Research Article

Hierarchical Social Recommendation Model Based on a Graph Neural Network

Zhongqin Bi¹, Lina Jing¹, Meijing Shan², Shuming Dou^{1,3} and Shiyang Wang¹

¹College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai 200090, China

²Institute of Information Science and Technology, East China University of Political Science and Law, Shanghai 201620, China

³China Electronic Systems Engineering Corp, Beijing 100141, China

Correspondence should be addressed to Shuming Dou; d2313776747@163.com

Received 9 July 2021; Revised 30 July 2021; Accepted 17 August 2021; Published 31 August 2021

Academic Editor: Xiaoxian Yang

Copyright © 2021 Zhongqin Bi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the continuous accumulation of social network data, social recommendation has become a widely used recommendation method. Based on the theory of social relationship propagation, mining user relationships in social networks can alleviate the problems of data sparsity and the cold start of recommendation systems. Therefore, integrating social information into recommendation systems is of profound importance. We present an efficient network model for social recommendation. The model is based on the graph neural network. It unifies the attention mechanism and bidirectional LSTM into the same framework and uses a multilayer perceptron. In addition, an embedded propagation method is added to learn the neighbor influences of different depths and extract useful neighbor information for social relationship modeling. We use this method to solve the problem that the current research methods of social recommendation only extract the superficial level of social networks but ignore the importance of the relationship strength of the users at different levels in the recommendation. This model integrates social relationships into user and project interactions, not only capturing the weight of the relationship between different users but also considering the influence of neighbors at different levels on user preferences. Experiments on two public datasets demonstrate that the proposed model is superior to other benchmark methods with respect to mean absolute error and root mean square error and can effectively improve the quality of recommendations.

1. Introduction

With the rapid development of the Internet and the explosive growth of information, recommendation systems help information consumers find the content that they are interested in from a large amount of information, effectively relieving the pressure of information overload. Collaborative filtering [1] is one of the most commonly used recommendation methods. It uses historical data of user-item interactions to learn users' preferences for projects without explicit user and project information [2]. In addition to user-project interactions, social relationships between users also provide potential information for modeling user preferences. According to the social theory, people in a social network are affected by their social relationships, resulting in the homogeneity of preferences of social neighbors. Therefore, the influence of social relations on recommendation systems has attracted increasing

attention [3]. The general approach of recommendation algorithms based on social information is to integrate social relations based on the rating matrix information. By contrast, social relations are the mapping of natural social relations and reflect the users' natural social relations [4–6].

In recent years, researchers have used the methods of convolutional networks, recurrent networks, and deep auto-encoders to define and design neural network structures for processing graph data. Such deep neural networks are called graph neural networks (GNNs) [10–14]. Some recommendation systems use GNNs as feature learning tools to extract useful features from social information [7–9]. The main idea is to use neural networks to iteratively aggregate feature information from the local graph neighborhood. At the same time, node information can be transmitted and aggregated through graphs after transformation. In addition, the data in the social recommendation system can be expressed as

user-user social graphs and user-item graphs that provide potential advantages for the learning of GNNs. Some studies use GNNs to incorporate social network information into users and item potential factors to learn. For example, GraphRec [6] proposed a social recommendation framework of GNN that aggregates user-item interaction information and social relationship information simultaneously when performing predictions. DiffNet [44] is a hierarchical influence propagation structure for the simulation of the recursive dynamic diffusion process in social recommendations. Through this structure, both users and projects can be expressed as embedded information containing collaboration and functional content.

Although these recommendations based on GNNs have been very successful, they do not take full advantage of social network information. First, most of the methods simply link the user representation extracted from the social relationship to the user or item information and lack in-depth learning of the social relationship. Second, most of them only consider the role of direct (local) neighbors and ignore the influence of indirect neighbors that come from a few hops away. When direct neighbors are sparse, indirect neighbors are also helpful for user representation, and the recommendation system needs to take this indirect social relationship into account. Finally, not all neighbors are useful when recommending users. For example, a user who likes Macs may not be indicative that all of his neighbors like Macs. If the influence of these neighbors is treated equally, then the recommendation effect will be reduced. Therefore, it is necessary for the recommendation algorithm to distinguish strong and weak social relations and reasonably allocate the influence weight of neighbors on users.

In this paper, we use a GNN to design a hierarchical social collaborative filtering framework called GHSCF. The model includes three modules: the embedded propagation layer, the embedded layer, and the sequential learning layer. The framework can make full use of social network information for recommendations. Compared with other social recommendation methods, this model has the following advantages. First, an embedded propagation method is designed to learn the influence of neighbors at different levels. When direct neighbors have no valuable information to share, users may use indirect neighbor information. In addition, the attention mechanism is introduced to distinguish strong and weak social connections, screen helpful neighbors, and effectively capture the influence weights of different neighbors. Second, the user preference sequence generated by different levels of communication learning represents user modeling of different levels of neighbors. The long-term and short-term memory (LSTM) network is used for serial modeling and makes final recommendations based on the influence of neighbors at different levels.

