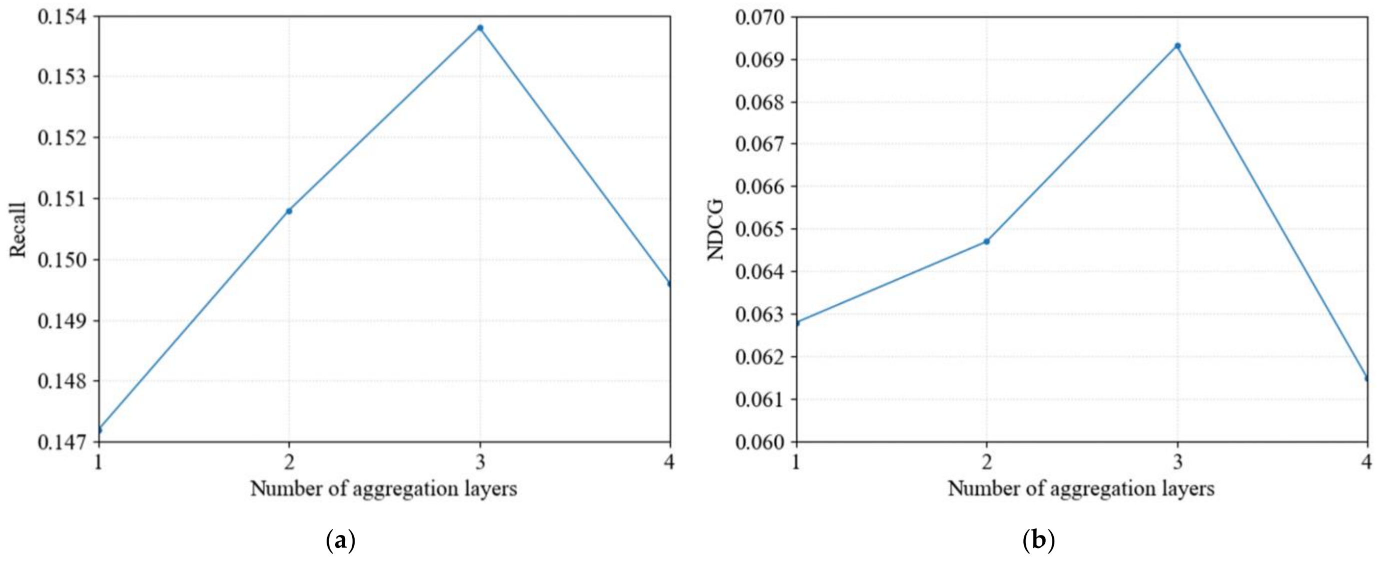


Figure 6. Effects of different aggregation layers l on the algorithm. (**a**) Effect of l on Recall; (**b**) Effect of l on NDCG.

Table 2. Experimental parameter settings.

Parameter	Parameter Value
User and item embedding dimension	$d = 64$
Regularization coefficient	$\lambda = 10^{-5}$
Batch size	128
Learning rate	0.01
Optimizer	Adam

5.4. *Performance Analysis*

5.4.1. Comparative Analysis of Mainstream Algorithms

In order to verify the effectiveness of the proposed algorithm, we compare GNNSR_MAM with the following three main algorithms and the experimental results are shown in Figure 7.

