# DPRec: Social Recommendation Based on Dynamic User Preferences

Haibo Hu, Dan Yang, Yu Zhang

Abstract—Social recommendation learns users' preferences by integrating social information and interaction information to complete the recommendation task. In recent years, social recommendation has begun to model high-order neighbors of users to learn their preferences. However, existing social recommendation models rarely model dynamic user preferences, ignoring the similarity relationships between items and the inconsistency problem of neighbors. Therefore, we propose a social recommendation algorithm based on dynamic user preferences (DPRec), which models dynamic user preferences through a RNN. we use the similarity coefficient to calculate similarity relationships between items and constructs an item similarity network graph. At the same time, DPRec dynamically samples neighbor information of nodes (users and items) to reduce the impact of social inconsistency neighbors on recommendation performance. Based on the different influences of different neighbors, DPRec uses an attention mechanism to learn the influence factors of different neighbors on the target node. The experiments on two public datasets reveal the importance of dynamic user preferences and verify the effectiveness of the algorithm.

*Index Terms*—Dynamic User Preference; Graph Neural Network; Attention Mechanism; Social Recommendation; Item Similarity Network

#### I. INTRODUCTION

Most existing recommendation systems suffer from the cold-start problem [1-2]. To address this issue, social recommendation systems incorporate social information among users as auxiliary information to user-item interactions. Previous research [3] has shown that two users with social connections can influence each other. This means that users with social connections have similar interests and are considered useful auxiliary information in recommendation systems [4]. Therefore, integrating users' social information into recommendation system can effectively improve recommendation performance. Social recommendation simultaneously uses user-item interactions and users' social connections to recommend items to target users. Moreover, leveraging social connections can effectively understand users' preferences due to homophily

Manuscript received March 9,2023; revised Jul 15,2023. This work was supported by the General Scientific Research Project of Liaoning Provincial Department of Education (LJKMZ20220646).

Haibo Hu is a postgraduate student at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (e-mail: ashubb@163.com).

Dan Yang is a professor at School of computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (corresponding author to provide e-mail: asyangdan@163. com).

Yu Zhang is a senior engineer at China Telecom Digital Intelligence Technology Co., Ltd,Beijing,China (e-mail: zhangy193@chinatelecom.cn). and social influence. Therefore, social recommendation has received increasing attention.

Recently, GNNs have rapidly developed in social recommendation and other recommendation systems [5-7] and have been shown to be of significant importance in graph representation learning [8]. In social recommendation, both the user social network and the interaction network can be represented as graphs, representing the user's friend relationships and the user's interaction relationships with items, respectively. Through these two types of user relationships, user preferences can be analyzed from different perspectives. Therefore, social recommendation systems can use GNNs to iteratively aggregate neighbor feature information, better learn user features and item features, and complete recommendation tasks. However, current social recommendation algorithms based on GNNs still face the following issues:

- Existing research has ignored the dynamic nature of user preferences. The essence of user preferences is dynamic and constantly changing over time. For instance, a user may like smart devices at time t1, but be more interested in music-related items at time t2. This change reflects the dynamic nature of user preferences, and learning this change process is more beneficial for representing user features.
- Existing research has seldom taken into account the similarity between items. Most of the existing social recommendation systems only utilize user social networks, while ignoring the similarity network among items. For instance, users who have purchased Huawei smartphones may prioritize buying other Huawei products (such as headphones, tablets, or computers) because of their shared attributes as Huawei branded products. Therefore, incorporating the similarity network among items can enhance the feature representation of items in social recommendations.
- The inconsistency of neighbors has been overlooked in existing research [29]. Social relationships may not accurately reflect the results of rating prediction, and aggregating inconsistent neighbor information can affect node feature representation and result in decreased recommendation performance. The inconsistency of neighbors is mainly divided into contextual inconsistency and relationship inconsistency. Contextual inconsistency means that users may have different preferences for items. As shown in Figure 1, interactive items of user u1 are all related to sports (table tennis, football), while interactive items of user u2 are all related to clothing. Therefore, at the contextual level, users u<sub>1</sub> and u<sub>2</sub> are inconsistent neighbors. In addition, when modeling user social network graphs and interaction network graphs,

multiple relationships may occur. In addition to social relationships, inconsistent neighbors can also be distinguished by users' ratings of items. In Figure 1, user  $u_2$  and user  $u_3$  are social neighbors, and both have rated the same item. User  $u_2$  likes the item (rating 5), while user  $u_3$  dislikes the item (rating 1). This leads to relationship inconsistency. Although users  $u_2$  and  $u_3$  have a social relationship, their corresponding item preferences are not consistent.



Fig. 1. An example of inconsistent neighbors

we propose a social recommendation algorithm based on dynamic user preferences, called DPRec, which can better model users and items. DPRec builds three graphs (social network graph, interaction network graph, and similarity network graph). The social network graph and interaction network graph provide user information from many perspectives, while item similarity network graph and interaction network graph contain information helpful for modeling items.In addition, we use LSTM to model dynamic user preferences and dynamically sample neighbor information through graph neural network. We use an attention mechanism to assign different impact factors to different neighbors, reducing the impact of inconsistent neighbors on recommendation performance. The main contributions of the paper are summarized as follows:

- we propose a social recommendation algorithm based on dynamic user preference (DPRec), which utilizes LSTM to model dynamic user preferences.
- The proposed social recommendation algorithm (DPRec) considers the similarity relationships between items and constructs an item similarity network to enrich item information. The dynamic sampling module is used to sample neighbors, which reduces the effect of neighbor inconsistency on recommendation performance.
- we conducted extensive experiments on two public datasets, and the results show that DPRec outperforms other baseline methods, confirming the effectiveness and feasibility of DPRec.

