TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 14/16 pp786–798
DOI: 10.26599/TST.2022.9010050
Volume 28, Number 4, August 2023

SGNR: A Social Graph Neural Network Based Interactive Recommendation Scheme for E-Commerce

Dehua Ma, Yufeng Wang*, Jianhua Ma, and Qun Jin

Abstract: Interactive Recommendation (IR) formulates the recommendation as a multi-step decision-making process which can actively utilize the individuals' feedback in multiple steps and optimize the long-term user benefit of recommendation. Deep Reinforcement Learning (DRL) has witnessed great application in IR for e-commerce. However, user cold-start problem impairs the learning process of the DRL-based recommendation scheme. Moreover, most existing DRL-based recommendations ignore user relationships or only consider the single-hop social relationships, which cannot fully utilize the social network. The fact that those schemes can not capture the multiple-hop social relationships among users in IR will result in a sub-optimal recommendation. To address the above issues, this paper proposes a Social Graph Neural network-based interactive Recommendation scheme (SGNR), which is a multiple-hop social relationships enhanced DRL framework. Within this framework, the multiple-hop social relationships among users are extracted from the social network via the graph neural network which can sufficiently take advantage of the social network to provide more personalized recommendations and effectively alleviate the user cold-start problem. The experimental results on two real-world datasets demonstrate that the proposed SGNR outperforms other state-of-the-art DRL-based methods that fail to consider social relationships or only consider single-hop social relationships.

Key words: Interactive Recommendation (IR); Deep Reinforcement Learning (DRL); Graph Neural Network (GNN)

1 Introduction

With the rapid development of technology for decades, the recommendation system is designed to provide users with preferred items to alleviate information overload. Recommendation systems have reshaped the world of e-commerce since online commercial platforms use recommendation systems to provide their customers with added value and increase their profits[1]. Previous research has shown that recommendation systems have a significant and positive impact on sales by reducing search costs and the risks associated with purchasing products that are not popular or are even of poor quality[2].

Recently, the Interactive Recommendation System (IRS) has attracted people's attention[3], because of its ability to optimize long-term user benefits. Unlike that in traditional recommender systems, where the recommendation is treated as a one-step prediction task, the recommendation in IRS is formulated as a multi-step decision-making process[3]. Usually, the decision made at the current step will not only incur

• Dehua Ma and Yufeng Wang are the School of Communications and lnformation Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China. E-mail: wfwang@njupt.edu.cn; mdehua@qq.com.

• Jianhua Ma is with the Digital Media Department, the Faculty of Computer and Information Sciences, Hosei University, Tokyo 194-0298, Japan. E-mail: jianhua@hosei.ac.jp.

• Qun Jin is with the Networked Information Systems Laboratory, Department of Human Informatics and Cognitive Sciences, Faculty of Human Sciences, Waseda University, Tokyo 169-8050, Japan. E-mail: jin@waseda.jp.

∗ To whom correspondence should be addressed.

Manuscript received: 2022-07-05; revised: 2022-08-31; accepted: 2022-10-17

the immediate reward, but also affect the expected rewards probably gained in the future. For instance, the feedback on an item recommended currently by a target user may provide valuable information about his/her interests, which can help the recommender agent make better recommendation decisions in the future[4]. The goal of Interactive Recommendation (IR) is to explore users' latent interests for future recommendation, and meanwhile to exploit the learned user preferences till now to provide accurate recommendations for current IR. That is, the goal is to balance the exploration and exploitation, to optimize the outcome of the entire recommendation sequence[5].

In literature, the IR problem has been modeled as Multi-Armed Bandits (MAB) or contextual bandits problem, which infers the expected reward of recommending items to a user through the balance of exploiting the historical interactions and exploring new feedback[6–8]. However, MAB strategies often use linear reward mapping models, in practical scenarios, the reward function may not be linear, or even be unknown to the recommendation system, which significantly limits the performance of MAB-based IR schemes[6–8].