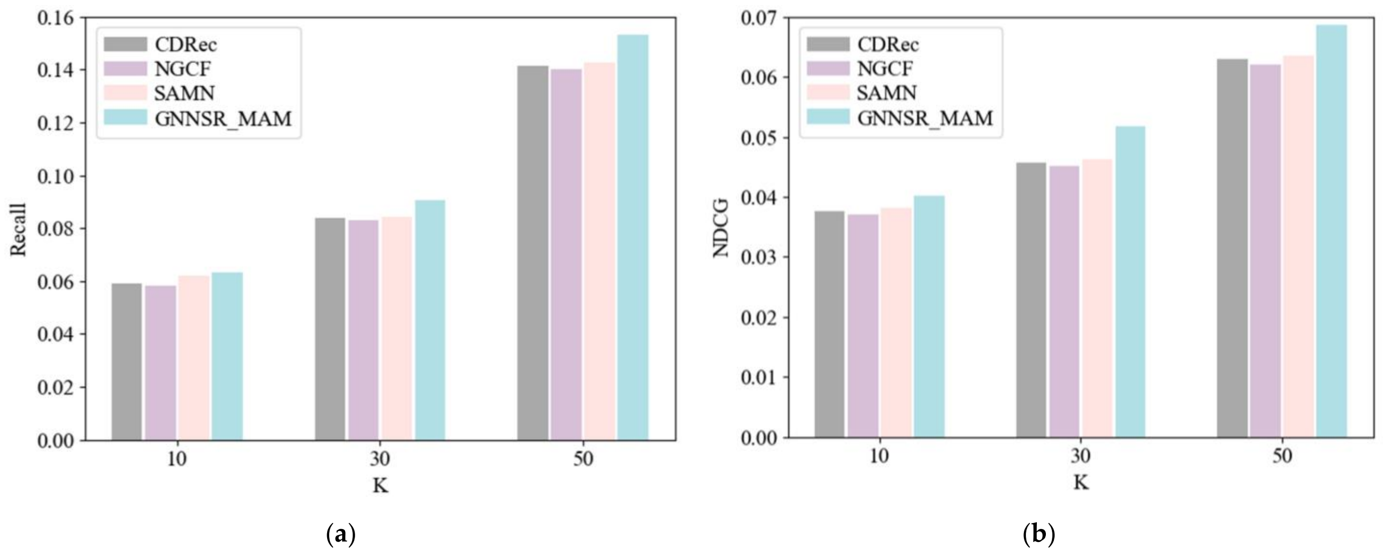


Figure 7. Comparison of the algorithms on the Epinions dataset. (**a**) Recall@K; (**b**) NDCG@K.

1. CDRec [25]: A collaborative filtering recommendation algorithm based on a multi-relational social network. The algorithm divides the community structure based on trust relationships and rating information to improve the recommendation efficiency.
2. NGCF [44]: A collaborative filtering recommendation algorithm based on the graph neural network. This algorithm applies GCN to the recommendation algorithm and obtains the prediction rating by extracting the embedding relationship between users and items.
3. SAMN [45]: Social recommendation algorithm based on attentional memory networks. This algorithm uses the attention mechanism to distinguish the importance of friends in social recommendation.

Figure 7 shows that our proposed GNNSR_MAM algorithm outperforms the other three recommendation algorithms in both evaluation metrics on the Epinions dataset. This reflects the positive impact of introducing both social relationships and graph neural network on the recommendation performance.

For SAMN and CDRec, the performance of SAMN is better than that of CDRec when K takes on three different values. The Recall@10 of SAMN increased by 5.25% compared with CDRec. The reason for this is that SAMN introduces an attention mechanism during neighbor aggregation. This shows that the model is not enough to rely on a single social relationship, but also needs to consider the impact of rich social relationships on users to distinguish the importance of friends.

The recommendation efficiency of GNNSR_MAM is better than that of SAMN, especially the value of NDCG@30, which improves by 12.12%. The main reason is that GNNSR_MAM uses the graph neural network while learning the association between users and items and considers the different effects of friends on target users in social relationships. At the same time, the multi-head attention mechanism is used to obtain more trusted friends of users, and the importance of different users can be obtained. Thus, GNNSR_MAM not only alleviates data sparsity by using social information, but can also learn the user embedding representation from the social relationship graph and enhance user final embedding representation. In addition, by considering the influence of non-linear characteristics of users and items based on the graph neural network, the recommendation performance of the system is improved.

It can be seen that our social recommendation algorithm GNNSR_MAM is superior to other algorithms, which verifies the effectiveness of our proposed algorithm.

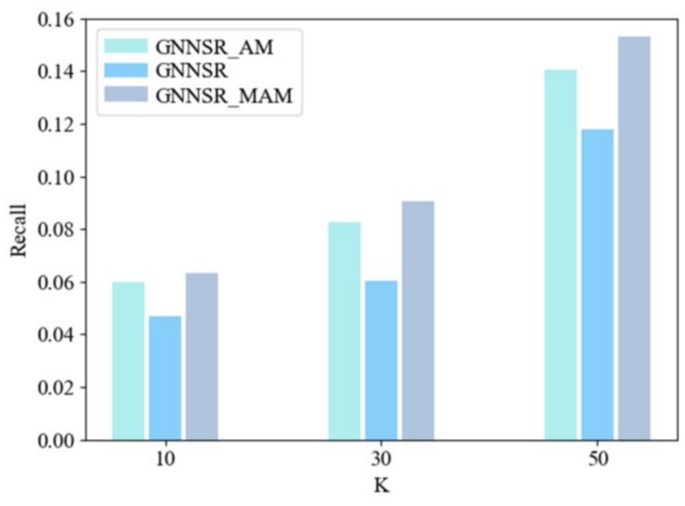
5.4.2. Comparative Analysis of Attention Mechanisms