#### II. RELATED WORK

This chapter provides a detailed introduction to the important literature relevant to the paper, mainly divided into two types: traditional social recommendation methods and methods based on graph neural networks.

#### A. Traditional social recommendation methods

With the increase of social software and mobile users, introducing social information into recommendation systems has become a trend, and existing research has proven that social information can improve recommendation performance [8]. Social recommendation

can use social relationships to alleviate the data sparsity and cold start problems in recommendation systems. The traditional social recommendation algorithms mainly used matrix factorization technology, which fills the blank data in the matrix through machine learning methods to complete the recommendation task. Reference [9] introduces a trust propagation mechanism, where the user's features are related to information of their neighbors in the social network, meaning that two users who have a social relationship have similar features. Reference [10] designs two regularization terms based on social information to constrain the objective function of matrix factorization. Reference [11] introduces social information into the SVD++ [12] model, considering the explicit and implicit influence of both social information and rating information, and modeling users and items using weighted regularization techniques.

#### B. Methods based on graph neural networks (GNNs)

The main function of the graph neural network in social recommendation is to aggregate the neighbor information of the target node in the social network graph and the interaction network graph. In social recommendation, the user interaction network and user social network are represented as graphs, and GNN technology is used to iteratively aggregate data in the graph to obtain representations of different nodes in the graph and complete the recommendation task. Reference <sup>[13]</sup> was the first to apply GNN to social recommendation, iteratively aggregating neighbor nodes in interaction network graph and social network graph, and using attention mechanisms to assign different weights to neighbors to learn user features. However, reference [13] only introduced social information among users and ignored the correlation between items. To address this issue, reference [14] introduced a similar network of items on the basis of GraphRec [13] to enrich item information. Reference [15] is an algorithm based on SVD++, which performs average pooling on adjacent nodes in user interaction network graph and iteratively aggregates high-order neighbor node information of users using GNN in user social network graph. Reference [16] is an extension of reference [15], considering that different nodes have different effects on the target node, and using graph attention mechanism to simulate different influences between user interests and social users, improving recommendation performance. Both reference [16] and reference [15] use hierarchical propagation mechanisms to simulate the dynamic diffusion process of social relationships. Reference [18] proposes a dual graph attention network [29] to learn deep implicit representations of double social effects, where one attention weight model is specific to user features and the other models dynamic, attention weights.

Traditional methods mainly rely on matrix factorization, and the information in social graphs and interaction graphs cannot be fully utilized. In order to solve the above problems, we utilize GNNs to aggregate neighbor information. We propose a social recommendation algorithm, DPRec, based on dynamic user preferences. Compared with previous methods, DPRec models user dynamic preferences and constructs a network graph of item similarities, reducing the impact of inconsistent neighbors on recommendation performance.

#### III. PRELIMINARIES

This chapter mainly introduces some important notations and related definitions used in the paper, and provides a brief overview of the problems.

#### A. Definitions and Notations

Social recommendation mainly includes two sets, we define the user set  $U = \{u_1, u_2, u_3, ..., u_n\}$  and the item set  $V = \{v_1, v_2, v_3, ..., v_m\}$ , n and m represent the number of users and items, respectively. The specific symbols are defined as shown in Table I.

TABLE I	
SYMBOL DEFINITION	

SYMBOL DEFINITION					
Symbols Definitions					
r <sub>ij</sub>	rating of item v <sub>j</sub> by user u <sub>i</sub>				
R	rating matrix				
$G_u$	social network graph				
$G_v$	similarity network graph				
G <sub>R</sub>	interaction network graph				
Eu	set of edges in social network graph				
$\epsilon_{\rm v}$	set of edges in similarity network graph				
$\mathbf{p}_{i}$	the embedding of user ui				
$q_j$	the embedding of item v <sub>j</sub>				
e <sub>ij</sub>	the embedding for rating				

Definition 1: User social network graph. User social network graph is defined as Gu={U, $\mathcal{E}_u$ },  $\mathcal{E}_u$  represents the set of user social relationships. For example,  $(u_i,u_j)\in\mathcal{E}_u$  represents that user  $u_i$  and user  $u_j$  have a social relationship.

Definition 2: Item similarity network graph. Item similarity network graph is defined as  $G_v = \{V, \mathcal{E}_v\}, \mathcal{E}_v$  represents the set of item similarity network. For example,  $(v_i, v_j) \in \mathcal{E}_v$  represents that item vi and item  $v_j$  are similar items.