In summary, the contributions of this article are as follows:

- (1) We designed a hierarchical social recommendation model based on the graph neural network to capture user relationships along social networks and simulate changes in influence between users at different levels to improve the accuracy of user representation

- (2) We propose an embedded propagation based on GNN that introduces a user relationship metric for the most relevant items and alleviates the problem of inconsistent user preferences for target items
- (3) We combine the attention mechanism with the bidirectional LSTM network and propose a bidirectional LSTM with an attention mechanism that can help sequence modeling
- (4) We conduct comprehensive experiments on two standard datasets and compare existing social recommendation frameworks in order to evaluate and demonstrate the effectiveness of the proposed approach, proving the superior performance of our method. Compared with the existing methods, our method not only considers the information from the neighbors but also considers the information of the original neighbors. Our method reasonably distinguishes strong and weak social relations

The rest of this article is organized as follows: in Section 2, we briefly introduce the related work. In Section 3, we provide some preliminary information. In Section 4, we introduce our GHSCF architecture model in detail. Section 5 describes the experimental results. Section 6 concludes the paper.

2. Related Work

The collaborative filtering recommendation algorithm [1] is the most widely used method in recommendation systems to predict the interaction between users and products. Using the historical interaction information between users and products, the users and products of interest can be modeled. In addition to user-item interaction, the use of social relationships to make suggestions has attracted great attention [20, 21]. Many social recommendation methods [22–27] have demonstrated the effectiveness of incorporating social relationships into recommendations. SoDimRec [26] uses the heterogeneity of social relations and the weak dependence on social networks to make recommendations. SOReg [17] proposes two kinds of social recommendation algorithms and uses a social regularization conditional constraint matrix to decompose the objective function. TRUSTMF [18] simulates the interaction between users and maps users to two low-dimensional spaces by decomposing the social trust network into the trust space and trusted space. Although the social recommendation method based on matrix decomposition has achieved a good recommendation effect, its linearity is insufficient to reveal the complex nonlinear relationship between users and products.

Recently, deep neural networks have been used to enhance recommendation systems [28, 29, 43]. Most of these systems use deep neural networks as feature learning tools to extract features from auxiliary information, such as the text description of items [30, 32, 33, 46]. Neural matrix factorization (NeuMF) [19] proposes a neural cooperative filtering (NCF) framework to learn the nonlinear interactions between users and projects. Later, researchers

attempted to incorporate social relationship information into the NCF model to improve recommendation performance. Neural social collaborative ranking (NSCR) [31] uses social network information as graph regularization to extend the NeuMF model so that nearby neighbors have similar potential vectors. The deep neural network model on social relations (DeepSoR) [12] learns the nonlinear characteristics of each user from social relations and integrates them into probability matrix factorization for rating prediction. The deep social collaborative filtering (DSCF) [45] framework uses a random walk layer to generate the item perceived social sequence and extract relevant information from distant neighbors for user-item nonlinear fusion. All of these works address the cross-domain task of social recommendation and thus differ from traditional social recommendation systems.

GNNs have demonstrated the ability to learn graph structure data [11, 34, 35, 42, 45, 48]. In the social recommendation task, the relationship between users employs typical graph data. Thus, GNNs have a natural advantage when used for this task. In [6, 36, 37, 47], SocialGCN [36] was proposed to capture user preferences by utilizing the advantages of GCNs and modeling the diffusion process of preference information in the figure. However, SocialGCN does not consider the relative importance of different neighbors and the degree of trust of the users for different neighbors when buying different items. GraphRec [6] implements a new graphical neural network framework for social recommendation. In particular, a principled approach is designed to capture interaction and rating information jointly and to simulate both types of graphs and heterogeneous intensities coherently. However, GraphRec does not take into account the influence of remote neighbors. This paper proposes a hierarchical social recommendation model that can make full use of social network information to make recommendations. This model can simulate the influence process of user preference information spread in social networks and model user preference considering the importance of neighbors at different levels.

3. Preliminaries

3.1. Attentional Mechanism. The attention mechanism [39] refers to the human visual attention and thinking mode to help the model quickly screen valuable information from a large amount of content. It uses the scoring method to extract the weight of key information and is widely used in various deep learning tasks, such as natural language processing, image classification, and speech recognition [40, 41].