To further illustrate the effectiveness of our proposed algorithm and verify the impact that the multi-head attention mechanism brings to the algorithm, GNNSR_MAM is compared with the following two algorithms in experiments:

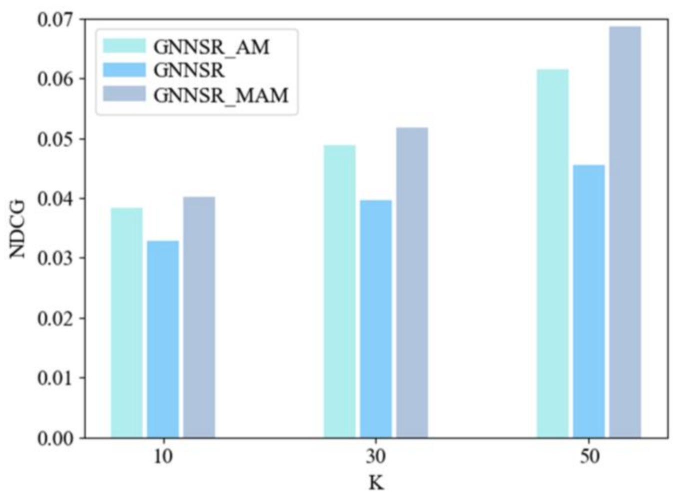
1. The algorithm GNNSR_AM is obtained by replacing the multi-head attention mechanism adopted by our proposed algorithm with the attention mechanism.
2. The algorithm GNNSR is obtained by removing the multi-head attention mechanism adopted by our proposed algorithm.

Figure 8 shows that GNNSR_MAM exhibits the best performance in terms of Recall and NDCG on the Epinions dataset. Compared with GNNSR, the Recall of GNNSR_AM increases by at least 19.24%, and the NDCG increases by at least 16.77%. It reflects that GNNSR_AM uses the network structure of attention mechanism in the learning process so it can better obtain the user representation.

Compared with GNNSR, the Recall of GNNSR_MAM increases by 43.9%, and the NDCG increases by 50.66%. This shows that considering the different influences of different friends on users and distinguishing the importance of friends' influence can help improve the interpretability of recommendation results. In addition, GNNSR_MAM shows a small improvement in recommendation effect compared to GNNSR_AM, which indicates that introducing multiple attention functions in the process of user neighbor aggregation can better learn user embedding representations and improve model recommendation quality.



(**a**)



(**b**)

Figure 8. Impact of different attention mechanisms on the algorithm. (**a**) Recall@K; (**b**) NDCG@K.

6. Conclusions

In this work, we designed a graph neural network social recommendation algorithm integrating multi-head attention mechanism to alleviate the data sparsity problem. Firstly, the graph neural network is used to model the user-item interaction relationship and user-trust relationship. Then, the multi-head attention mechanism is introduced to fully consider the trust degree of users to their neighbors, distinguish the importance of different users, and mine better embedding representations to improve the accuracy of recommendations. Experimental results show that the proposed algorithm can effectively improve the recommendation quality. We only use user-item interaction graphs when calculating item similarity, but latent information such as category descriptions related to items can also affect the similarity between items. In the future, we will explore the potential links between the data and further investigate the impact of feature fusion on recommendation models.

Author Contributions: Conceptualization, H.Y. and J.L.; methodology, H.Y.; software, H.Y.; validation, W.X., X.L. and H.Q.; formal analysis, J.L.; investigation, H.Y.; resources, H.Y.; data curation, J.L.; writing—original draft preparation, H.Y.; writing—review and editing, J.L.; supervision, H.Y.; project administration, W.X.; funding acquisition, H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by the Applied Foundation Research Project of Liaoning Province (No. 2022JH2/101300278), the Foundation Research Project of the Educational Department of Liaoning Province (No. LJKZZ20220085), and the Cooperation Innovation Plan of Yingkou for Enterprise and Doctor (No. 2022-13).

Data Availability Statement: Our training set Epinions dataset is publicly available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gosh, S.; Nahar, N.; Wahab, M.A.; Biswas, M.; Hossain, M.S.; Andersson, K. Recommendation System for E-commerce Using Alternating Least Squares (ALS) on Apache Spark. In Proceedings of the Intelligent Computing and Optimization, Hua Hin, Thailand, 8–9 October 2020; pp. 880–893.
2. Park, S.-J.; Kang, C.-U.; Byun, Y.-C. Extreme Gradient Boosting for Recommendation System by Transforming Product Classification into Regression Based on Multi-Dimensional Word2Vec. *Symmetry* **2021**, *13*, 758. [CrossRef]
3. Urdaneta-Ponte, M.C.; Mendez-Zorrilla, A.; Oleagordia-Ruiz, I. Recommendation Systems for Education: Systematic Review. *Electronics* **2021**, *10*, 1161. [CrossRef]
4. Roy, D.; Dutta, M. A systematic review and research perspective on recommender systems. *J. Big Data* **2022**, *9*, 59. [CrossRef]
5. Gu, L.; Yang, P.; Dong, Y. An dynamic-weighted collaborative filtering approach to address sparsity and adaptivity issues. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 3044–3050.