Definition 3: Interaction network graph. Interaction network graph is defined as  $G_R = \{U, V, R\}$ , R is rating matrix, defined as  $R_{(n \times m)} = \{r_{ij} | i \in U, j \in V\}$ . For example, the rating of user  $u_i$  on item  $v_j$  is defined as  $r_{ij}$ ,  $r_{ij}$  represents the degree of user  $u_i$ 's liking for item  $v_j$ .

### B. Problem Description

The goal of this paper is to improve the user-item rating matrix and predict the rating score  $\dot{r_{ij}}$  for any user  $u_i \in U$  on the non-interacted item  $v_i \in V$ .

Input: three kinds of graphs (G<sub>v</sub>, G<sub>R</sub> and G<sub>u</sub>).

Output: the value of edge that connects user  $u_i$  and item  $v_{j\cdot}$ 

## IV. THE PROPOSED FRAMEWORK

This chapter presents the proposed recommendation algorithm DPRec, which is illustrated in Figure 2. DPRec consists of five parts: 1) Graph construction, which preprocesses the data to construct interaction network graph, social network graph, and similarity network graph using Jaccard similarity coefficient. 2) Feature embedding, which uses the embedding layers to obtain user embedding representations, item embedding representations, and rating embedding representations, and uses a MLP to obtain node interaction embedding representations, user social embedding representations, and item similarity embedding representations. 3) Node static modeling, which aggregates the neighbor information of nodes through a dynamic neighbor sampling module and uses attention mechanisms to assign different influence factors to different neighbors to obtain static representations of nodes. 4) Node dynamic modeling, which uses LSTM to model the changes in user interests and item attractiveness over time to obtain the dynamic representations of nodes. 5) Rating prediction, which combines the static and dynamic representations of nodes to obtain the feature representations of nodes and ultimately completes the recommendation task.

# A. Graph construction

DPRec first needs to construct interaction network graph  $G_R$ , social network graph  $G_u$ , and similarity network graph  $G_v$ . The interaction network graph and social network graph can be directly constructed from the raw data. The item similarity network graph uses the interaction information in  $G_R$  to associate two similar items. Considering that the number of interactions for each item may be different, we use the similarity function to calculate the similarity between each pair of items. The calculation of item similarity is as follows:

$$S(i,j) = \frac{G_R(i) \cap G_R(j)}{G_R(i) \cup G_R(j)}$$
(1)

S(i,j) represents the similarity between  $v_i$  and  $v_j$ , and  $G_R(i)$  represents all users who have interacted with item  $v_i$ . We sets S(i,j) > 0.5 to indicate the existence of a similarity relationship between item  $v_i$  and item  $v_j$ , and constructs an item similarity network graph based on the similarity relationships between items.

#### B. Feature embedding

The interaction network diagram not only includes the interactions between users and items but also includes the user's ratings of items. These ratings reflect the user's static preferences at that time and the static attractiveness of the item, which is helpful for modeling users and items. The user's interaction embedding representation is obtained by concatenating user embedding and the rating embedding. The item's interaction embedding representation is obtained by concatenating item embedding and the rating embedding of the interacting user. The detailed is as follows:

$$U_{ij} = MLP([e_{ij} \oplus p_i])$$
(2)  
$$V_{ij} = MLP([e_{ij} \oplus q_i])$$
(2)

$$\varphi_{ji} = MLP([e_{ij} \oplus q_j])$$
(3)

 $U_{ij}$  and  $V_{ji}$  are the interaction embedding representations of  $u_i$  and  $v_j$ ,  $p_i$  and  $q_j$  are the embedding representations of  $u_i$  and  $v_j$ ,  $e_{ij}$  is the embedding of the rating of  $u_i$  on  $v_j$ , and  $\bigoplus$  denotes the concatenation operation of two embedding representations. MLP refers to a two-layer perceptron.

The user social network graph contains the social relationships among users, while the item similarity network graph contains the similarity relationships among items. Both types of relationships are helpful for modeling nodes. Soc<sub>ij</sub> is the social embedding representation of users  $u_i$  and  $u_j$ , and  $Sim_{ji}$  is the similarity embedding representation of items  $v_i$  and  $v_j$ . The specific calculation formulas for Soc<sub>ij</sub> and Sim<sub>ji</sub> are as follows:

$$Soc_{ij} = MLP([p_i \oplus p_j])$$
 (4)

$$Sim_{ji} = MLP([q_i \oplus q_j])$$
(5)



Fig. 2. Overview of DPRec

In addition the aforementioned embedding to representations, we also generate query embedding representations by mapping the embedding user representations and item embedding representations. The specific calculations are as follows:

$$Q_{i,j} = \sigma(W_Q^T(q_j \oplus p_i))$$
(6)

 $Q_{(i,j)}$  is the query embedding representation of  $u_i$  and  $v_j$ ,  $W_Q \epsilon R^{(2d \times d)}$  is the mapping matrix of the query layer, and  $\sigma$  is an activation function.