Each layer of the model uses an input information vector $X = [x_1, x_2, \dots, x_n]$ to search and select important information in X in a given query vector q . A “soft” selection mechanism is usually adopted to capture the weight distribution of each piece of information in the input information. The weight of x_i is calculated as follows:

$$e_i = \mathbf{v}^T \tanh(W\mathbf{x}_i + b_i),$$

$$\alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^N \exp(e_j)}. \quad (1)$$

The probability vector constituted by α_i is called the attention distribution. W , U , and V are learnable network parameters. The input information is summarized by the weighted average method and is calculated as follows:

$$s = \sum_{i=1}^N \alpha_i x_i. \quad (2)$$

3.2. LSTM. An LSTM network [38] is a sequential convolutional network derived from a cyclic neural network that alleviates the vanishing and exploding gradient problems of cyclic neural networks by using cell states and gate mechanisms. LSTM can learn long-term dependence, making it suitable for solving many long sequence learning problems. The excellent performance of LSTM arises from its unique internal structure. Specifically, LSTM uses three gate structures to control the flow of the information state. In each time step t , the current cell state c_{t-1} and the hidden layer state h_{t-1} are updated through an input sequence element value x_t , the cell state c_t , and the hidden layer state h_t of the time step $t - 1$. The formula is as follows:

$$\begin{bmatrix} i_t \\ f_t \\ o_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \end{bmatrix} (W[h_{t-1}; x_t] + b), \quad (3)$$

$$\hat{c}_t = \tanh(W_c[h_{t-1}; x_t] + b_c),$$

$$c_t = f_t \odot \hat{c}_{t-1} + i_t \odot \hat{c}_t,$$

$$h_t = o_t \odot \tanh(c_t),$$

where i_t , f_t , and o_t represent gate activation, σ is the sigmoid activation function, $\tanh$ is the hyperbolic tangent activation function, and $\odot$ represents vector element multiplication. Intuitively, ForgetGate f_t controls the extent to which the previous memory unit is forgotten, and input gate i_t controls how many input gates are updated by each unit. The output gate o_t controls the output of the internal memory state. The hidden layer state h_t represents the output information of the internal memory unit of the LSTM unit.

The prediction may need to be determined based on both previous and subsequent inputs, improving the performance of the network. A more common approach is to use bidirectional LSTM to simulate textual semantics from the forward and backward states of the hidden layer. For a sequence vector $[x_1, x_2, \dots, x_T]$, the forward LSTM reads the sequence x_1 to x_T , and the backward LSTM reads the sequence x_T to x_1 and then connects the forward and backward hidden layer states $\vec{h}_t$ and $\overleftarrow{h}_t$, h_t integrating all of the information of the sequence around x_t .

4. Model

In this section, we describe in detail the proposed model, namely, GHSCF. The framework of the model is shown in Figure 1. The model includes three modules: the embedded propagation layer captures the intensity of different levels

FIGURE 1: Hierarchical recommendation model based on a graph neural network.

of neighbor social relations and carries out user preference modeling, the embedding layer models the user-item interaction, and the sequence learning layer uses LSTM to fuse user-item interaction sequences composed of different levels. The remainder of this section describes each module in detail.

4.1. Embedded Propagation Layer. In this layer, a GNN-based messaging structure is constructed so that the model can capture user relationships along with the social network and optimize user preferences with different levels of neighbors, as shown in Figure 2. To better illustrate the internal structure of the embedded propagation layer, we first introduce the propagation structure of one layer and then extend it to multiple layers.

In social recommendation, when trying to recommend an item for a given user u , the user's preferences are influenced by direct neighbors, while indirect neighbors can also provide valuable help. At the same time, the connection between users can be transmitted through social networks to influence the behavior of other users further. Therefore, the recommendation algorithm should include all of these neighbors in the potential factors of user preferences and introduce the attention mechanism to select representative social neighbors to describe the social information of users. Therefore, this paper proposes a user potential representation method based on social space that aggregates the potential factors of neighboring users to learn the preferences of target users. Specifically, it iteratively learns the potential representation of user u according to its neighbor set N_u , as follows:

$$e_u^{(1)} = \text{LeakyReLU} \left(W_1 e_u + W_2 \sum_{k \in N_u} a_{uk} e_{u_k} \right). \quad (4)$$

FIGURE 2: Embedded propagation model.

In the formula, $e_u^{(1)}$ represents the representation of user u obtained after the first embedded propagation layer, a_{uk} represents the influence weight of neighbors, that is, $1/|N_u|$, e_{u_k} represents the potential representation of neighbor k , and LeakyReLU [15] is the activation function that allows the use of positive signals and small negative signals to encode messages. Notably, in addition to integrating the neighborhood information, the model also preserves the user's original characteristics.