6. Koren, Y.; Bell, R. Advances in Collaborative Filtering. In *Recommender Systems Handbook*; Ricci, F., Rokach, L., Shapira, B., Eds.; Springer: Boston, MA, USA, 2015; pp. 77–118.
7. Salah, A.; Rogovschi, N.; Nadif, M. A dynamic collaborative filtering system via a weighted clustering approach. *Neurocomputing* **2016**, *175*, 206–215. [CrossRef]
8. Polatidis, N.; Georgiadis, C.K. A multi-level collaborative filtering method that improves recommendations. *Expert Syst. Appl.* **2016**, *48*, 100–110. [CrossRef]
9. Zhang, J.; Tang, J.; Liang, B.; Yang, Z.; Wang, S.; Zuo, J.; Li, J. Recommendation over a Heterogeneous Social Network. In Proceedings of the 2008 The Ninth International Conference on Web-Age Information Management, Zhangjiajie, China, 20–22 July 2008; pp. 309–316.
10. Qian, X.; Feng, H.; Zhao, G.; Mei, T. Personalized Recommendation Combining User Interest and Social Circle. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 1763–1777. [CrossRef]
11. Ma, H.; Jia, M.; Zhang, D.; Lin, X. Combining Tag Correlation and User Social Relation for Microblog Recommendation. *Inf. Sci.* **2017**, *385*, 325–337. [CrossRef]
12. Gong, C.; Sun, G.; Chen, C.-C.; Bin, S. Matrix Decomposition Recommendation Algorithm Based on Multiple Social Relationships. In Proceedings of the 2020 IEEE 2nd Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability (ECBIOS), Tainan, Taiwan, 29–31 May 2020; pp. 197–200.
13. Xin, M.; Chen, S.; Zang, C. A Graph Neural Network-Based Algorithm for Point-of-Interest Recommendation Using Social Relation and Time Series. *Int. J. Web Serv. Res.* **2021**, *18*, 51–74. [CrossRef]
14. Ahmadian, S.; Joorabloo, N.; Jalili, M.; Meghdadi, M.; Afsharchi, M.; Ren, Y. A temporal clustering approach for social recommender systems. In Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Barcelona, Spain, 28–31 August 2018; pp. 1139–1144.
15. Yu, J.; Yin, H.; Li, J.; Gao, M.; Huang, Z.; Cui, L. Enhancing Social Recommendation With Adversarial Graph Convolutional Networks. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 3727–3739. [CrossRef]
16. Jamali, M.; Ester, M. A matrix factorization technique with trust propagation for recommendation in social networks. In Proceedings of the RecSys'10: Fourth ACM Conference on Recommender Systems, Barcelona, Spain, 26–30 September 2010; pp. 135–142.
17. Ma, H.; Zhou, D.; Liu, C.; Lyu, M.R.; King, I. Recommender systems with social regularization. In Proceedings of the WSDM'11: Fourth ACM International Conference on Web Search and Data Mining, Hong Kong, China, 9–12 February 2011; pp. 287–296.
18. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **2020**, *1*, 57–81. [CrossRef]
19. Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W.L.; Leskovec, J. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. *arXiv* **2018**, arXiv:1806.01973.
20. Pletnev, A.; Rivera-Castro, R.; Burnaev, E. Graph Neural Networks for Model Recommendation using Time Series Data. *arXiv* **2020**, arXiv:2009.03474.
21. Huang, B.; Bi, Y.; Wu, Z.; Wang, J.; Xiao, J. UBER-GNN: A User-Based Embeddings Recommendation based on Graph Neural Networks. *arXiv* **2020**, arXiv:2008.02546.
22. Wang, G.; Guo, Z.; Li, X.; Yin, D.; Ma, S. SceneRec: Scene-Based Graph Neural Networks for Recommender Systems. *arXiv* **2021**, arXiv:2102.06401.
23. Ma, H.; Yang, H.; Lyu, M.R.; King, I. SoRec: Social recommendation using probabilistic matrix factorization. In Proceedings of the CIKM08: Conference on Information and Knowledge Management, Napa Valley, CA, USA, 26–30 October 2008; pp. 931–940.
24. Zhu, L.; Xu, C.; Guan, J.; Zhang, H. SEM-PPA: A semantical pattern and preference-aware service mining method for personalized point of interest recommendation. *J. Netw. Comput. Appl.* **2017**, *82*, 35–46. [CrossRef]
25. Bin, S.; Sun, G. Collaborative Filtering Recommendation Algorithm Based on Multi-relationship Social Network. *Comput. Sci.* **2019**, *46*, 56–62. [CrossRef]
26. Jia, J.; Liu, P.; Chen, W. Improved Matrix Factorization Algorithm Using Social Information for Recommendation. *Comput. Eng.* **2021**, *47*, 97–105. [CrossRef]
27. 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 Proceedings of the AAAI'15: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 123–129.
28. Yang, B.; Lei, Y.; Liu, J.; Li, W. Social Collaborative Filtering by Trust. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1633–1647. [CrossRef]
29. Chen, Z.; Li, H.; Du, J. Research on Recommendation Algorithm Based on Heterogeneous Graph neural Network. *J. Hunan Univ. Nat. Sci.* **2021**, *48*, 137–144. [CrossRef]
30. Bi, Z.; Jing, L.; Shan, M.; Dou, S.; Wang, S.; Yang, X. Hierarchical Social Recommendation Model Based on a Graph Neural Network. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 9107718. [CrossRef]
31. Zhou, H.; Liu, J.; Wang, H. A Social Movie Recommendation Model Based on Graph Neural Network and Tag Overlapping Community. *Inf. Stud. Theory Appl.* **2021**, *44*, 164–170. [CrossRef]