# C. Node static modeling

By iteratively aggregating the interactive items and social friends (users connected to the target user in G<sub>u</sub>), the static representation of the user can be obtained. By iteratively aggregating the users who rate the item and the similar items, the static representation of the item can be obtained. When users purchase items, they may ask their social friends who have purchased similar items for advice and decide whether to purchase the item based on their friends' suggestions. The role of the dynamic neighbor sampling module is to sample this subset of neighboring nodes (nodes with interactive relationships with the target node). Therefore, we apply the dynamic neighbor sampling module to the graph neural network, dynamically sampling different neighbors based on different target nodes to reduce the impact of inconsistent neighbors on recommendation performance. For node v, the sampling probability of neighbor node i is expressed as follows:

$$prob(i) = s(i,Q) / \sum_{j \in N_v} s(j,Q)$$
(7)

prob(i) represents the sampling probability of neighbor node i of node v. s(i, Q) is the similarity score between the embedding representation  $h_i \epsilon(p_i, q_i)$  of neighbor node i and the query embedding representation Q, and the detailed is as follows:

$$s(i,Q) = exp(-||Q - h_i||_2^2)$$
(8)

We sets a hyperparameter  $\lambda(0 < \lambda \le 1)$  to represent the sampling rate and dynamically sample neighboring nodes based on the sampling rate. Therefore, if a node is connected to more nodes, we will sample more neighboring

nodes. After neighbor sampling, we use GNNs to iteratively aggregate neighboring nodes and learn the static representations of nodes.

For the interaction network graph, we iteratively aggregate the sampled neighbor information and uses attention mechanism to assign weights to neighbors to obtain the static representations of nodes. The specific calculation formulas for  $h_i^{ui}$  and  $h_i^{ui}$  are as follows:

$$h_i^{ui} = \sigma \left( \omega_0 \cdot \sum_{j \in Samp_i^{ui}} \alpha_{ij} \cdot U_{ij} + b_0 \right)$$
(9)

$$h_{j}^{ui} = \sigma \left( \omega_0 \cdot \sum_{i \in Samp_j^{ui}} \alpha_{ji} \cdot V_{ji} + b_0 \right)$$
(10)

 $h_i^{ui}$  and  $h_j^{ui}$  are the static representations of  $u_i$  and  $v_j$  in interaction network graph.  $Samp_i^{ui}$  is the sampled neighbor set for the interaction items of  $u_i$ , and  $Samp_j^{ui}$  is the sampled neighbor set for the rating users of  $v_j$ .  $\omega_0$  and  $b_0$  represent the weight matrix and bias term of neural network.  $\sigma$  is a non-linear activation function.  $\alpha_{ij}$  represents the attention weight of the interaction item  $v_j$  in user  $u_i$ , and  $\alpha_{ji}$  represents the attention weight of the rating  $u_i$  in  $v_j$ . The specific calculation formulas for  $\alpha_{ij}$  and  $\alpha_{ji}$  are as follows:

$$\alpha_{ij} = \frac{exp(\alpha_{ij}^*)}{\sum_{j \in Samp_i^{ui}} exp(\alpha_{ij}^*)}$$
(11)

$$\alpha_{ij}^* = \omega_1 \cdot \sigma \big( \omega_2 \cdot \big[ p_i \bigoplus U_{ij} \big] + b_2 \big) + b_1 \quad (12)$$

$$\alpha_{ji} = \frac{\exp(\alpha_{ji}^{*})}{\sum_{i \in Samp_{j}^{ui}} \exp(\alpha_{ji}^{*})}$$
(13)

$$\alpha_{ji}^* = \omega_1 \cdot \sigma(\omega_2 \cdot [q_j \bigoplus V_{ji}] + b_2) + b_1 \quad (14)$$

According to related research <sup>[19,20]</sup>, users' preferences are influenced by social friends, that is, users with social relationships have similar preferences. In addition, two similar items may have the same appeal to the target user. Therefore, social users and similar items can be aggregated in the user social network graph and item similarity network graph to obtain the static representation of nodes in the graph. Considering that different neighboring nodes have different impacts on the target node, we use attention mechanism to assign different influence factors to neighboring nodes. The specific calculation is as follows:

$$h_i^{soc} = \sigma \left( \omega_0 \cdot \sum_{j \in Samp_i^{soc}} \alpha_{ij}^{soc} \cdot Soc_{ij} + b_0 \right)$$
(15)  
$$h_j^{sim} = \sigma \left( \omega_0 \cdot \sum_{i \in Samp_j^{sim}} \alpha_{ji}^{sim} \cdot Sim_{ji} + b_0 \right)$$
(16)

 $h_i^{soc}$  represents the static representation of user  $u_i$  in the social network graph.  $h_j^{sim}$  represents the static representation of item  $v_j$  in the item similarity network graph. Samp\_i^{soc} and Samp\_j^{sim} represent the sets of neighbors for user  $u_i$  and item  $v_j$  in the social network graph and similarity network graph after sampling.  $\alpha_{ij}^{soc}$  represents the attention weight of  $u_j$  in  $u_i$ , and  $\alpha_{ji}^{sim}$  represents the attention weight of  $v_j$  in  $v_j$ .