Due to the powerful expressiveness and learning ability, Deep Reinforcement Learning (DRL) methods are widely used in many fields, like mobile robot navigation[9], video transmission[10], and social evolution modeling[11]. Recently, Deep Reinforcement Learning (DRL)-based models have been used in IR[12–14]. Compared with the MAB-based methods, DRL-based methods can use deep neural networks to model the nonlinear interaction between users and items.

However, most DRL-based IR schemes only exploit user-item feedback data to learn users' latent interests as user state and correspondingly conduct recommendations. Commonly, for recommendation systems serving a huge number of users, the recommender agent may have no or few feedback from users, which lead to the recommender having no idea about users' latent interests or cannot capture users' latent interests accurately. Note that the problem is named as the user cold-start issue in our paper.

In recent years, the emergence of Facebook, Twitter, Weibo, and other multimedia social networks have gradually become an integral part of people's lives[15, 16]. The wide usage of online social media enables the possibility to intentionally incorporate the social information of users to enhance the recommendation system. Social relationships among users play a crucial role in users' decision-making in recommendation systems[17]. Social recommendation systems can take advantage of social relationships among users to significantly alleviate the impact of the lack of users' feedbacks and make a more personalized recommendation[18–20]. Social recommendations are based on social influence theory[21], where socially connected users influence each other in decisions. In the field of social recommendation, several schemes use Graph Neural Networks (GNN) to model multiple-hop social influence among socially connected users[22–24] by diffusing the social influence among users, to alleviate the user cold-start problem. Compared with those schemes that only consider single-hop social relationships[20, 25], these schemes achieve better performance due to the strong representation ability of GNN. However, these GNN-based social diffusion methods are used in single-step recommendation schemes, and their performance for interactive recommendation schemes is rarely investigated.

In this work, we smoothly integrate the interactive feedback of users with the multiple-hop social influence among users to get user state, and then enhance the DRL-based IR performance. That is, when modeling the user state in the DRL-based IR, we not only consider the users' interests, but also capture influence between socially connected users, which is called user social influence vector technically. As users interact with the recommender agent, users' social influence vectors are continuous learned to capture the influence between users and their social neighbors. So, the quality of the user's social influence vector is not compromised by no or few interactions record of the target user. Therefore, for our model, even when the target user has no interaction history, our model can also give users a high-level recommendation according to the user's social influence vector. To the best of our knowledge, this is the first work to introduce multiple-hop social relationships into IRS. Specifically, the contributions of our paper can be summarized as follows.

- First, we propose a Social Graph Neural network-based interactive Recommendation scheme (SGNR) to explicitly characterize the multiple-hop social relationships among users by graph neural network to alleviate user cold-start problem and make more personalized recommendations under the IR environment. That is, we capture influence between socially connected users which is called user social influence vector. And the quality of the user's social influence vector is not compromised by no or few interaction record of the target user. Therefore, for our model, our model can effectively solve the user cold-start problem.

- We present the detailed workflow of each module in SGNR. Moreover, several SGNR variants are intentionally designed and tested, to illustrate the impact of main modules and factors in SGNR on recommendation performance.
- Thorough experiments on two real datasets demonstrate that SGNR performs better than other typically existing IR schemes without incorporating social relationships or only utilizing single-hop social relationships. Moreover, compared with other schemes, the performance of the early phases in IR is explicitly given to illustrate the advantage that SGNR can effectively address the user cold-start issue.

The rest of this paper is organized as follows. Section 2 systematically summarizes the related recommendation schemes according to two comparison dimensions: multi-step (or single-step) and social network-aware (or social network-oblivious). The clear problem statement is formulated in Section 3. Section 4 presents our proposed SGNR framework and its main components. Section 5 gives the real datasets used for experiments and presents thorough simulations and comparison results. A short discussion is presented in Section 6. Finally, we briefly conclude this paper in Section 7.

2 Related Work

As shown in Fig. 1, we categorize various related works according to the dimensions of interactivity and sociality into four classes: single-step social-oblivious recommendation, single-step social recommendation, interactive social-oblivious recommendation, and interactive social recommendation. Note that our approach belongs to the type of interactive social recommendations.

