Dual Side Deep Context-aware Modulation for Social Recommendation

Bairan Fu
fubr@smail.nju.edu.cn
Nanjing University
Nanjing, China

Wenming Zhang
zhangwm@smail.nju.edu.cn
Nanjing University
Nanjing, China

Guangneng Hu
njuhgn@gmail.com
Hong Kong University of Science and
Technology
Hong Kong, China

Xinyu Dai*
daixinyu@nju.edu.cn
Nanjing University
Nanjing, China

Shujian Huang
huangsj@nju.edu.cn
Nanjing University
Nanjing, China

Jiajun Chen
chenjj@nju.edu.cn
Nanjing University
Nanjing, China

ABSTRACT

Social recommendation is effective in improving the recommendation performance by leveraging social relations from online social networking platforms. Social relations among users provide friends' information for modeling users' interest in candidate items and help items expose to potential consumers (i.e., item attraction). However, there are two issues haven't been well-studied: Firstly, for the user interests, existing methods typically aggregate friends' information contextualized on the candidate item only, and this shallow context-aware aggregation makes them suffer from the limited friends' information. Secondly, for the item attraction, if the item's past consumers are the friends of or have a similar consumption habit to the targeted user, the item may be more attractive to the targeted user, but most existing methods neglect the relation enhanced context-aware item attraction.

To address the above issues, we proposed DICER (*Dual sIde deep Context-awarE modulation for social Recommendation*). Specifically, we first proposed a novel graph neural network to model the social relation and collaborative relation, and on top of high-order relations, a dual side deep context-aware modulation is introduced to capture the friends' information and item attraction. Empirical results on two real-world datasets show the effectiveness of the proposed model and further experiments are conducted to help understand how the dual context-aware modulation works.

CCS CONCEPTS

• **Information systems → Social recommendation**; **Recommender systems**.

KEYWORDS

recommender systems, social recommendation, graph neural networks, context-aware recommendation

*Corresponding author.

1 INTRODUCTION

With the rapid development of the Internet, information overload is becoming increasingly challenging in providing personalized information for users. Recommender systems designed to filter information and provide personalized recommendations play a vital role in various web services nowadays. Widely used recommendation methods are based on Collaborative Filtering (CF) techniques [14, 19, 20, 30], which mainly make use of the user-item interaction history, e.g., ratings, clickings. Moreover, along with the increasing popularity of social networking platforms, social recommendation, which incorporates social relations into recommender systems, has been developed and shows promising potential to improve the recommendation performance. Social relations among users can provide friends' information for modeling user preference better and also provide more possible perspectives for items' exposure to relevant users [28].

To incorporate social relations into the recommender system, existing social-aware recommendation methods make several attempts in various ways. Typical matrix factorization methods [11, 15, 16, 24] assume that users who have social relations may have a similar preference. Thus, these methods use social relations as a kind of social regularization to restrain the user embedding learning process. Moreover, some other methods [2, 10] consider that connected people would influence each other based on the social influence theory. These methods incorporate friends' opinion on candidate items to model the user's preference. For example, TrustSVD [10] incorporates the friends' decisions to model the users' preference and SAMN [2] design a two-level attention mechanism to model the friends' influence. However, these methods only consider the first-order local neighbors' information and neglect the helpful information from distant neighbors.

Recently, deep neural networks for graph data, which are known as Graph Neural Networks (GNNs), have shown an effective performance and have experienced rapid development. The core idea
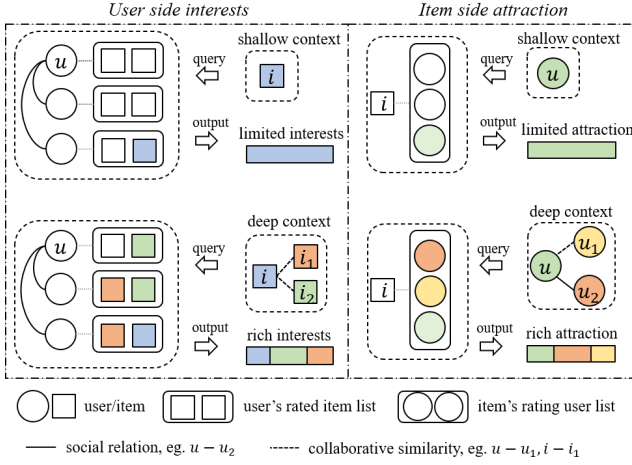
Figure 1: Illustration of dual side deep context. The left part is user side interest while the right part is item side attraction. Deep context have been shown to be helpful to model users' rich interests and items' rich attraction.

of GNNs is about the propagation and aggregation of information from neighbor nodes [1, 4, 18], which naturally accord with social recommendations. Several recent works that utilize GNNs for social recommendations [7, 40, 41, 43] have been proposed. For example, the DiffNet++ method [40] developed a GNN based model to simulate both the social influence and user interest diffusion process. The DANSER model [43] developed a dual-graph attention network to model the two-fold social effects collaboratively.

Although the aforementioned social recommendation methods have shown performance improvement, they do not fully take advantage of social network information in a deep way. First, when modeling user side interest, most methods consider the friends' information equally without considering the specific recommendation context. This results in a lot of noisy information are aggregated from friends' information. And some other methods consider the candidate item as a context to model the context-aware friends and user's interest (as shown in the top-left part of Figure 1). However, considering the candidate item as a (shallow) context only leads to interest information biased to some extend, and thus limited user and friends' interest can be obtained. In fact, when modeling user side interests from user's and friends' historical interacted items, not only the information related to the candidate item can help, the information related to candidate's similar items can also reflect the user's interests (as shown in the bottom-left part of Figure 1), especially when the candidate item has only limited interaction history.

Furthermore, the related information from the candidate item's interaction history to the targeted user also reflects the item's attraction to the users. Few social recommendation work [43] models the item attraction based on the targeted user. As shown in the top-right part of Figure 1, considering the targeted user only is also a shallow context-aware method and results in poor item attraction being modeled. In fact, if the item's past consumers is a friend of the targeted user or have the similar consumption habit to the target

user, the item may be more attractive to the users, as shown in the bottom-right part of Figure 1. Hence, it is obvious that considering the social relation and similarity relations among users helps to extract more useful information from the candidate item's interaction history.

While it is of great potential to leverage social relations and similarity relations in mining interaction history information for the recommendation, there are still several significant challenges. First, the high-order social relations and similarity relations are complex and it is not easy to extract the most related information for modeling user preference and item attribute. Second, it is not trivial to extract the user interest and item attraction from the interaction history based on the high-order relation enhanced context, as there may be much noisy information.

To tackle the challenges mentioned above, in this paper, we proposed DICER (**D**ual s**I**de deep **C**ontext-awar**E** modulation for social **R**ecommendation) which incorporating the high-order neighbor information to model the enhanced user preference and item attribute, and extracting the most related information from the interaction history based on the graph enhanced deep context.