$$\alpha_{ij}^{soc} = \frac{exp(\alpha_{ij}^*)}{\sum_{j \in Samp_i^{soc}} exp(\alpha_{ij}^*)}$$
(17)

$$\alpha_{ij}^* = \omega_1 \cdot \sigma \left( \omega_2 \cdot \left[ p_i \bigoplus_{evn} Soc_{ij} \right] + b_2 \right) + b_1 \quad (18)$$

$$\alpha_{ji}^{sim} = \frac{\exp\left(\alpha_{ji}\right)}{\sum_{i \in Samp_j^{Sim}} \exp\left(\alpha_{ji}^*\right)}$$
(19)

$$\alpha_{ji}^* = \omega_1 \cdot \sigma(\omega_2 \cdot [q_j \oplus Sim_{ji}] + b_2) + b_1 \quad (20)$$

We combine the user's personal preference  $h_i^{ui}$  with the preference influenced by social friends  $h_i^{soc}$  to obtain the static representation of the user. At the same time, we combine the self-attractiveness of the item  $h_j^{ui}$  with the attractiveness influenced by similar items  $h_j^{sim}$  to obtain the static representation of the item. The specific calculation is as follows:

$$h_i^S = g^u \left( \left[ h_i^{soc} \bigoplus h_i^{ui} \right] \right)$$
(21)

$$h_j^{\mathcal{S}} = g^{\nu}([h_j^{\mathcal{S}^{lm}} \oplus h_j^{ul}])$$
(22)

 $h_i^S$  and  $h_j^S$  represent the static representations of  $u_i$  and  $v_j$ .  $g^u$  and  $g^v$  are two-layer perceptrons.

# D. Node dynamic modeling

Using static features alone cannot effectively reflect user preferences. User interests are not static, but change. The attractiveness of an item also changes over time for a specific user. We utilize LSTM to model the interaction sequences for each time period, capturing the dynamic preferences (attraction) of users (items) in each time period.At time period T, given a sequence of k interaction items  $inter_i^u = \{v_1, v_2, ..., v_k\}$  for user u<sub>i</sub> and a sequence of k evaluation users  $inter_j^v = \{u_1, u_2, ..., u_k\}$  for item v<sub>j</sub>, the corresponding item embedding sequence  $Seq_i^u =$  $\{U_{i1}, U_{i2}, ..., U_{ik}\}$  and user embedding sequence  $Seq_i^v$  are  $\{v_{j1}, v_{j2}, ..., v_{jk}\}$ ) are obtained. Finally,  $Seq_i^u$  and  $Seq_j^v$ are passed to the LSTM to learn the dynamic preferences and attraction. The specific calculation is as follows:

$$h_i^D = LSTM(Seq_i^u)$$
(23)  
$$h_i^D = LSTM(Seq_i^v)$$
(24)

 $h_j^{D} = LSIM(Seq_j)$  (24)  $h_i^{D}$  and  $h_j^{D}$  represent the dynamic preferences of  $u_i$  and the dynamic attraction of  $v_j$ .  $Seq_i^{u}$  and  $Seq_j^{v}$  are both sorted sequence information by time.  $Seq_i^{u}$  contains the interaction items and rating information of user  $u_i$ .  $Seq_j^{v}$ contains the rating users and corresponding rating information for item  $v_j$ . The specific calculation of  $Seq_i^{u}$ and  $Seq_j^{v}$  is as follows:

$$Seq_i^u = \left(U_{ij}\right)_{i \in inter_i^u} \tag{25}$$

$$Seq_j^{\nu} = \left(V_{ji}\right)_{i \in inter_i^{\nu}}$$
(26)

*inter*<sup>*u*</sup><sub>*i*</sub> is the k interaction items of user u<sub>i</sub> sorted by time, and *inter*<sup>*v*</sup><sub>*j*</sub> is the k evaluation users of item v<sub>j</sub> sorted by time. In this paper, k is set to 30.

# E. Rating prediction

We combine the static representation and dynamic representation of the nodes to obtain the representation of nodes. The specific calculation formulas for  $h_i^U$  and  $h_j^V$  are as follows:

$$h_i^U = g\left(\left[h_i^D \oplus h_i^S\right]\right) \tag{27}$$

$$h_j^V = g([h_j^D \oplus h_j^S])$$
(28)

 $h_i^U$  and  $h_j^V$  represent the user representation and item representation, respectively. g is a two-layer perceptron.

The DPRec is applied to rating prediction, and the score  $\mathbf{r}'_{ij}$  of  $u_i$  for  $v_j$  is calculated as follows:

$$\dot{r_{ij}} = h_i^U \odot h_j^V \tag{29}$$

# F. Training of the recommendation model

DPRec is to predict user ratings for items, therefore we choose MSELoss to optimize the model parameters. The specific calculation of MSELoss is shown in equation (30):

$$MSELoss = \frac{1}{2|N|} \cdot \sum_{(i,j) \in N} (r'_{ij} - r_{ij})^2 + \delta ||\theta||^2$$
(30)

N represents the number of interactions in dataset,  $|||\theta||^2$  represents regularization, and  $\delta$  is used to control its weight.

To optimize the loss function, we use the Adam optimizer [21]. The learning rate parameter in Adam is insensitive and has strong robustness. Meanwhile, we use Dropout to prevent overfitting. The strategy of Dropout is to randomly drop out some neurons, update only a portion of neurons, and only work in the training set.

# V. EXPERIMENTS

This chapter introduces the data sets and evaluation indicators required for the experiment. We conduct a large number of experiments, and provide a detailed analysis of the experimental results.