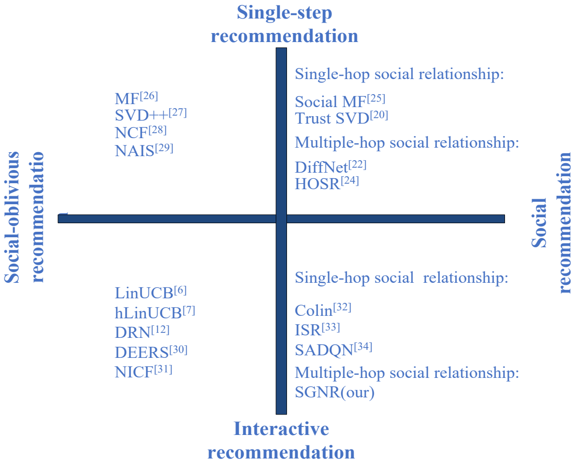
Single-step recommendation
Social-oblivious recommendatio
Social recommendation
Interactive recommendation
MF[26]
SVD++[27]
NCF[28]
NAIS[29]
Single-hop social relationship:
Social MF[25]
Trust SVD[20]
Multiple-hop social relationship:
DiffNet[22]
HOSR[24]
LinUCB[6]
hLinUCB[7]
DRN[12]
DEERS[30]
NICF[31]
Single-hop social relationship:
Colin[32]
ISR[33]
SADQN[34]
Multiple-hop social relationship:
SGNR(our)

Fig. 1 The category of related work.

2.1 Single-step social-oblivious recommendations

Single-step social-oblivious recommendation schemes neither consider the multi-step inter-dependent interactions between users and recommendation systems, nor incorporate the explicit social relationships among users. Matrix Factorization (MF)[26] is a popular recommendation model, in which each user and each item will be embedded in same dimensional latent vector space. The prediction rating matrix is inferred by the inner product of the user latent vector and the item latent vector. SVD++[27] combines the implicit and explicit influence of the interacted item on the user through modeling user's preference with a latent vector learned from ratings and the latent vector of the interacted items. A neural network-based collaborative filtering method is proposed in Neural Collaborative Filtering (NCF)[28], which leverages multi-layer perceptron to learn the user-item interaction function. Considering different items may have a different impact on the user, an attention-based[35] neural network model is proposed for item-based collaborative filtering, which can distinguish the importance of different items in a user profile[29].

2.2 Single-step social recommendation

Data sparseness is a fundamental problem in single-step social-oblivious recommendation methods[36, 37], and social relationships[20] have become an effective resource and can be properly used to alleviate data sparseness problem in the recommendation system since social relationships influence user's decisions. Social MF[25] combines social trust propagation into the MF model by considering the influence of direct social neighbors on the user's preference. TrustSVD[20] improves SVD++[27] by incorporating the explicit and implicit influence of user trust into collaborative filtering. Though these approaches achieve great success, only considering the single-hop social influence among users limits their performance.

Due to the ability of modeling multiple-hop relationships, Graph Neural Networks (GNN) have recently achieved good results in the recommendation system[38, 39]. An influence neural network based model (DiffNet)[22] is proposed, which enforces information aggregation and integration among users in social networks by stacking multiple Graph Convolutional Network (GCN) layers. Similar to Ref. [22], HOSR[24] uses GCN to capture high-order social relationships among users, and an attention mechanism is used to solve the over-smoothing problem in GNN through fusing user embeddings produced by different convolution layers that characterize the social relationships of different hops.

The success of the above single-step social recommendation schemes demonstrate that GNNs have a strong ability to characterize the nonlinear and multiple-hop relationship among users in social networks. However, the effect of GNN in interactive social recommendations is rarely explored.

2.3 Interactive social-oblivious recommendation

The category of IR schemes can be divided into two types: MAB-based and DRL-based. Among MAB-based approaches, contextual bandits algorithms (e.g., LinUCB[6]) are widely used, in which the reward of recommending one item to a specific user is regarded as the dot product between the latent user interests modeled as a linear vector and the extracted item feature. Considering it is difficult to extract all item features, Ref. [7] combines the contextual bandits algorithms with hidden features learning, in which the item feature consists of the observable part and the hidden part (to be learned).