Our contribution in this paper can be summarized as follows:

- We propose a novel neural model for the social recommendation, which utilizes relation-aware Graph Neural Network to use multi-relation and high-order neighbor information effectively. A Deep Context-aware Modulation is introduced to model user side interests and item side attraction.
- To the best of our knowledge, we are the first to consider the social relation and collaborative similarity among users as deep context to model the item attraction representations. In duality, we also consider the collaborative similarity among items as deep context to model the user and friends' interest representations.
- Experiments on two real-world benchmark datasets are conducted to demonstrate the effectiveness of our model. We show that our model consistently outperforms the state-of-the-art models.

The remainder of this paper is organized as follows. We introduce the proposed model in Section 3. In Section 4, we conduct experiments on two real-work datasets to demonstrate the effectiveness of the proposed model. In Section 5, we review related works. Finally, we conclude with future directions in Section 6.

2 PRELIMINARY

We first introduce the and notations used throughout the paper and then describe the problem formulation.

2.1 Notations

We define $U(|U| = M)$ and $I(|I| = N)$ as the sets of users and items respectively. We use u,v to index users, and i, j to index items. We consider a user-item interaction matrix $R = [r_{u,i}]_{M \times N} \in \{0, 1\}$ indicating whether u has interacted (e.g., purchased, clicked) on item i. We use $R_I(u)$ and $R_U(i)$ to respectively denote the set of items rated by user u and the set of users who have rated item i.

We consider the user-user social network as a graph $G_U^S = (V_U, E_U^S)$ where V_U is the set of users and E_U^S is the set of edges that connect two users. We use $F_U(u)$ to denote the set of nodes

Table 1: Summary of notations

Symbols	Definitions and Descriptions
U	set of users
I	set of items
R	user-item interactions
G_U^S	user-user social network
G_U^R	user-user collaborative similarity network
G_I^R	item-item collaborative similarity network
$R_I(u)$	the set of items rated by user u
$R_U(i)$	the set of users who have rated item i
$F_U(u)$	the set of nodes adjacent to u in G_U^S
$N_U(u)$	the set of nodes adjacent to u in G_U^R
$N_I(i)$	the set of nodes adjacent to i in G_i^R
p_u	the embedding of user u
q_i	the embedding of item i
$h_u^{S,l}$	the l-th layer representation of user u in G_U^S
$h_u^{R,l}$	the l-th layer representation of user u in G_U^R
z_i^l	the l-th layer representation of item i in G_I^R
$h_u^\star$	the graph enhance user preference of user u
$z_i^\star$	the graph enhance user preference of item i
m_u^i	the user interests of user u in candidate item i
x_u^i	the integrated user interests of user u in candidate item i
$\alpha_{u,v}$	the friends attention of friend v in contributing to x_u^i
y_i^u	the item attraction of item i to targeted user u
D	the number of embedding dimension
W	the weight matrix in neural network
$\odot$	the element-wise product operation

adjacent to u in G_U^S. We define $P = \{p_u\}_{D \times M}$, where D is the embedding dimension and p_u denotes the embedding vector for user u; similarly, $Q = \{q_i\}_{D \times N}$, where q_i denotes the embedding vector for item i. The mathematical notations used in this paper are summarized in Table 1.

2.2 Problem Formulation

The social recommendation problem is defined as [35, 43]: given the user-item interactions R and the user-user social network G_U^S, the goal is to predict the unobserved interactions in R, i.e., the probability of a targeted user u clicking or purchasing an unobserved candidate item i.

2.3 Background

Graph Neural Network. Graph Neural Networks have been proposed to learn the graph topological structure and nodes' feature information. And the most representative method is Graph Convolutional Networks (GCN) [18], which learns representation for a node by aggregating its neighbors' representation iteratively. GCN treats the neighbor information equally without considering the different importance of the neighbors. In contrast, Graph Attention Networks (GAT) [37] specifying different weights to different neighbors, which helps the model learn more related information from neighbors. Moreover, Neural Graph Collaborative Filtering (NGCF) [39] proposed an effective message propagation approach to aggregate the more similar information from neighbors, which enable the model to model the high-order relation information and filter the noisy neighbor effectively.

3 THE PROPOSED MODEL

In this section, we give the overall architecture of the proposed model DICER and then introduce modules in detail. Finally, we discuss the training of the model.

3.1 Architecture Overview

The architecture of the proposed model is shown in Figure 2. The model consists of four modules. The first module is collaborative graph construction, which constructs two collaborative similarity graphs for users and items from user-item interactions, respectively. The second module is the high-order relation exploitation module, which is to learn the users' graph enhanced preference and items' graph enhanced attribute based on the social graph and the two collaborative similarity graphs. The third module is the dual side deep context-aware modulation, which is to model the user interest from user's and friends' interaction history based on the item's graph enhanced attribute and the item attraction from item's interaction history based on the user's graph enhanced preference. The fourth module is the user and item matching module based on the user's interest and preference and the item's attraction and attribute. We describe these four modules in detail next.

3.2 Collaborative Graph Constructing

Collaborative Relation Definition. We define *collaborative similar users* as users who have similar consumption habits and *collaborative similar items* as items with similar clicks or rated history. And one way to calculate the collaborative similarity is by the common interaction history of the users (items) [31] [43]. For any user u and user v, we define the strength of collaborative similarity $sim_{u,v}$ between user u and user v as below.

$$sim_{u,v} = \frac{|R_I(u) \cap R_I(v)|}{\sqrt{|R_I(u)| \cdot |R_I(v)|}} \quad (1)$$

Then user u is collaborative similar to user v if and only if $sim_{u,v} > \eta$ with η a fixed threshold, and the items' collaborative similarity relation is the same.

We construct the collaborative similarity graph based on the collaborative similarity relation as $G_U^R = (V_U, E_U^R)$ and $G_I^R = (V_I, E_I^R)$ for users and items respectively, where V_U is the set of users and E_U^R is the set of edges that connect two collaborative similar users, and V_I is the set of items and E_I^R is the set of edges that connects two collaborative similar items.