TABLE II				
STA	ATISTICS OF THE DAT.	ASETS		
Dataset	Ciao	Epinions		
#Users	2379	22167		
#Items	16862	296278		
#Ratings	35990	920073		
#Relations	57544	355813		
#Avg. events/user	15.12	41.50		
#Avg. friends/user	24.18	16.05		

# A. Experimental Settings

1) Datasets

We use two publicly available datasets extracted from two real shopping websites: Ciao and Epinions [27]. These datasets are primarily used for social recommendations based on rating prediction. Ciao and Epinions are both well-known shopping websites that allow any user to rate products. Users can also add other users to their friend lists and establish social networks. Detailed data in two datasets are shown in Table II.

### 2) Evaluation Metrics

The main task of this paper is to predict the user's evaluation score for the item. Therefore, MAE and RMSE [22] are chosen as evaluation metrics. The smaller the

PERFORMANCE COMPARISON OF DPREC AND OTHER BASELINE METHODS							
Algorithms	Ciao		Epinions				
	MAE	RMSE	MAE	RMSE			
SoRec	0.8410	1.0652	0.8961	0.9119			
SoReg	0.8611	1.0848	0.9119	1.1703			
TrustMF	0.7681	1.0432	0.8396	1.1365			
SocialMF	0.8270	1.0501	0.8837	1.1328			
DeepSoR	0.7739	1.0316	0.8383	1.0972			
DGRec	0.8029	0.9943	0.8511	1.0684			
GraphRec	0.7486	0.9894	0.8123	1.0673			
GraphRec+	0.7431	0.9750	0.8113	1.0627			
ConsisRec	0.7376	0.9744	0.8108	1.0611			
HOSR	0.7372	0.9740	0.8106	1.0605			
DPRec	0.6631	0.8791	0.7976	1.0553			

TABLE III

evaluation indicators, the better the performance. A slight improvement in the indicators may have a significant impact on the prediction results [23]. The specific calculation formulas for MAE and RMSE are as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |\dot{r}_{ij} - r_{ij}|$$
(31)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (r_{ij} - r_{ij})^2}{n}}$$
(32)

 $r_{ii}$  represents the evaluation scores of u<sub>i</sub> for v<sub>j</sub>, and r<sub>ij</sub> represents the actual evaluation score of ui for vi.

3) Baselines

To evaluate the effectiveness of the model, we compare DPRec with two other types of baseline methods: social recommendation methods based on MF and social recommendation methods based on GNNs.

Social recommendation methods based on MF:

SoRec [25]: it decomposes the interaction matrix and the user social matrix.

SoReg [10]: it uses regularization to model social networks.

TrustMF [9]: it decomposes the user trust network, mapping users to trust and trusted spaces.

SocialMF [2]: It incorporates a trust propagation mechanism into the original matrix factorization framework.

DeepSoR [24]: it uses deep neural networks to learn the nonlinear features of each user from social relationships.

Social recommendation methods based on GNN: DGRec [17]: it models social influence and user dynamic interests using graph attention mechanisms.

GraphRec [13]: it constructs user interaction graphs and social graphs to better learn user features.

GraphRec+ [14]: It adds item-item correlation graphs on the basis of the GraphRec model to better learn item features.

HOSR [26]: It models users' high-order social relationships using graph convolutional networks.

ConsisRec [28]: It solves the problem of inconsistent neighbors by sampling consistent neighbors based on the consistency score calculated between neighboring nodes and the target node.

# 4) Parameter Settings

The important parameters in DPRec are learning rate, neighbor aggregation ratio, feature embedding dimension, and node sequence length in node dynamic modeling. The learning rate in the optimizer Adam has a small impact on recommendation results, so we set learning rate to the default value of 0.001. The neighbor aggregation ratio takes

values in the set {0.5, 0.6, 0.7, 0.8, 0.9, 1}, the embedding dimension takes values in the set {8, 16, 32, 64, 128, 256}, and the interaction sequence length takes values in the set  $\{10, 20, 30, 40, 50\}$ . The parameters in the Dropout layer are set to 0.5 to prevent overfitting. At the same time, we add an early termination command, that is, the training stops if the evaluation metric does not improve for five consecutive times.

#### Recommendation performance evaluation B

We conducted extensive experiments on the Ciao and Epinions datasets and compared them with other baseline methods. Table III includes the experimental results of all methods. The results show that the performance of the baseline methods based on GNNs is always better than that of the baseline methods based on MF, which proves that the role of GNNs is more significant in social recommendation. both datasets, DPRec proposed in this paper In outperformed all baseline methods, while HOSR achieved the best performance among all baseline methods. In Ciao dataset, DPRec improved the MAE and RMSE evaluation metrics by 10.05% and 9.49%, respectively. In the Epinions dataset, DPRec improved the MAE and RMSE evaluation metrics by 1.17% and 0.52%, respectively. Therefore, the DPRec algorithm that simultaneously considers dynamic user preferences, item similarity relationships, and neighbor inconsistency problems can better model users and items and complete the rating prediction recommendation task.

#### Model Analysis С.

We will investigate the effectiveness of different modules in DPRec and the influence of important parameters, and conduct corresponding ablation experiments and parameter sensitivity analysis.