32. Chang, J.; Gao, C.; Zheng, Y.; Hui, Y.; Niu, Y.; Song, Y.; Jin, D.; Li, Y. Sequential Recommendation with Graph Neural Networks. In Proceedings of the SIGIR'21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Online. 11–15 July 2021; pp. 378–387.
33. Yu, Z.; Lian, J.; Mahmoody, A.; Liu, G.; Xie, X. Adaptive User Modeling with Long and Short-Term Preferences for Personalized Recommendation. In Proceedings of the IJCAI'19: The 28th International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 4213–4219.
34. Peng, D.; Yuan, W.; Liu, C. HARSAM: A Hybrid Model for Recommendation Supported by Self-Attention Mechanism. *IEEE Access* **2019**, *7*, 12620–12629. [CrossRef]
35. Pang, G.; Wang, X.; Hao, F.; Wang, L.; Wang, X. Efficient point-of-interest recommendation with hierarchical attention mechanism. *Appl. Soft Comput.* **2020**, *96*, 106536. [CrossRef]
36. Zhuo, J.; Lei, J.; Zhou, X. AGRU-GNN Graph Network for Social Recommendation. *Comput. Syst. Appl.* **2021**, *30*, 219–227. [CrossRef]
37. Tao, Z.; Wei, Y.; Wang, X.; He, X.; Huang, X.; Chua, T.-S. MGAT: Multimodal Graph Attention Network for Recommendation. *Inf. Process. Manag.* **2020**, *57*, 102277. [CrossRef]
38. Huang, W.; Wei, W.; Zhang, J.; Deng, Z. Recommendation Method Based on Attention Mechanism and Review Text Deep Model. *Comput. Eng.* **2019**, *45*, 176–182. [CrossRef]
39. Vijaikumar, M.; Shevade, S.; Murty, M.N. GRAM-SMOT: Top-N Personalized Bundle Recommendation via Graph Attention Mechanism and Submodular Optimization. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Ghent, Belgium, 14–18 September 2020; pp. 297–313.
40. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
41. Wang, C.-S.; Chen, B.-S.; Chiang, J.-H. TDD-BPR: The topic diversity discovering on Bayesian personalized ranking for personalized recommender system. *Neurocomputing* **2021**, *441*, 202–213. [CrossRef]
42. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
43. Massa, P.; Avesani, P. Trust-aware recommender systems. In Proceedings of the 2007 ACM conference on Recommender systems, Minneapolis, MN, USA, 19–20 October 2007; pp. 17–24.
44. Wang, X.; He, X.; Wang, M.; Feng, F.; Chua, T.-S. Neural Graph Collaborative Filtering. In Proceedings of the SIGIR'19: The 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 165–174.
45. Chen, C.; Zhang, M.; Liu, Y.; Ma, S. Social Attentional Memory Network: Modeling Aspect- and Friend-Level Differences in Recommendation. In Proceedings of the WSDM'19: The Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, VIC, Australia, 11–15 February 2019; pp. 177–185.