Different from those MAB-based methods, DRL-based approaches naturally use nonlinear reward mapping functions[13], i.e., use neural network structures to model the user-item relationships and explicitly incorporate the long-term returns probably gained by the current recommendation. Basically, among them, due to their simplicity and easy-to-use, Deep Q-Network (DQN)-based methods find wide application in IR[40].

An Deep Reinforcement learning interactive News recommendation scheme (DRN) is proposed by Ref. [12] which uses dueling DQN[41] to predict users' ratings on news and adopts Dueling Bandit Gradient Descent (DBGD)[42] to explore user's interests and improve the diversity of recommendations. DEERS[30] applies Gate Recurrent Unit (GRU) to model user's preference by exploiting both user's positive and negative feedbacks. A Neural Interactive Collaborative Filtering (NICF) method is proposed to solve the interactive recommendation problem[31]. In detail, each user's interests are modeled by all interactive items with the user via a stack of multiple self-attention layers and point-wise feed-forward layers, and the user's rating is predicted by the trained DQN.

2.4 Interactive social recommendation

In this subsection, we discuss some interactive schemes that explicitly utilize the social relationships among users. Collaborative LinUCB (Colin)[32] is a contextual bandit algorithm that explicitly models the underlying dependency among users. Observing that users and some of their friends may have similar interests in certain aspects, but others may be completely irrelevant, Interactive Social Recommendation (ISR)[33] proposes an MAB method that can adaptively learn the trust level between users and their different friends.

Social-Attentive DQN (SADQN)[34] proposes a framework that utilizes social relationships to relieve the user cold-start problem in the DRL-based IR. SADQN divides the action-value function of DQN into personal and social sub-functions. The former evaluates the action value based on the user's interests, and the latter evaluates the action value based on the social-enhanced context of the target user.

However, SADQN only exploits the single-hop social influence among users, while our work intentionally models and characterizes the multiple-hop social relationships among users by the Graph Attention Network (GAT), which, in turn, brings better IR performance, as shown in our experiments.

3 Problem Formulation

We consider a social IR system is composed of two entities, i.e., user set $U(|U| = M)$ and item set $I(|I| = N)$, and a given social network denoted as graph $G = (U, B)$. Specially, U is the user set and B represents the social connections between users. Note that, there are usually two graph types, i.e., directed graph and undirected graph to represent social relationships among users. The former usually characterizes the directed social network, while the latter represents the undirected social network. Since our method works on two social networks, both types of social graphs can be applied to our scheme, and for ease of description and without losing generality, we adopt the undirected graph in this article. In detail, $B \in \mathbf{R}^{M\times M}$ is a symmetric matrix representing the social relationships among users, in which each element $b_{u_1,u_2} = 1$ denotes that user u_1 trusts or has a social relationship with user u_2, and $b_{u_1,u_2} = 0$ indicates that user u_1 do not trust user u_2, where $u_1, u_2 \in U$. Let $R \in \mathbf{R}^{M\times N}$ denotes the user-item feedback matrix, in which each element $r_{u,i} = 1$ indicates that the user u gives positive feedback to the item i (i.e., click the recommended item), and $r_{u,i} = -1$ indicates the opposite (i.e., no click the recommended item). We adopt RL to model and solve the interactive recommendation problem. Specifically, at each time step t, the recommendation agent sends an item $i_t \in I$ to the target user $u_t \in U$ based on the state of the target user which is extracted from the interaction history and social relationships of the target user. After the target user u_t gives feedback on the item i_t, the agent observes the user feedback, updates the user state, and makes the next round of recommendations according to the updated state of each user. The process continues until the user leaves the recommendation system. The goal of the IRS is to learn an optimal recommendation strategy $\pi : S \to I$, to maximize the cumulative reward of the entire recommendation sequence (i.e., maximize user benefit), shown as follows:

$$\pi^* = \mathop{\arg\max}_{\pi \in \Pi} \mathrm{E}\left[\sum_{t=0}^{T} R_{u_t,i_t}\right] \tag{1}$$

where the target user u_t has a state $\boldsymbol{s}_t \in S$ at time step t and S is the set of states, at each time t, state $\boldsymbol{s}_t$ is dynamically learned from the user interaction history and user social relationships, $\mathrm{E}(x)$ is the expectation of x. Π is the set of strategies. R_{u_t,i_t} is an immediate reward for recommending an item i_t to the target user u_t calculated by user feedback, which is abbreviated as R_t in the following paper.