# 1) Ablation study

The existing graph neural network-based baseline methods either ignore the problem of inconsistent neighbors, or ignore the dynamic changes of user preferences, or ignore the similarity relationships between items. Therefore, we remove some modules from DPRec to obtain three variant methods, as follows:

(A)DPRec-Item, ignored the similarity between items.

(B)DPRec-Dym, ignored the dynamic changes in user preferences.

(C)DPRec-Smp, ignored the issue of inconsistent neighbors.

The experimental results of DPRec and its variant methods A, B, and C are shown in Figure 3, where the horizontal axis represents DPRec and its variant methods,





and the vertical axis represents the evaluation metrics. From Figure 3, it can be observed that DPRec achieves the best performance on both datasets, and methods A, B, and C also outperform other baseline methods. These two points demonstrate that the neighbor sampling module, dynamic user preferences, and item similarity play important roles in improving recommendation performance.

# 2) Impact of embedding dimension

This section studies the impact of feature embedding dimensions on DPRec, and Figure 4 shows the experimental results of DPRec with different embedding dimensions on the Ciao and Epinions datasets. The x-axis represents the feature embedding dimensions, and the yaxis represents the evaluation metrics. In other words, when the dimensions of the feature embeddings are increased, the performance of DPRec initially improves, but eventually starts to decline in both datasets. When the embedding dimension is increased from 8 to 128, the performance of DPRec gradually increases and reaches the optimal performance. When the embedding dimension is greater than 128, the performance of DPRec gradually decreases. The experimental results show that increasing the embedding dimensions can improve the recommendation performance. However, when the embedding dimension is too high, it not only increases the complexity of DPRec but also reduces its operating efficiency. Therefore, this study chooses 128 as the feature embedding dimension, which ensures both performance and reduces training time.

# 3) Impact of sampling ratio

We will investigate the impact of sampling ratio on DPRec. Figure 5 shows the experimental results of DPRec with different neighbor sampling ratios on the Ciao and Epinions datasets. The horizontal axis represents the sampling ratio, and the vertical axis represents the evaluation metric. In Ciao dataset, the number of neighbors for each user is relatively small, and the social relationships are dense, making the impact of the sampling ratio more significant. With increasing sampling ratio, DPRec performance exhibits an initial improvement followed by a subsequent decline. The performance of DPRec reaches the optimal value when the sampling ratio is 0.8. When the sampling ratio is greater than 0.8, the noise from the neighbors has a greater impact on DPRec, and the performance of DPRec gradually decreases. In Epinions dataset, the number of neighbors for each user is relatively large, and social relationships are sparse, resulting in a weaker impact of the sampling ratio on the experimental results. In Epinions dataset, the performance of DPRec reaches the optimal value when the sampling ratio is 0.6. When the sampling ratio is greater than 0.6, the value of experimental results gradually increases, and performance of DPRec gradually decreases. When the sampling ratio is less than 0.6, the value of the experimental results gradually decreases, and the performance of DPRec gradually improves. The impact of neighbor sampling ratio on DPRec varies with different datasets.

# 4) Impact of sequence length

This section discusses the impact of the node interaction sequence length k on DPRec in dynamic node modeling. Keeping other parameters constant, the performance of DPRec is observed by adjusting the sequence length. Figure 6 shows experimental results of DPRec in the Ciao and Epinions datasets with different sequence lengths, with the x-axis representing the sequence length and the y-axis representing the evaluation metric. Based on experimental results shown in Figure 6, it can be observed that the node interaction sequence length k has a significant impact on the performance of DPRec. In both the Ciao and Epinions datasets, when the sequence length is 30, the performance of DPRec reaches its optimal value; when the sequence length is less than 30, the performance of DPRec gradually improves; when the sequence length is greater than 30, the performance of DPRec gradually deteriorates. Therefore, in this paper, the node interaction sequence length is set to 30 to achieve the best performance of DPRec.



#### VI. CONCLUSION AND FUTURE WORK

We study existing social recommendation methods and propose a social recommendation algorithm based on dynamic user preferences (DPRec), which models dynamic user preferences using LSTM. To address the issue of insufficient item information, DPRec constructs an item similarity network graph and combines it with interaction network graph to jointly model items. In addition, DPRec dynamically samples neighbor nodes, reducing the impact of neighbor inconsistencies. In comparison to existing methods, DPRec demonstrates superior performance across Ciao and Epinions datasets, and experimental findings affirm the efficacy of its dynamic user preference, item similarity, and neighbor sampling modules.

In future work: 1) When aggregating node neighbor information, investigate how to utilize higher-order neighbors to improve recommendation performance. 2) When modeling nodes, study how to distinguish different types of interaction information (purchases, browsing, etc.) and self-attribute information (user age, gender, item category, etc.) to enhance the interpretability of the recommendation system.