Although users can be modeled based on neighborhood information, no explicit recommendations are specified; that is, users share social relationships with their neighbors' individual interacting items. However, the preferences of target users and their neighbors may not be the same. Sometimes, the neighbors' information does not help recommend target items, and only the information related to the item contributes. Therefore, the model needs to select an item related to the target item for each user to realize similarity modeling between the user and a neighbor. It is important to note that only the most relevant items contribute to the recommendation of a particular item. This is because the most relevant items are the most important influencing factors for the target item, while other items may not be helpful or contribute

to noise. Specifically, for the target item p , we select the item most related to p from the item interaction set of user u :

$$p_u = \arg \max_{p_i \in V_u} \cos(p_i, p). \quad (5)$$

In the formula, v_u is the user's item interaction set, and $\cos(p_i, p)$ is a function to measure the similarity between item p and item p_i . Cosine similarity is used according to experience:

$$\cos(p_i, p) = \frac{f(p_i)^T f(p)}{|f(p_i)| |f(p)|}. \quad (6)$$

In the formula, the f function generates the appropriate feature representation for the item. In this paper, the NeuMF model is used to extract the representation of objects and to calculate the similarity.

As mentioned earlier, there are strong ties and weak ties in a social network, and users may share more similar items with strong ties. Therefore, through the most similar items between users, we can learn the relationship strength.

$$\begin{aligned} \beta_{uk} &= \text{LeakyReLU} \left(v^T \cdot [W e_{p_k} \oplus W e_{p_u}] \right), \\ a_{uk} &= \frac{\exp(\beta_{uk})}{\sum_{i \in N_u} \exp(\beta_{ui})}, \end{aligned} \quad (7)$$

where e_{p_k} represents the embedding representation of the most relevant item of user k and e_{p_u} represents the embedding representation of the most relevant item p of user u . a_{uk} is the degree of correlation between users. With the first layer of direct user modeling, the simulation network propagation mode overlays more indirect neighbors, leading to deeper user preference modeling, which is crucial for the subsequent serialization modeling.

Through n embedded propagation layers, user u can receive information transmitted by neighbors on n layers. In the NTH layer, user u is recursively expressed as

$$e_u^{(n)} = \text{LeakyReLU} \left(W_1^n e_u^{(n-1)} + W_2^n \sum_{k \in N_u} a_{uk}^n e_k^{(n)} \right), \quad (8)$$

where $W_1^n W_2^n$ is a trainable transformation matrix, e_u^{n-1} is the user preference of the upper layer generated by the transmission signal of the previous layer and aggregates the neighbor information of the upper layer, e_k^n is the neighbor K embedding vector of the N^{th} layer, and a_{uk}^n is the influence factor. Through the propagation of social networks, on the basis of direct neighbor modeling, the modeling information of indirect neighbors' user preferences is accumulated layer by layer, which is represented as a user preference sequence $[e_u^{(1)}, e_u^{(2)}, \dots, e_u^{(n)}]$ that reflects that users at different friend levels will be affected by different neighbors.

4.2. Embedded Layer. The user preference sequence is composed of user modeling by neighbors and users in a social

network under the influence of the strength of item interaction. Therefore, it is necessary to integrate the user preference sequence into the user-item interaction first, so that the user's various preference information is fully considered. Clearly, the bad and good comments from the user's social neighbors reflect the user's choice of items in a sequential manner. The interaction between the user and the item is highly nonlinear, and a multilayer perceptron (MLP) can fuse the interaction information with the user preference information. The input of the MLP includes user embedding e_u , single-layer user preference embedding $e_u^{(i)}$, $i \in [1, n]$, item embedding as input e_p , and output user-item interactive embedding as m_i , $i \in [1, n]$, that is given by

$$m_i = f_{u,p} \left(e_u \oplus e_p \oplus e_u^{(i)} \right), \quad i \in [1, n]. \quad (9)$$

Using an MLP, a sequence of all user-item interactions can be obtained from the neighbors $[m_u^1, m_u^2, \dots, m_u^n]$.