For clarity of presentation, the main notations and their meanings used in the paper are given in Table 1. Note that lowercase bold letters are used to represent vectors and uppercase bold letters to represent matrices.

Table 1 Main notations and descriptions.

Notation	Description
U, I	Sets of users and items, respectively
M, N	Number of users and items, respectively
G	Social network
B	Social relationships matrix
R	User-item feedback matrix
S	Set of users' states
Π	Set of strategies
$N(u)$	Set of 1-hop neighbors of the user u on the social network
$\boldsymbol{i}_t$	Embedding of item i_t at time step t
$\boldsymbol{I}$	Item embedding matrix
$\boldsymbol{Q}$	User embedding matrix
d	Dimension of item embedding
f	Dimension of user embedding
$\boldsymbol{p}_{u,t}$	User u's preference vector at time step t
$\boldsymbol{q}_u^k$	Aggregated social influence vector of the user u after k hops diffusion
$\boldsymbol{s}_{u,t}$	User u's state vector at time step t
$r_{u,i}$	Feedback of user u on the recommended item i
T	Length of the recommendation sequence
θ_V	Parameter of the state value function
θ_A	Parameter of the advantage value function

4 SGNR Framework and Its Components

The proposed SGNR framework is shown in Fig. 2, in which the left part illustrates the interaction between the target user and the recommendation agent, and the right part shows how the recommendation agent learns the user's state by exploiting the user's interaction history and social relationships. Especially, our scheme consists of five modules: the user and item embedding module, the GRU based user dynamic interests modeling module, the GAT based multiple-hop social influence diffusion module, the state fusion module, and the Double Dueling Deep Q-Network (DDDQN) module. The first four modules are used to represent the user's state and action. Then the formed representation of the user's state and action is entered into DDDQN to infer the expected return, and the agent recommends the item with the highest expected return to the target user. Details of these five modules will be presented in the following subsections.

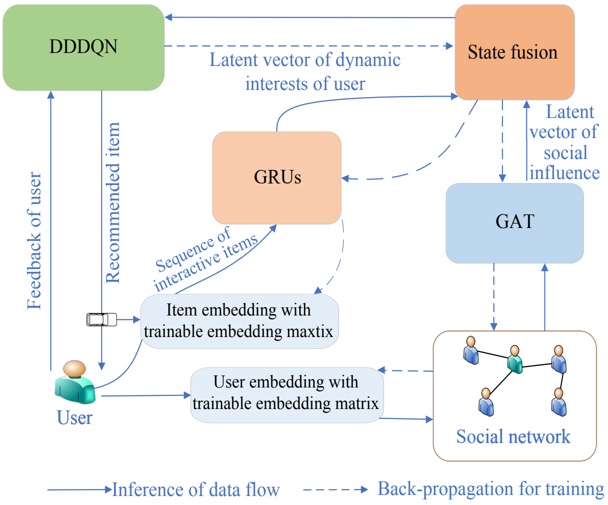
DDDQN
State fusion
Latent vector of dynamic interests of user
Latent vector of social influence
GRUs
GAT
Feedback of user
Recommended item
Sequence of interactive items
Item embedding with trainable embedding maxtix
User
User embedding with trainable embedding matrix
Social network
Inference of data flow
Back-propagation for training

Fig. 2 Our proposed SGNR framework.

4.1 Item and user embedding module

This module makes a vectorized representation of users and items. The shallow embedding method is used to embed users and items in low-dimensional vectors, where the encoder function is a simple embedding matrix. Specifically, each item has a corresponding low-dimensional learnable embedding vector (i.e., item embedding) $\boldsymbol{i} \in \mathbf{R}^d$, and the shallow encoder function is simply a learnable embedding matrix $\boldsymbol{I} = \theta^I \in \mathbf{R}^{d \times N}$, where d is the item embedding dimension, and N is the number of items. Similarly, the user embedding $\boldsymbol{q} \in \mathbf{R}^f$ can be obtained from the shallow encoder function $\boldsymbol{Q} = \theta^Q \in \mathbf{R}^{f \times M}$, where f is the user embedding dimension, and M is the number of users.