3.3 High-order Relation Exploitation

Relation-aware graph neural network module. In order to exploit the high-order relation in social network G_U^S and collaborative similarity relation in G_U^R and G_I^R, we propose a relation-aware Graph Neural Network (RGNN), an extension of NGCF which can efficient aggregate and select the most related information from neighbors. Let $z_i^0 = q_i$ and $h_u^{S,0} = h_u^{R,0} = p_u$ as the first layer inputs for the RGNN module, by feeding the first layer inputs into three RGNN layers respectively, these layers can recursively model the propagation of item attribute or user preference in different graph. The detail process is as below:

I. RGNN to aggregate item-item collaborative similar neighbors. For each item i, given its l-th layer input z_i^l, we model the updated
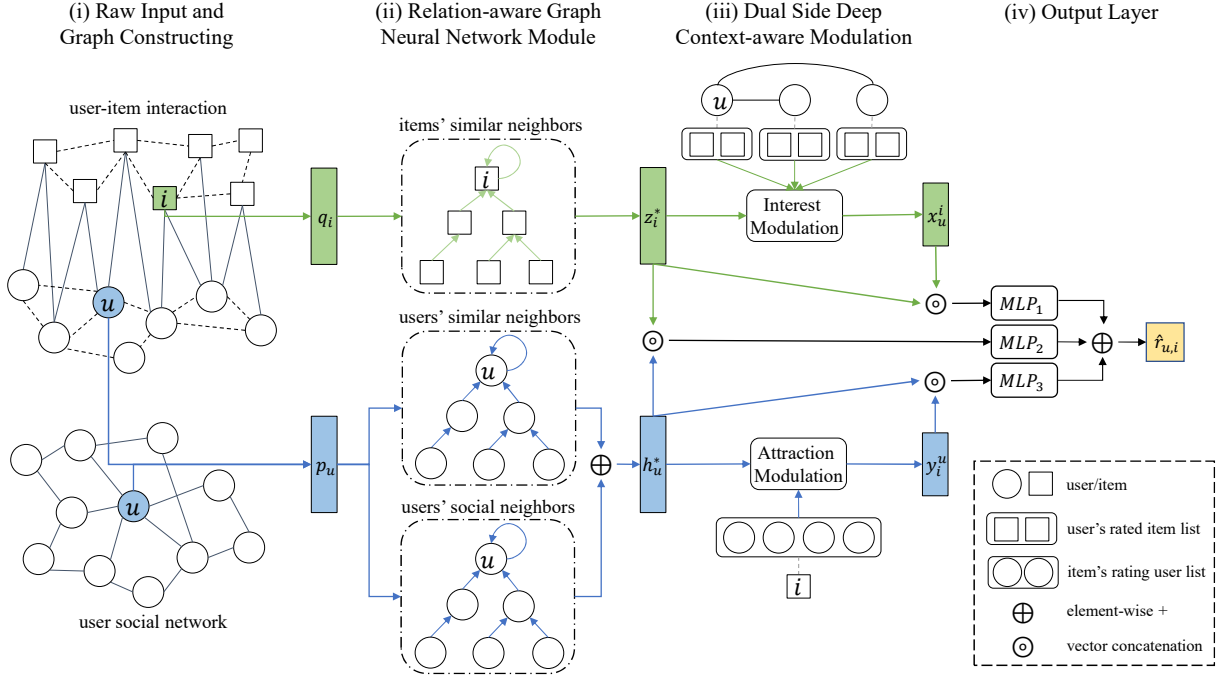
Figure 2: Architecture of the proposed DICER model with four modules. i) The Raw input which includes the user-item interaction network and user social network, we construct two collaborative similarity network based on the interaction network. ii) In the relation-aware GNN module, RGNN are to model the graph enhanced user preference $h_u^\star$ and item attribute $z_i^\star$. iii) In the dual side deep context-aware modulation layer, these graph enhanced representation is used as a deep context to capture user interest x_u^i and item attraction y_i^u from interaction history respectively. iv) Finally, user's preference and interest and item's attribute and attraction are input into the output layer to predict the final score $\hat{r}_{u,i}$.

user representation z_i^{l+1} at the $(l+1)$-th layer from G_I^R as:

$$
\begin{aligned}
z_i^{l+1} &= AGG_I^R \left(z_i^l, \; z_j^l, \forall j \in N_I(i) \right) \\
&= \sigma \left(\sum_{j \in N_I(i)} \left(W_1^I z_i^l + W_2^I (z_i^l \odot z_j^l) \right) \right)
\end{aligned}
\tag{2}
$$

where σ is the LeakyReLU activation function, $\odot$ denotes the element-wise product, W_1^I, W_2^I are the weight matrix, $N_I(i)$ is the collaborative similar neighbors of item i from G_I^R and z_j^l is the l-th layer representation of item j. After the propagation process, the updated representation z_i^{l+1} aggregate the most related information from the collaborative similar neighbors of l-th layer.

II. RGNN to aggregate social network neighbors. For each user u, given its l-th layer input $h_u^{S,l}$, we model the updated user representation $h_u^{S,l+1}$ at the $(l+1)$-th layer from G_U^S as:

$$
\begin{aligned}
h_u^{S,l+1} &= AGG_U^S \left(h_u^{S,l}, \; h_f^{S,l}, \forall f \in F_U(u) \right) \\
&= \sigma \left(\sum_{f \in F_U(u)} \left(W_1^S h_f^{S,l} + W_2^S (h_u^{S,l} \odot h_f^{S,l}) \right) \right)
\end{aligned}
\tag{3}
$$

where W_1^S, W_2^S are the weight matrix, $F_U(u)$ is the social neighbors of user u from G_U^S and $h_f^{S,l}$ is the l-th layer representation of user f. After the propagation process, the updated representation $h_u^{S,l+1}$

aggregate the most related information from the social network neighbors of l-th layer.

III. RGNN to aggregate user-user collaborative similar neighbors. Similarly, for each user u, given its l-th layer input $h_u^{R,l}$, we model the updated user representation $h_u^{R,l+1}$ at the $(l+1)$-th layer from G_U^R as:

$$
\begin{aligned}
h_u^{R,l+1} &= AGG_U^R \left(h_u^{R,l}, \; h_v^{R,l}, \forall v \in N_U(u) \right) \\
&= \sigma \left(\sum_{v \in N_U(u)} \left(W_1^U h_f^{R,l} + W_2^U (h_u^{R,l} \odot h_v^{R,l}) \right) \right)
\end{aligned}
\tag{4}
$$

where W_1^U, W_2^U are the weight matrix, $N_U(u)$ is the collaborative similar neighbors of user u from G_U^R and $h_v^{R,l}$ is the l-th layer representation of user v. After the propagation process, the updated representation $h_u^{R,l+1}$ aggregate the most related information from the collaborative similar neighbors of l-th layer.

As users play a central role in both the social network G_U^S and collaborative similarity network G_U^R, her final embedding h_u^l is a fusion of the aggregated neighbors information $h_u^{S,l+1}$ and $h_u^{R,l+1}$ from the G_U^S and G^{R_U} respectively. In fact, we try different kinds of fusion functions, including the concatenation and the addition, and find the addition always shows the best performance. Therefor,

we use the addition as the fusion function in Eq.(5)

$$h_u^l = h_u^{S,l} + h_u^{R,l} \qquad (5)$$

Finally, After the iterative propagation process with l times in each RGNN layers, we obtain the updated representation set of u and i with h_u^l and z_i^l for $l = [0, 1, 2, ..., l]$. Then for each user u, her final graph enhanced preference is denoted as: $h_u^\star = [h_u^0 \| h_u^1 \| ... \| h_u^l]$ that concatenates her representation at each layer. Similarly, each item's final graph enhanced attribute is: $z_i^\star = [z_i^0 \| z_i^1 \| ... \| z_i^l]$.