4.3. Sequential Learning Layer. The purpose of the sequence learning layer is to integrate the sequence of rating vectors further or to integrate the information of rating vectors under different levels to obtain a unified representation. Since all of the neighbors in the sequence will affect the prediction, for the distant neighbors, the model needs to capture the remote social information between these neighbors and the user u . In addition, the influence between users in a social networks is bidirectional, and bidirectional LSTM has inherent advantages in sequence modeling:

$$\begin{aligned} \vec{h}_i &= \overrightarrow{\text{LSTM}}(m_i), \quad i \in [1, n], \\ \overleftarrow{h}_i &= \overleftarrow{\text{LSTM}}(m_i), \quad i \in [1, n]. \end{aligned} \quad (10)$$

The output feature $h_i = [\vec{h}_i; \overleftarrow{h}_i]$ is obtained by connecting the forward hidden layer state $\vec{h}$ and the backward hidden layer state $\overleftarrow{h}$ of the sequence. Then, the attention mechanism is used to learn the important features in the sequence:

$$s = \sum_{i=1}^n \alpha_i h_i. \quad (11)$$

In the formula, α_i is the weight of attention. α_i is used to assign the importance of information at different levels of the sequence:

$$\begin{aligned} a_i &= v_a^T \tanh(W_a h_i + b_a), \\ \alpha_i &= \frac{\exp(a_i)}{\sum_{i \in N_u} \exp(a_i)}. \end{aligned} \quad (12)$$

4.4. Model Learning. In this work, we apply the proposed model to the recommendation task of rating prediction

TABLE 1: Statistics for the two datasets.

Dataset	Users	Items	Ratings	Links	Rating density	Social connection density
Ciao	7,317	10,4975	283,319	111,781	0.0368%	0.2087%
Epinions	18,088	261,649	764,352	355,813	0.0161%	0.1087%

and finally use the MLP to predict user u 's rating score for item p .

$$r'_{up} = f(s). \quad (13)$$

Since the task focused on in this work is rating prediction, the model selects a commonly used objective function:

$$\text{Loss} = \frac{1}{2|O|} \sum_{u,p \in O} (r'_{up} - r_{up})^2, \quad (14)$$

where $|O|$ is the number of ratings observed and r_{up} is the true rating of user u on item p .

5. Experiment

5.1. Experimental Data. In this experiment, we selected two commonly used public datasets to evaluate the effectiveness of this method, namely, Ciao and Epinions. They are item rating datasets taken from two different social platforms that contain user rating records and social relationship information. The user rating in the dataset is recorded in intervals of 1 for each user to view the items, with [1, 5] as the rating interval. The social relationship information gives the set of all social neighbors for each user. The statistics for these two datasets are shown in Table 1.

For the convenience of subsequent use, 80% of the data in each dataset were randomly selected for training, 10% were selected for validation, and 10% were selected for testing. All hyperparameters were selected based on the validation set. In this model, Minh and Salakhutdinov's method [16] was selected as the optimization algorithm, and the initial learning rate was set to 0.005. At the same time, to alleviate the overfitting problem, dropout technology was used in the training process, and the ratio was set to 0.6. Unless otherwise mentioned, three hidden layers were used for all neural networks, and the number of neighboring layers was set to 4. For the embedded size, the model tests the values of {8,16,32,64,128,256} and searches for the batch size and learning rate in {16,32,64,128,512} and {0.0005, 0.001, 0.005, 0.01, 0.05, 0.1}. After repeated tests, the embedded size of the model was set to 64, the batch size to 128, and the learning rate to 0.005. In addition, following a rule of thumb, the size of the hidden layer was set to the embedded size (the dimension of the underlying factor), and the activation function is set to ReLU. Finally, the parameters of the baseline algorithm were initialized and repeatedly adjusted through testing to achieve the best performance.

To evaluate the quality of the recommendation algorithm, two commonly used prediction accuracy indexes,

namely, the mean absolute error (MAE) and root mean square error (RMSE), were used in this paper. Smaller MAE and RMSE values indicate better prediction accuracy.

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{1}{N} \sum_{i=1}^N (y^{\wedge(i)} - y^{(i)})^2}, \\ \text{MAE} &= \frac{1}{N} \sum_{i=1}^N |y^{\wedge(i)} - y^{(i)}|, \end{aligned} \quad (15)$$

where N represents the number of test examples.

5.2. Contrast Experiment

- (i) PMF [16]: probabilistic matrix factorization is a standard rating prediction model that uses only ratings for collaborative filtering
- (ii) SoRec [17]: social regularization models social network information as regularization terms and constrains the matrix factorization framework
- (iii) SocialMF [18]: this model effectively integrates social information into the conventional matrix decomposition framework. However, it assumes that the trust relationship between all users is the same and simulates social influence based on that assumption
- (iv) NeuMF [19]: this method is a state-of-the-art neural CF model that uses multiple hidden layers above the elements and connects embedded users and objects to capture their nonlinear feature interactions
- (v) DeepSoR [12]: this model uses a deep neural network to learn the representation of each user from social relations and integrates it into probability matrix factorization for rating prediction
- (vi) DSCF [8]: a deep social collaborative filtering framework that extracts valuable information from social relationships for recommendation and generates an item-aware social sequence for integration into user-item interaction
- (vii) DiffNet [44]: a social recommendation framework based on a GNN. It proposes a hierarchical diffusion method to simulate the transmission process of the users' potential preferences in social trust networks

TABLE 2: RMSE and MAE experimental results of the models.