#### REFERENCES

- Zhu Y, Lin J, He S, et al. Addressing the Item Cold-Start Problem by Attribute-Driven Active Learning[J]. IEEE, 2020, 32(4): 631-644.
- [2] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In Proceedings of the fourth ACM conference on Recommender systems. ACM, 135–142.
- [3] Cialdini R B , oldstein N J . Social Influence: Compliance and Conformity[J]. Annual Review of Psychology, 2004, 55(1):591-621.
- [4] Liu H, Jing L, Yu J, et al. Social Recommendation With Learning Personal and Social Latent Factors[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, PP(99):1-1.
- [5] Wenhao Pan, and Kai Yang. Enhanced Multi-Head Self-Attention Graph Neural Networks for Session-based Recommendation. Engineering Letters, vol. 30, no.1, pp37-44, 2022.
- [6] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019. ACM, 417–426.
- [7] Wenqi Fan, Yao Ma, Qing Li, Jianping Wang, Guoyong Cai, Jiliang Tang, and Dawei Yin. 2020. A Graph Neural Network Framework for Social Recommendations. IEEE Transactions on Knowledge and Data Engineering (2020).
- [8] Tyler Derr, Yao Ma, and Jiliang Tang. 2018. Signed Graph Convolutional Networks. In 2018 IEEE International Conference on Data Mining (ICDM). IEEE, 929–934.
- [9] B. Yang, Y. Lei, J. Liu, W. Li, Social collaborative filtering by trust, IEEE Trans. Pattern Anal. Mach. Intell. 39 (8) (Aug 2017) 1633– 1647.
- [10] Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: The 4th ACM International Conference on Web Search and Data Mining. pp. 287–296 (2011).
- [11] Guibing Guo, Jie Zhang, Neil Yorke-Smith, Trustsvd: Collaborative

filtering with both the explicit and implicit influence of user trust and of item ratings, Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15, AAAI Press, 2015, pp. 123–129.

- [12] Yehuda Koren, in Factorization meets the neighborhood: A multifaceted collaborative filtering model In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA, 2008.
- [13] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019. ACM, 417–426.
- [14] Wenqi Fan, Yao Ma, Qing Li, Jianping Wang, Guoyong Cai, Jiliang Tang, and Dawei Yin. 2020. A Graph Neural Network Framework for Social Recommendations. IEEE Transactions on Knowledge and Data Engineering (2020).
- [15] Wu. Le, Peijie Sun, Fu. Yanjie, Richang Hong, Xiting Wang, Meng Wang, A neural influence diffusion model for social recommendation, in Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, Falk Scholer (Eds.), Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019 ACM, July 21–25, 2019, Paris, France, 2019, pp. 235–244.
- [16] L. Wu, J. Li, P. Sun, R. Hong, Y. Ge, M. Wang, DiffNet++: A Neural Influence and Interest Diffusion Network for Social Recommendation, IEEE Trans. Knowl. Data Eng. (2020), 1-1.
- [17] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. Session-based social recommendation via dynamic graph attention networks. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pages 555–563, 2019.
- [18] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In The World Wide Web Conference, pages 2091–2102, 2019.
- [19] Peter V Marsden and Noah E Friedkin. 1993. Network studies of social influence. Sociological Methods & Research 22, 1 (1993), 127–151.
- [20] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather:Homophily in social networks. Annual review of sociology 27, 1 (2001), 415–444.
- [21] Kingma D, Ba J. Adam: A Method for Stochastic Optimization[J]. Computer Science, 2014.
- [22] Wang, X., Hoi, S.C., Ester, M., Bu, J., Chen, C.: Learning personalized preference of strong and weak ties for social recommendation. In: The 26th International Conference on World Wide Web. pp. 1601–1610 (2017).
- [23] Wang, X., Hoi, S.C., Ester, M., Bu, J., Chen, C.: Learning personalized preference of strong and weak ties for social recommendation. In: The 26th International Conference on World Wide Web. pp. 1601–1610 (2017).
- [24] Wenqi Fan, Qing Li, and Min Cheng. 2018. Deep Modeling of Social Relations for Recommendation. In AAAI.
- [25] H. Ma, H. Yang, M. R. Lyu, and I. King. SoRec: Social recommendation using probabilistic matrix factorization. 17th ACM Conf. Inf. Knowl. Mining (CIKM), 2008, pp. 931–940.
- [26] Y. Liu, L. Chen, X. He, J. Peng, Z. Zheng, and J. Tang. Modelling high-order social relations for item recommendation. IEEE Trans. Knowl. Data Eng., vol. 34, no. 9, pp. 4385–4397, Sep. 2022.
- [27] J. Tang, H. Liu, H. Gao, and A. Das Sarmas. ETrust: Understanding trust evolution in an online world. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2012, pp. 253–26.
- [28] L. Yang, Z. Liu, Y. Dou, J. Ma, and P. S. Yu. ConsisRec: Enhancing GNN for social recommendation via consistent neighbor aggregation. 2021, arXiv:2105.02254.
- [29] Zhiwei Liu, Yingtong Dou, Philip S Yu, Yutong Deng, and Hao Peng. 2020. Alleviating the Inconsistency Problem of Applying Graph Neural Network to Fraud Detection. In Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020. ACM, 1569–1572.
- [29] Xinmiao Yu, Zhengpeng Li, Jiansheng Wu, and Mingao Liu. Multimodule Fusion Relevance Attention Network for Multi-label Text Classification. Engineering Letters, vol. 30, no.4, pp1237-1245, 2022.