3.4 Dual Side Deep Context-aware Modulation

Since only considering the candidate item or targeted user is unable to model the rich user interests or item attraction, in this section, we introduce our dual side deep context-aware modulation module which considers the graph enhanced user preference and item attribute as deep context to model the user interests and item attraction respectively. We will go into the details of each side modulation.

*3.4.1 **Deep context-aware user interest modulation**.* In social recommendation models, considering the information related to the candidate item from the user and friends' interaction history is necessary to model a better user interest [8, 43]. However, consider the related information only based on the candidate item is not enough and may lead to missing information and a narrow understanding of user interest. In fact, users' interests in candidate's similar items are also useful.

In order to fully mine the information which can reflect the users' interest in the candidate item from their interaction history, we consider the graph enhanced item attribute as deep context which incorporates the high-order collaborative similar neighbors information from G_I^R, then select the most related information from the user's rated items based on the deep context by a modulation operation as below:

$$m_u^i = f_I\left(z_i^\star, \ z_j^\star, \forall j \in R_I(u)\right) \qquad (6)$$

where $R_I(u)$ is the rated items of user u, and the $f_I(\cdot)$ is:

$$m_u^i = MP_{j \in R_I(u)}(\{z_{j,d}^\star \odot z_{i,d}^\star\}), \forall d = 1, ..., D \qquad (7)$$

where $z_{j,d}^\star, z_{i,d}^\star$ are the d-th feature of $z_j^\star, z_i^\star$ respectively. MP is the max pooling operation which can help to focus on the related information from the user's rated history based on the graph enhanced item attribute. Thus the obtained representation m_u^i capture the user u's rich interest in candidate item i .

After obtained the user and her friend's interests in the candidate item, we aggregate the friend's interest with no-uniform weight. And the weight is varied when it comes to different candidate items. Intuitively, if a friend's interest in the candidate item is more similar to the targeted user, he should significantly influence the user's decision on the candidate item. Formally, given the user interest m_u^i and friend interest m_f^i, we compute the attention weight $\alpha_{u,f}$:

$$\alpha_{u,f}^* = (m_u^i)^\top \cdot (m_f^i) \qquad (8)$$

Then, the final attention weights of the friend interests are obtained by normalizing the above attentive scores using softmax function,

which can be interpreted as the influence of the friend to the final user interest representation of user u as:

$$\alpha_{u,f} = \frac{exp(\alpha_{u,f}^*)}{\sum_{f \in F_U(u)} exp(\alpha_{u,f}^*)} \qquad (9)$$

Finally, the final user interest (which integrated friends' interest) of user u is through the sum:

$$x_u^i = m_u^i + \sum_{f \in F_U(u)} \alpha_{u,f} \, m_f^i \qquad (10)$$

which considers both the users' own interest and the influence of her friends. Note that we tried a lot of strategies of combining the different features, such as concatenation, addition or full-connected neural layer, and find the addition always shows the best performance.

*3.4.2 **Deep context-aware item attraction modulation**.* Similar to the user interest, the related information to the targeted user from item's interaction history can reflect the attraction of the candidate item [27, 43]. However, consider the related information only based on the targeted user is also not enough and may result in poor item attraction. In fact, if the candidate items' past consumers are the friends of the targeted user or have similar consumption habits to the targeted user, the item may have much attraction to the targeted user.

In order to fully mine the information which can reflect the items' attraction to the targeted user from their interaction history, we consider the graph enhanced user preference as deep context which incorporates the high-order collaborative similar neighbors information from G_U^R and soical neighbors information from G_U^S, then select the most related information from the item's past consumers based on the deep context by a modulation operation as below:

$$y_i^u = f_U\left(h_u^\star, \ h_v^\star, \forall v \in R_U(i)\right) \qquad (11)$$

where $R_U(i)$ is the user who rated item u, and the $f_U(\cdot)$ is:

$$y_i^u = MP_{v \in R_U(i)}(\{h_{v,d}^\star \odot h_{u,d}^\star\}), \forall d = 1, ..., D \qquad (12)$$

where $h_{v,d}^\star, h_{u,d}^\star$ are the d-th feature of $h_v^\star, h_u^\star$ respectively. Thus the obtained representation y_i^u capture the item i's rich attraction to targeted user u based on the user-user social relation and collaborative similarity relation.

3.5 Output Layer

Since user's decision on candidate item depend on both user preference and item attribute, we firstly predict the score based on the graph enhanced representation of user and item:

$$r_{ui}^O = MLP_1(h_u^\star, z_i^\star) \qquad (13)$$

Next, the user's interest in the item and the item's attraction to the user reflect the matching score between them from two perspectives. Then we predict another two matching scores based on the two perspectives to make the model more robust.

$$r_{ui}^U = MLP_2(x_u^i, z_f^\star) \qquad (14)$$

$$r_{ui}^I = MLP_3(y_i^u, h_u^\star) \qquad (15)$$

Table 2: Statistics of the datasets

Dataset	Ciao	Epinion
Num. of Users	7,375	20,608
Num. of Items	106,797	23,585
Num. of Ratings	282,650	454,002
Num. of Relations	111,781	351,486
Rating Density	0.0359%	0.0934%
Relation Density	0.2055%	0.0828%

The final predicted probability that user u will interact with item i is calculated by weighted-sum of the 3 scores,

$$\hat{r}_{u,i} = \lambda_1 r_{ui}^O + \lambda_2 r_{ui}^U + \lambda_3 r_{ui}^I \qquad (16)$$

where λ_1, λ_2 and λ_3 are hyperparameters, and $\lambda_1 + \lambda_2 + \lambda_3 = 1, \lambda_1 >= 0, \lambda_2 >= 0, \lambda_3 >= 0$.

3.6 Model Training

To learn model parameters of DICER, we need to specify an objective function to optimize. For implicit feed-back, the most widely adopted loss function is the cross-entropy defined as:

$$\mathcal{L} = -\sum_{(u,i)} r_{u,i} \log \hat{r}_{u,i} + (1 - r_{u,i}) \log(1 - \hat{r}_{u,i}).$$

To optimize the objective function, we adopt mini-batch Adaptive Moment Estimation (Adam) [17] as the optimizer in our implementation. Its main advantage is that the learning rate can be self-adapted during the training phase which eases the pain of choosing a proper learning rate. We also adopt the dropout strategy [32] to alleviate the overfitting issue in optimizing deep neural network models.