Models	Ciao		Epinions	
	MAE	RMSE	MAE	RMSE
PMF	0.9021	1.1238	0.9952	1.2128
SoRec	0.8611	1.0848	0.9119	1.1703
SocialMF	0.8270	1.0501	0.8837	1.1328
NeuMF	0.8062	1.0617	0.9072	1.1476
DeepSoR	0.7739	1.0316	0.8383	1.0972
DSCF	0.7270	0.9867	0.8275	1.0667
DiffNet	0.7194	0.9782	0.8158	1.0741
GHSCF	0.7206	0.9194	0.7968	0.9731

5.3. Results and Analysis. As shown in Table 2, probabilistic matrix factorization (PMF) performed poorly on both datasets. This indicates that the internal product is insufficient for capturing the complex relationship between the user and item, further limiting performance. NeuMF consistently outperformed PMF in all cases, demonstrating the importance of nonlinear characteristic interactions between the user and item embedding. Both SoRec and SocialMF utilize ratings and social network information, while PMF only uses rating information. These results indicate that social network information is complementary to rating information. DeepSoR, DSCF, and DiffNet outperform SoRec and SocialMF, and all of these methods use ratings and social network information. However, DeepSoR, DiffNet, and DSCF are based on neural network structures, further demonstrating the superiority of neural networks in making recommendations. In contrast to the other methods, DiffNet shows quite a strong performance, implying that GNNs can deeply mine the underlying information of graphic data. Notably, GHSCF consistently outperformed all baseline approaches. Compared to DeepSoR and DSCF, GHSCF provides advanced modeling components to combine different levels of neighbors in social network information. Further analysis was performed to better understand the contributions of the model components.

5.4. Model Analysis

5.4.1. Effectiveness of the Number of Embedded Propagation Layers. To investigate whether the model can benefit from multiple embedded propagation layers, the model performance is tested by changing the number of layers of the model. We explore model performance at levels 1 through 4 according to the experimental results shown in Table 3.

With increasing model layers, the RMSE and MAE decrease gradually and tend to become stable, indicating that the model can not only effectively utilize the direct neighbor information but also capture the influence of indirect neighbors on the modeling of user preferences. It is important to note that in dataset Ciao, overfitting occurs, possibly due to the noise that an architecture that applies too deeply can generate for the learning representation. When the number of propagation layers is changed, the model is always superior to other methods on the two datasets, indicating that

TABLE 3: RMSE and MAE of different communication levels.

Layers	Ciao		Epinions	
	MAE	RMSE	MAE	RMSE
GHSCF-1	0.7271	0.9605	0.8081	1.0558
GHSCF-2	0.7238	0.9598	0.8075	1.0247
GHSCF-3	0.7206	0.9194	0.8071	0.9731
GHSCF-4	0.7229	0.9601	0.8068	1.0331

the propagation mechanism in this model can effectively capture the neighbor information at different levels, which is helpful for improving user preference modeling.

5.4.2. Neighbor Relationship Modeling and LSTM Validity Analysis. To better understand the proposed model, this experiment compared three variants of the GHSCF model: GHSCF- β , GHSCF- α , and GHSCF-Item. These three variants are defined as follows:

- (i) GHSCF- β : the purpose of this variant is to study the effect of the attention factor β on the strength of the learning user's relationship, assuming that each neighbor has the same influence on the target user
- (ii) GHSCF- α : in this variant, the bidirectional LSTM structure is replaced by the serial average; that is, the user preferences of different levels are integrated by averaging, and the differences between different levels are ignored
- (iii) GHSCF-item: this variant only uses the user vector to evaluate the strong or weak relationship between the user and his neighbors; that is, the influence of the most relevant items is no longer considered

The results on the Ciao dataset are shown in Figure 3. The results using the Epinions dataset are not shown here because similar observations can be made. First, the performance of the GHSCF and GHSCF items is better than that of the GHSCF β . This indicates that not all neighbors have the same influence on target users, and correctly distinguishing the strong and weak relations of neighbors is beneficial for user preference modeling. In addition, the performance of the GHSCF items is worse than that of the GHSCFs. This is because adjacent users may not have an item intersection, and it is not sufficient to capture the strength of user relationships only by considering the user's own social network information. Therefore, not all information from neighbors is useful when recommending a particular project, and it is more valuable to select neighbor-related projects to measure user relationships. Second, the performance of GHSCF- α is significantly lower than that of GHSCF. This indicates that the bidirectional LSTM component better learns the representation of the sequence of user-item interactions at different levels. Therefore, it is further proven that target users in social networks will have different degrees of preferences for target products due to the influence of neighbors separated by different distances.