4 EXPERIMENT

To comprehensively evaluate the proposed model DICER [1], we conduct experiments to answer the following research questions:
RQ1 How does DICER perform compare with state-of-the-art methods for recommendation and social recommendation?
RQ2 Are the key components in DICER, including relation-aware GNN module and deep context-aware modulation, necessary for improving performance?
RQ3 How do hyper-parameters in DICER impact recommendation performance?

4.1 Experiment Setup

4.1.1 ***Datasets***. We conduct experiments on two representative datasets: *Ciao* and *Epinion*, which are taken from popular social network website Ciao [2] and Epinion [3]. Each social networking service allows users to clicked items and add friends. Hence, they provide a large amount of rating information and social information.

---

[1]The codes are released at https://github.com/Drone-Banks/DICER-WWW-2021.
[2]http://www.ciao.co.uk
[3]http://www.Epinion.com

As long as some user-user or user-item interactions exist, the corresponding rating is assigned a value of 1 as implicit feedback. The statistical details of these datasets are summarized in Table 2

4.1.2 ***Baselines***. To illustrate the effectiveness of our model, we compare DICER with two classical collaborative filtering (CF) models, two social-based recommendation models, two deep learning based recommendation models, and two deep-learning based social recommendation models.

The first group of models are CF models:

- **BPR** [30]: This is a competing latent factor model for implicit feedback based recommendation. It designed a ranking based function that assumes users prefer items they like compared to unobserved ones.
- **FM** [29]: This model is a unified latent factor based model that leverages the user and item attributes. In practice, we use the user and item features as introduced above.

The second group contains social based recommendation models that utilize social relation information:

- **TrustMF** [45]: This is a matrix factorization method, which maps users into two spaces and opti mizes user embedding to retrieve the trust matrix.
- **TrustSVD** [10]:This is another matrix factorization-based method, wich incorporates friends' embedding vectors into targeted user's predicted rating.

The third group is deep learning based recommendation models:

- **NCF** [12]: This is a deep learning based recommendation model which leverage a multi-layer perceptron to learn the user-item interaction function.
- **NGCF** [39]: This is a graph based recommendation model that model the high-order connectivity in the user-item graph and inject the collaborative signal into the embedding process in an explicit manner.

The fourth group is deep-learning based social recommendation models:

- **SAMN** [2]: This is a strong baseline for social recommendation, which leverages attention mechanisms to model both aspect- and friend-level differences for social-aware recommendation.
- **DiffNet++** [40]: This is another strong baseline, which adopts GNN and considers both the social influence diffusion and the latent collaborative interests diffusion in social-aware recommendation.

The comparison of DICER and the baseline methods are listed in Table 3.

4.1.3 ***Evaluation Metrics***. We adopt *Recall@K* and *NDCG@K* to evaluate the performance of all methods. The two metrics have been widely used in previous recommendation studies [2, 44, 47]. *Recall@K* considers whether the ground truth is ranked among the top K items, while *NDCG@K* is a position-aware ranking metric.

4.1.4 ***Experiments Details***. The proposed DICER was implemented with PyTorch and we use the Xavier initializer [9] to initialize the model parameters. For each dataset, we use 80% as a training set to learn parameters, 10% as a validation set to tune hyper-parameters and 10% as a test set for the final performance

Models	Social Domain		Item Domain		User Interest		Item Attraction		DL
	S	HS	I	HI	SC	DC	SC	DC	
TrustMF	√	\	√	\	\	\	\	\	\
TrustSVD	√		√	\	√	\	\	\	\
NCF	\		√	\	\	\	\	\	√
NGCF	\		√	√	\	\	\	\	√
SAMN	√	\	√	\	√	\	\	\	√
DiffNet++	√	√	√	√	\	\	\	\	√
DICER	√	√	√	√	√	√	√	√	√

comparison. The hyper-parameter settings are as follow: learning rate is 0.001, training batch size is 4096, embedding size $D = 64$, the RGNN layers $l = 3$, coefficient $\lambda_1 = \lambda_2 = \lambda_3 = \frac{1}{3}$, collaborative similarity threshold $\eta = 0.1$, the LeakyReLU slope is 0.2. In the training process, as there are many more unobserved items for each user, we randomly select 8 times pseudo negative samples for each user at each iteration. Since each iteration we change the pseudo negative samples, each unobserved item gives a weak signal. For all the baselines, we carefully tune the parameters to ensure the best performance.

4.2 Comparative Results: RQ1

The comparison of different methods on two datasets is shown in Table 4. We set the length K = 5, 10, and 15 in our experiments to evaluate on different recommendation lengths. From the results, the following observations can be made:

First, methods incorporating social information generally perform better than non-social methods. For example, in Table 4, the performance of TrustSVD and TrustMF is better than BPR and FM, and SAMN, DiffNet++ and DICER outperform NGCF and NCF. This is consistent with previous work [2, 44, 48], which indicates that social information is helpful to improve the recommendation performance.

Second, our method DICER achieves the best performance on the two datasets. Specifically, compared to SAMN – an attention-based deep learning model, DICER exhibits average improvements of 10.40% and 6.89% on the two datasets. And compared to DiffNet++ - a recently proposed and very expressive GNN-based model, DICER exhibits average improvements of 9.59% and 3.21% on the two datasets. The substantial improvement of our model over the baselines could be attributed to two reasons: (1) our model use relation-aware GNN to deal with the high-order social relation and collaborative similarity relation, which allow the related information from multi-relation neighbors to be utilized; (2) we model the users' interests and items' attraction based on the deep context, i.e., the graph enhanced user and item representation.

4.3 Study of DICER: RQ2

In this subsection, we study the impact of model components.

4.3.1 Effect of Dual Side information. In the last subsection, we have demonstrated the effectiveness of the proposed model. The models provide dual side modulation to (1) capture the more diverse user interest and (2) model the rich item attraction. To prove the effective of the dual side information, we compare DICER with its two variants: DICER w/o ui, and DICER w/o ia. These two variants are defined in the following:

- DICER w/o ui: The user side modulation and the user interest information is removed. This variant only use the users' graph enhanced preference $h_u^\star$ and items' graph enhanced attribute $z_i^\star$ and items' attraction y_i^u to predict the score, while ignoring the user interest x_u^i.
- DICER w/o ia: The item side modulation and the item attraction information is removed. This variant only use the items' graph enhanced attribute $z_i^\star$ and users' graph enhanced preference $h_u^\star$ and users' interest x_u^i to predict the score, while ignoring the item attraction y_i^u.

The performance of DICER and its variants on Ciao and Epinion are given in Figure 3. From the results, we have the following findings:

- **User Interest**: We now focus on analyzing the effectiveness of user interest, which combine the related information extracted from the user and friends' interaction history. DICER w/o ui performs worse than DICER, On average, the relative reduction is 3.53% on NDCG metric and 4.54% on Recall metric. It verifies that the user interest information is important to predict the score and boost the recommendation performance.
- **Item Attraction**: We can see that without the item attraction, the performance decrease significantly. On average, the relative reduction is 5.14% on NDCG metric and 5.79% on Recall metric, respectively. It justifies our assumption that item attraction has informative information that can help to predict the final score and improve the performance of recommendation.

4.3.2 Effect of deep context and modulation. To get a better understanding of the proposed DICER model, we further do some ablation studies for the key components of DICER - the deep context-aware modulation and the results are shown in Table 5. There are three different relations for deep context, including item's collaborative similarity α, user's collaborative similarity β, user's social relation μ. We first compare DICER with its four variants: DICER-α, DICER-β, DICER-μ and DICER-$\alpha\&\beta\&\mu$. Moreover, we use the attention mechanism to replace the modulation as the variant DICER-*attn* to prove the effectiveness of modulation. These five variants are defined in the following:

- DICER-α: The item's collaborative similarity relation is eliminated during modeling the user interest. This variant only considers the candidate item as the context.
- DICER-β: The user's collaborative similarity relation is eliminated during modeling the item attraction. This variant only

Table 4: Comparisons of different methods on Two datasets. Best baselines are underlined. The proposed method achieves best performances on all metrics which are in boldface. The last column "RI" indicates the relative improvement of DICER over the corresponding baseline on average.

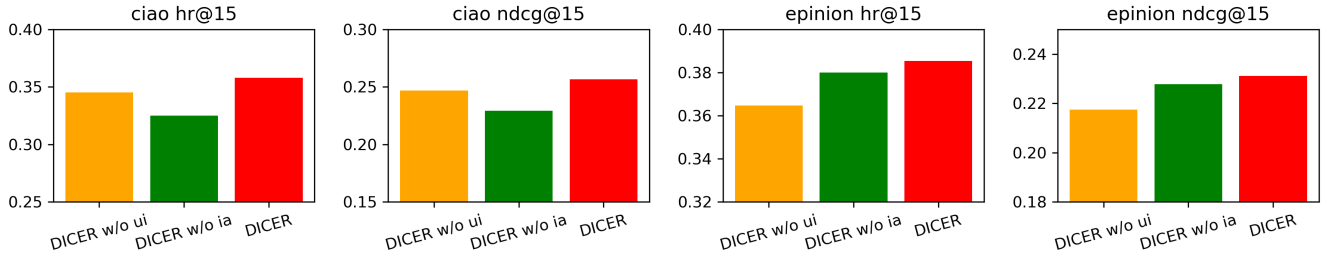| Ciao	Recall@5	Recall@10	Recall@15	NDCG@5	NDCG@10	NDCG@15	RI
BPR	0.1782	0.2143	0.2469	0.1618	0.1720	0.1814	+42.84%
FM	0.1852	0.2269	0.2613	0.1638	0.1760	0.1861	+37.83%
TrustMF	0.2151	0.2631	0.3027	0.1916	0.2062	0.2179	+18.28%
TrustSVD	0.2159	0.2698	0.3117	0.1884	0.2056	0.2179	+17.53%
NCF	0.1840	0.2268	0.2609	0.1644	0.1773	0.1873	+37.62%
NGCF	0.2330	0.2821	0.3185	0.2063	0.2212	0.2319	+10.53%
SAMN	0.2322	0.2836	0.3245	0.2030	0.2205	0.2332	+10.40%
DiffNet++	0.2330	0.2844	0.3259	0.2063	0.2226	0.2351	+9.59%
DICER	**0.2554**	**0.3151**	**0.3579**	**0.2243**	**0.2437**	**0.2565**	-
Epinion	Recall@5	Recall@10	Recall@15	NDCG@5	NDCG@10	NDCG@15	RI
BPR	0.1616	0.2264	0.2716	0.1253	0.1484	0.1622	+44.06%
FM	0.1592	0.2273	0.2763	0.1233	0.1476	0.1627	+44.38%
TrustMF	0.1816	0.2602	0.3163	0.1374	0.1651	0.1821	+27.75%
TrustSVD	0.1927	0.2623	0.3090	0.1466	0.1712	0.1852	+24.31%
NCF	0.1834	0.2624	0.3187	0.1397	0.1675	0.1844	+26.28%
NGCF	0.2099	0.2918	0.3488	0.1618	0.1908	0.2080	+11.80%
SAMN	0.2206	0.3055	0.3625	0.1697	0.1996	0.2170	+6.89%
DiffNet++	0.2298	0.3183	0.3786	0.1742	0.2055	0.2236	+3.21%
DICER	**0.2370**	**0.3269**	**0.3854**	**0.1818**	**0.2134**	**0.2312**	-

Figure 3: Effect of dual side information on Ciao and Epinion datasets

Table 5: Effect of deep context and modulation on Ciao

Models	recall@5	recall@10	recall@15	ndcg@5	ndcg@10	ndcg@15
DICER-α	0.2307	0.2842	0.3302	0.2030	0.2193	0.2327
DICER-β	0.2340	0.2928	0.3380	0.2066	0.2251	0.2388
DICER-μ	0.2401	0.2945	0.3357	0.2108	0.2289	0.2401
DICER-α&β&μ	0.2187	0.2733	0.3169	0.1918	0.2093	0.2222
DICER-*attn*	0.2150	0.2688	0.3097	0.1875	0.2054	0.2179
DICER	**0.2554**	**0.3151**	**0.3579**	**0.2243**	**0.2437**	**0.2565**

considers the targeted user and her social neighbors as the context.
- DICER-μ: The user's social relation is eliminated during modeling the item attraction. This variant only considers the targeted user and her collaborative similar neighbors as the context.

- DICER-α&β&μ: This variant eliminates the three relations (item's collaborative similarity α, user's collaborative similarity β, user's social relation μ) and onyl consider the candidate item or targted user as context.
- DICER-*attn*: This variant replaces the modulation function with a attention mechanism like Eq.(8) and Eq.(9) to select the related information from interaction history.

The results of different relation based context on DICER and the DICER-*attn* are shown in Table 5. From the results, we have the following findings:

- The three relations all contribute to the deep context-aware modulation for modeling the user interest and item attraction. And the more relation the deep context is based on, the context-aware information can be better for improving the performance.
- The modulation function is more effective than the attention mechanism when model the related information from

Table 6: Effect of GNN module on Ciao

Models	recall@5	recall@10	recall@15	ndcg@5	ndcg@10	ndcg@15
DICER-embed	0.1823	0.2190	0.2535	0.1610	0.1718	0.1818
DICER-GAT	0.2498	0.3077	0.3548	0.2182	0.2369	0.2511
DICER-RGNN	**0.2554**	**0.3151**	**0.3579**	**0.2243**	**0.2437**	**0.2565**

Table 7: Effect of rgnn layer numbers on Ciao

Models	recall@5	recall@10	recall@15	ndcg@5	ndcg@10	ndcg@15
DICER-1	0.2526	0.3074	0.3534	0.2211	0.2385	0.2521
DICER-2	0.2530	0.3088	0.3530	0.2226	0.2404	0.2537
DICER-3	**0.2554**	**0.3151**	**0.3579**	**0.2243**	**0.2437**	**0.2565**

the interaction history, This may due to the max-pooling operation in modulation is aiming to select the most similar feature which is more effect than the attention mechanism.