FIGURE 3: Results of MAE and RMSE experiments for different variants.

FIGURE 4: Effect of different embedded sizes.

5.4.3. Effect of the Embedded Size. Figure 4 shows the impact of the embedded sizes of users and items on model performance. As the embedded size increases, the recommended performance first increases and then decreases. When the embedded size is increased from 8 to 64, the performance is significantly improved. However, when the embedding size is 256, the performance is degraded. This shows that using a large embedded size has a powerful representation. If the length of embedding is too large, then the complexity of the model will increase significantly, and overfitting may occur. Therefore, the correct embedding length must be found to balance performance and complexity.

6. Conclusion and Future Work

We propose a hierarchical social recommendation model that designs an embedded propagation layer. This model takes into account information not only from immediate neighbors but also from distant neighbors, helping to model user preferences at different neighborhood levels. In addition, this paper introduces an object perception method to capture the strength of neighborhood relationships, taking into account the importance of close objects between users. Finally, we propose a bidirectional LSTM with an attention mechanism to extract user-item interaction sequences with

different preferences. In addition, to solve the overfitting problem, this paper uses the dropout strategy. Experiments show that neighbors at different levels play an important role in social information modeling. Finally, the feasibility of these changes is tested by a comparative experiment that shows that the GNN can learn more accurate social influence and further improve the recommendation performance.

In this work, we only use user-item interaction information to measure the similarity between projects. By contrast, tangential information associated with the project, such as text description and visual content of images, may be more valuable. In addition, we now consider ratings with social information to be static. However, ratings and social information are dynamic. Therefore, in future work, we will consider the establishment of a dynamic GNN for social recommendation.

Data Availability

The experimental data, named Ciao and Epinions, are from previously reported studies and datasets, which have been cited. Ciao is available at <https://www.librec.net/datasets.html>, and Epinions is available at <http://www.trustlet.org/epinions.html>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Nature Science Foundation of China (No. 61972357).

References

- [1] B. Sarwar, G. Karypis, and J. Konstan, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, 2001.
- [2] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence. Advances in artificial intelligence*, vol. 2009, pp. 1–19, 2009.
- [3] J. Tang, X. Hu, and H. Liu, "Social recommendation: a review," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1113–1133, 2013.
- [4] E. Bakshy, I. Rosenn, and C. Marlow, "The role of social networks in information diffusion," in *Proceedings of the 21st international conference on World Wide Web*, pp. 519–528, 2012.
- [5] W. Fan, T. Derr, and Y. Ma, "Deep adversarial social recommendation," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019.
- [6] W. Fan, Y. Ma, and Q. Li, "Graph neural networks for social recommendation," in *The World Wide Web Conference*, pp. 417–426, 2019.
- [7] R. Berg, T. N. Kipf, and M. Welling, *Graph Convolutional Matrix Completion*, 2017, arXiv preprint arXiv:1706.02263.
- [8] W. Fan, Y. Ma, and D. Yin, "Deep social collaborative filtering," in *Proceedings of the 13th ACM Conference on Recommender Systems*, pp. 305–313, 2019.
- [9] T. N. Kipf and M. Welling, *Semi-supervised Classification with Graph convolutional networks*, 2016, arXiv preprint arXiv:1609.02907.
- [10] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1025–1035, 2017.
- [11] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in Neural Information Processing Systems*, vol. 29, pp. 3844–3852, 2016.
- [12] W. Fan, Q. Li, and M. Cheng, "Deep modeling of social relations for recommendation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [13] X. Wang, X. He, and M. Wang, "Neural graph collaborative filtering," in *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 165–174, 2019.
- [14] H. Gao, L. Kuang, Y. Yin, B. Guo, and K. Dou, "Mining consuming behaviors with temporal evolution for personalized recommendation in mobile marketing apps," *Mobile Networks and Applications*, vol. 25, no. 4, pp. 1233–1248, 2020.
- [15] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," *Proc. ICML*, vol. 30, no. 1, 2013.
- [16] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," *Advances in Neural Information Processing Systems*, pp. 1257–1264, 2008.
- [17] H. Ma, D. Zhou, and C. Liu, "Recommender systems with social regularization," in *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 287–296, 2011.
- [18] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the fourth ACM conference on Recommender systems*, pp. 135–142, 2010.
- [19] X. He, L. Liao, and H. Zhang, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, pp. 173–182, 2017.
- [20] J. Tang, C. Aggarwal, and H. Liu, "Recommendations in signed social networks," in *Proceedings of the 25th International Conference on World Wide Web*, pp. 31–40, 2016.
- [21] B. Yang, Y. Lei, J. Liu, and W. Li, "Social collaborative filtering by trust," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1633–1647, 2017.
- [22] G. Guo, J. Zhang, and N. Yorke-Smith, "Trustsvd: 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*, vol. 29, 2015no. 1.
- [23] S. Purushotham, Y. Liu, and C. C. J. Kuo, "Collaborative topic regression with social matrix factorization for recommendation systems," in *Proceedings of the 24th International Conference on Machine Learning*, 2012.
- [24] X. Wang, W. Lu, and M. Ester, "Social recommendation with strong and weak ties," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 5–14, 2016.