4.3.3 Effect of GNN module. To prove the effectiveness of the RGNN module, We denote our model as DICER-RGNN and simply our model as DICER-embed and DICER-GAT, while DICER-embed removes the GNN module and DICER-GAT replaces the GNN module with GAT. And we compare the three variants with DICER. As we can see in Table 6, GAT improve performance by specifying different weight to aggregate the neighbors' information. Moreover, we can find that DICER with the RGNN module further improves performance by aggregate more related information via the max-pooling operation in modulation (about extra 2.6% impv. for NDCG over DICER-GAT). The result show that the most related information from the multi-relation neighbors can bring significant performance improvement.

To sum up, DICER can leverage the high-order multi-relation information and model the rich user interests and item attraction via the deep context-aware modulation, which can boost the recommendation performance.

4.4 Parameter Sensitivity: RQ3

In this subsection, we first analyze the effect of embedding size of user embedding P, item embedding Q, on the performance of our model. It should be noted that, if the embedding size is D, the graph enhanced embedding vector's size will be $(l+1) \times D$ and it is $4D$ in our implementation. Figure 4 presents the performance comparison w.r.t. the length of embedding of our proposed model on Ciao and Epinion datasets. In general, with the increase of the embedding size, the performance also increases. As we can see, increasing embedding size from 8 to 32 can bring significant performance improvement, a further increase to 64 brings less improvement but a larger computational complexity. Therefore, we need to choose an appropriate embedding size to balance model performance and efficiency.

Secondly, to investigate whether DICER can benefit from multiple RGNN propagation layers, we search the layer numbers in the range of {1,2,3}. Table 7 summarizes the experimental results, where DICER-n indicates the model with n RGNN propagation layers. As we can see, increasing the number of propagation layers substantially enhances the recommendation performance. We attribute the improvement to the effective modeling of neighbors

information: the related information from neighbors can enhance the user preference and item attribute, which contribute to the deep context-aware modulation, respectively.

5 RELATED WORK

In this section, we briefly review the related work about the social recommendation, graph neural network techniques employed for recommendation, and the context-aware recommendation.

Social Recommendation. In recent years, there are lots of works exploiting user's social relations for improving the recommender system [33, 34, 42, 45]. Most of them assume that users' preference is similar to or influenced by their friends, which can be suggested by social theories such as social homophily [26] and social influence [25]. According to the assumptions above, social regularization has been proposed to restrain the user embedding learning process in the latent factor based models [13, 15, 23, 24]. And TrustMF [45] model is proposed to model the mutual influence between users by mapping users into two low-dimensional space: truster space and trustee space and factorize the social trust matrix. By treating the social neighbors' opinion as the auxiliary implicit feedbacks of the targeted user, TrustSVD [10] is proposed to incorporate the social influence from social neighbors on top of SVD++ [19]. Moreover, some recent studies like [2, 6, 38] and [3, 8, 21] leverage deep neural network and transfer learning or adversarial learning approach respectively, to learn a more complex representation or model the shared knowledge between social domain and item domain. However, comparing with our models in this paper, the common limitations of existing studies are: i) they did not leverage the high-order social relation and collaborative relation among users; ii) they ignore the related information from interaction history based on the relation enhanced deep context.

Graph Neural Network for Recommendation. More recently, Graph Neural Networks (GNNs) have been proven to have great potential to learn the graph structure data [4, 5, 18]. In the task of recommender systems, the user-item interaction records obviously form a typical graph. Hence, there are many works that adopted GNNs to solve the recommendation problem [36, 39, 46]. GCMC [36] proposed a graph auto-encoder framework to predict unobserved interactions in the user-item matrix. Pinsage [46] proposed a random-walk graph neural network to learn the node embeddings in web-scale graphs. And NGCF [39] proposed a multi-layer GNNs which can model the higher-order collaborative signals between users and items during the users and items embedding learning process. As the social relation among users could be naturally formulated as a user-user graph, there are also some works [7, 40, 41, 43] using GNNs to capture social information for recommendation. The Diffnet++ [40] developed a GNN based model to simulate both the social influence and user interest diffusion process. And DANSER [43] developed a dual graph attention networks to model the two-fold social effects collaboratively. Although these works model the social influence with GNNs, they didn't take advantage of high-order relations. Our work differs from these works. We model the high-order relation with multi-relation GNNs and consider the relation enhanced information as a deep context to extract the related information from interaction history.
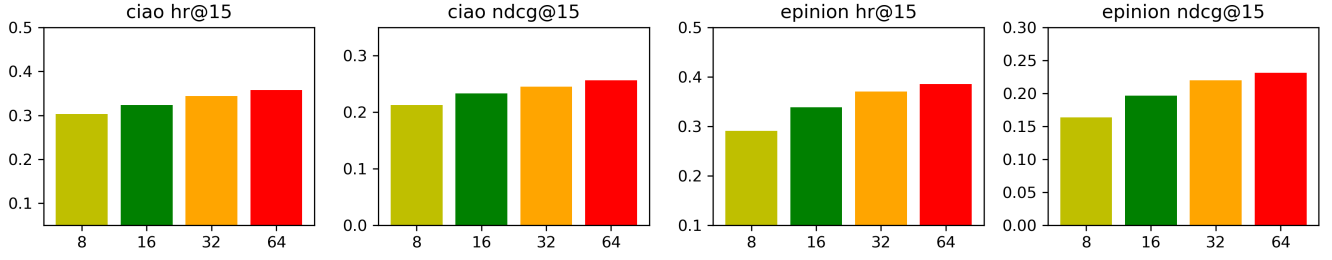
Figure 4: Effect of embedding size D on Ciao and Epinion datasets. Noted that, the graph enhanced embedding size will be $4 \times D$.

Context-aware Recommendation. Context-aware recommendation aim to further improve performance accuracy and user satisfaction by fully utilizing the contextual information [22]. And the basic context is the candidate item and targeted user. Such as [49] proposed a deep interest network to adaptively learn the user interests from historical behaviors concerning a candidate item. As for the social recommendation, [8] proposed DSCF to consider the local and distant social neighbors' information under the specific recommendation context. And SAMN [2] proposed an attention-based memory module to extract the most related information from the social neighbors. Compared to these methods, our model possesses two key difference: i) For modeling the user's and friends' interest from the interaction history, we further consider the collaborative relations among items as a deep context ; ii) For modeling the item' attraction to the targeted user, we are the first to consider the social relation and collaborative relation among users as a deep social context and capture the more related information from item's past consumers.

6 CONCLUSION

In this paper, we proposed DICER, which utilizes a multi-relation graph neural network to learn the graph information and extract the most related information under the graph enhanced deep context. Our method is equipped with good performance because: i) The multi-relation graph neural network module can capture the high-order relation information in both social graph and collaborative similarity graph. ii) The dual side deep context-aware modulation can model the rich user interest and item attraction from the interaction history. Our comparative experiments and ablation studies on the two benchmark datasets showed that the multi-relation graph neural network module could model the better high-order relation information. The deep context-aware modulation plays a crucial role in both the user and item side.

For future work, we plan to investigate the dual side information fusion strategy further. Moreover, we also seek to deploy our method on real-world recommender systems.

ACKNOWLEDGMENTS

REFERENCES

[1] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral Networks and Locally Connected Networks on Graphs. In *ICLR 2014 : International Conference on Learning Representations (ICLR) 2014*.

[2] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2019. Social attentional memory network: Modeling aspect-and friend-level differences in recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 177–185.

[3] Chong Chen, Min Zhang, Chenyang Wang, Weizhi Ma, Minming Li, Yiqun Liu, and Shaoping Ma. 2019. An Efficient Adaptive Transfer Neural Network for Social-aware Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 225–234.

[4] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). 3844–3852.

[5] Tyler Derr, Yao Ma, and Jiliang Tang. 2018. Signed Graph Convolutional Networks. In *2018 IEEE International Conference on Data Mining (ICDM)*. 929–934.

[6] Wenqi Fan, Qing Li, and Min Cheng. 2018. Deep Modeling of Social Relations for Recommendation.. In *AAAI*. 8075–8076.

[7] 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 on*. 417–426.

[8] Wenqi Fan, Yao Ma, Dawei Yin, Jianping Wang, Jiliang Tang, and Qing Li. 2019. Deep social collaborative filtering. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 305–313.

[9] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 249–256.

[10] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. 2015. Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings.. In *Aaai*, Vol. 15. 123–125.

[11] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. 2016. A Novel Recommendation Model Regularized with User Trust and Item Ratings. *IEEE Transactions on Knowledge and Data Engineering* 28, 7 (2016), 1607–1620.

[12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW '17 Proceedings of the 26th International Conference on World Wide Web*. 173–182.

[13] Guang-Neng Hu, Xin-Yu Dai, Yunya Song, Shu-Jian Huang, and Jia-Jun Chen. 2015. A synthetic approach for recommendation: combining ratings, social relations, and reviews. In *Proceedings of the 24th International Conference on Artificial Intelligence*. 1756–1762.

[14] Yifan Hu, Y. Koren, and C. Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *2008 Eighth IEEE International Conference on Data Mining*. 263–272.

[15] 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*. 135–142.

[16] Meng Jiang, Peng Cui, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2014. Scalable Recommendation with Social Contextual Information. *IEEE Transactions on Knowledge and Data Engineering* 26, 11 (2014), 2789–2802.

[17] Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR 2015 : International Conference on Learning Representations 2015*.

[18] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR 2017 : International Conference on Learning Representations 2017*.

[19] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 426–434.

[20] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* 42, 8 (2009), 30–37.

[21] Adit Krishnan, Hari Cheruvu, Cheng Tao, and Hari Sundaram. 2019. A Modular Adversarial Approach to Social Recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1753–1762.

[22] Wang Li. 2012. Context-Aware Recommender Systems. *Journal of Software* (2012).

[23] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. 2008. SoRec: Social Recommendation Using Probabilistic Matrix Factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)*. Association for Computing Machinery, New York, NY, USA, 931–940.

[24] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*. 287–296.

[25] Peter V. Marsden and Noah E. Friedkin. 1993. Network Studies of Social Influence. *Sociological Methods & Research* 22, 1 (1993), 127–151.

[26] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a Feather: Homophily in Social Networks. *Review of Sociology* 27, 1 (2001), 415–444.

[27] Jiarui Qin, Kan Ren, Yuchen Fang, Weinan Zhang, and Yong Yu. 2020. Sequential Recommendation with Dual Side Neighbor-based Collaborative Relation Modeling. In *WSDM 2020: The 13th ACM International Conference on Web Search and Data Mining*. 465–473.

[28] Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke. 2017. Social Collaborative Viewpoint Regression with Explainable Recommendations. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 485–494.

[29] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[30] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI '09 Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. 452–461.

[31] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[32] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[33] Jiliang Tang, Charu Aggarwal, and Huan Liu. 2016. Recommendations in Signed Social Networks. In *WWW '16 Proceedings of the 25th International Conference on World Wide Web*. 31–40.

[34] Jiliang Tang, Xia Hu, Huiji Gao, and Huan Liu. 2013. Exploiting local and global social context for recommendation. In *IJCAI '13 Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. 2712–2718.

[35] Jiliang Tang, Xia Hu, and Huan Liu. 2013. Social recommendation: a review. *Social Network Analysis and Mining* 3, 4 (2013), 1113–1133.

[36] Rianne van den Berg, Thomas N. Kipf, and Max Welling. 2017. Graph Convolutional Matrix Completion. *arXiv preprint arXiv:1706.02263* (2017).

[37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[38] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2017. 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*. 185–194.

[39] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[40] Le Wu, Junwei Li, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2020. DiffNet++: A Neural Influence and Interest Diffusion Network for Social Recommendation. *CoRR* abs/2002.00844 (2020). arXiv:2002.00844 https://arxiv.org/abs/2002.00844

[41] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A Neural Influence Diffusion Model for Social Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 235–244.

[42] Le Wu, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2021. Collaborative Neural Social Recommendation. *IEEE Transactions on Systems, Man, and Cybernetics* 51, 1 (2021), 464–476.

[43] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. 2019. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *The World Wide Web Conference*. 2091–2102.

[44] Lin Xiao, Zhang Min, Zhang Yongfeng, Liu Yiqun, and Ma Shaoping. 2017. Learning and Transferring Social and Item Visibilities for Personalized Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 337–346.

[45] Bo Yang, Yu Lei, Jiming Liu, and Wenjie Li. 2017. Social Collaborative Filtering by Trust. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 8 (2017), 1633–1647.

[46] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 974–983.

[47] Wenhui Yu, Huidi Zhang, Xiangnan He, Xu Chen, Li Xiong, and Zheng Qin. 2018. Aesthetic-based Clothing Recommendation. In *WWW 2018: International World Wide Web Conferences*. 649–658.

[48] Tong Zhao, Julian McAuley, and Irwin King. 2014. 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*. 261–270.

[49] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1059–1068.