- [25] T. Zhao, J. McAuley, and I. King, "Leveraging social connections to improve personalized ranking for collaborative filtering," in *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pp. 261–270, 2014.
- [26] J. Tang, S. Wang, and X. Hu, "Recommendation with social dimensions," in *Thirtieth AAAI Conference on Artificial Intelligence*, pp. 251–257, 2016.
- [27] H. Gao, Y. Xu, Y. Yin, W. Zhang, R. Li, and X. Wang, "Context-aware QoS prediction with neural collaborative filtering for Internet-of-Things services," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4532–4542, 2019.
- [28] M. Jiang, P. Cui, F. Wang, W. Zhu, and S. Yang, "Scalable recommendation with social contextual information," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 11, pp. 2789–2802, 2014.
- [29] Y. Yin, Q. Huang, H. Gao, and Y. Xu, "Personalized APIs recommendation with cognitive knowledge mining for industrial systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, 2021.
- [30] H. Wang, N. Wang, and D. Y. Yeung, "Collaborative deep learning for recommender systems," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1235–1244, 2015.
- [31] X. Wang, X. He, and L. Nie, "Item silk road: recommending items from information domains to social users," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 185–194, 2017.
- [32] S. Deng, L. Huang, G. Xu, X. Wu, and Z. Wu, "On deep learning for trust-aware recommendations in social networks," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 5, pp. 1164–1177, 2017.
- [33] H. Gao, W. Huang, and Y. Duan, "The cloud-edge-based dynamic reconfiguration to service workflow for mobile e-commerce Environments," *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 1, pp. 1–23, 2021.
- [34] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.
- [35] R. Ying, R. He, and K. Chen, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 974–983, 2018.
- [36] L. Wu, P. Sun, and R. Hong, *SocialGCN: an efficient graph convolutional network based model for social recommendation*, 2018, arXiv preprint arXiv:1811.02815.
- [37] W. Song, Z. Xiao, and Y. Wang, "Session-based social recommendation via dynamic graph attention networks," in *Proceedings of the Twelfth ACM international conference on web search and data mining*, pp. 555–563, 2019.
- [38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [39] D. Bahdanau, K. Cho, and Y. Bengio, *Neural machine translation by jointly learning to align and translate*, 2014, arXiv preprint arXiv:1409.0473.
- [40] K. Ahmed, N. S. Keskar, and R. Socher, *Weighted transformer network for machine translation*, 2017, arXiv preprint arXiv:1711.02132.
- [41] K. Xu, J. Ba, R. Kiros et al., *Show, attend and tell: neural image caption generation with visual attention*, 2015, <https://arxiv.org/abs/1502.03044v3>.
- [42] H. Gao, X. Qin, and R. J. D. Barroso, "Collaborative learning-based industrial IoT API recommendation for software-defined devices: The implicit knowledge discovery perspective," in *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2020.
- [43] T. Bai, J. R. Wen, and J. Zhang, "A neural collaborative filtering model with interaction-based neighborhood," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1979–1982, 2017.
- [44] L. Wu, P. Sun, and Y. Fu, "Neural influence diffusion model for social recommendation," in *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pp. 235–244, 2019.
- [45] T. Derr, Y. Ma, and J. Tang, "Signed graph convolutional networks," in *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 929–934, IEEE, 2018.
- [46] C. Chen, M. Zhang, and Y. Liu, "Neural attentional rating regression with review-level explanations," in *Proceedings of the 2018 World Wide Web Conference*, pp. 1583–1592, 2018.
- [47] C. Fan, H. Xu, H. Gao, X. Jiaoxiong, and W. Wei, "TRG-DAtt: the target relational graph and double attention network based sentiment analysis and prediction for supporting decision making," *ACM Transactions on Management Information Systems (TMIS)*, 2020.
- [48] Y. Huang, H. Xu, H. Gao, and W. Hussain, "SSUR: an approach to optimizing virtual machine allocation strategy based on user requirements for cloud data center," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 2, pp. 670–681, 2021.