4.2 GRU based dynamic interest modeling

In this paper, the recommender agent only recommends one item to the target user at each time t. Since the user's rating (i.e., the explicit feedback) is not always available, it is very common to use implicit feedback in the recommendation system[12, 34]. Similarity, in our paper, we consider the recommendation problem with implicit feedback. That is, we regard user's clicking items as user's positive feedback, and user's non-clicking items as user's negative feedback. In general, the user's interests can be inferred from her/his historically clicked items. Similarly, the negative feedback of a user (i.e., non-clicking on the recommended items) also contains the user's interests. Similar to DEERS[30], we use both user's positive items and negative items to capture the user's interests via GRUs. For ease of description, we first give a solution that does not distinguish between positive and negative feedback to characterize the user's dynamic interests, and then we extend it to using both positive and negative feedbacks of the user.

Given the user u's interaction records denoted as $I_{u,t}$ Since IR is a sequential decision-making process, the recommendation system needs to recommend items according to current user interests. To capture the sequential information of the user-item interactions and drifting of user's interests, the Recurrent Neural Network (RNN) with a GRU[43] is used to model user's dynamic interests. The update function of a GRU cell is defined,

$$\begin{aligned} z_t &= \sigma_g \left(\boldsymbol{W}_z \boldsymbol{i}_t + \boldsymbol{U}_z \boldsymbol{h}_{t-1} + \boldsymbol{b}_z \right), \\ r_t &= \sigma_g \left(\boldsymbol{W}_r \boldsymbol{i}_t + \boldsymbol{U}_r \boldsymbol{h}_{t-1} + \boldsymbol{b}_r \right), \\ \widehat{\boldsymbol{h}_t} &= \sigma_h \left(\boldsymbol{W}_h \boldsymbol{i}_t + \boldsymbol{U}_h \left(r_t \boldsymbol{h}_{t-1} \right) + \boldsymbol{b}_h \right), \\ \boldsymbol{h}_t &= (1 - z_t) \boldsymbol{h}_{t-1} + z_t \widehat{\boldsymbol{h}_t} \end{aligned} \tag{2}$$

where $\boldsymbol{W}_z$, $\boldsymbol{W}_r$, $\boldsymbol{W}_h$, $\boldsymbol{U}_z$, $\boldsymbol{U}_r$, $\boldsymbol{U}_h$, $\boldsymbol{b}_z$, $\boldsymbol{b}_r$, and $\boldsymbol{b}_h$ are leanable parameters in GRU, and $\sigma_g()$ and $\sigma_h()$ are nonlinear functions. $\boldsymbol{i}_t$ denotes the input vector (i.e., the interacted item embedding with the target user in time t), z_t and r_t denote the update gate and reset gate, respectively. And the hidden state $\boldsymbol{h}_t$ is used to represent the user's dynamic interests vector, that is, $\boldsymbol{p}_{u,t} = \boldsymbol{h}_t$. To simplify representation, the above process is formulated in the following:

$$\boldsymbol{p}_{u,t} = GRU\left(I_{u,t}\right) \tag{3}$$

In our work, two GRU networks, i.e., GRU_+ and GRU_- are utilized to represent the user's dynamic interests from the positive and negative feedbacks, respectively. That is, GRU_+ is used to process the sequence of the target user's positive items, while GRU_- is used for the negative items. Therefore, in time step t, for the target user u with negative items set $I_{u,t,-}$, the rating of u for each item $i^- \in I_{u,t,-}$ is negative, i.e., $r_{u,i^-} = -1$, and the positive items set $I_{u,t,+}$, the rating of u for each item $i^+ \in I_{u,t,+}$ is positive, i.e., $r_{u,i^+} = 1$. For the positive items set $I_{u,t,+}$, his/her dynamic interests can be represented as follows:

$$\boldsymbol{p}_{u,t} = GRU_-\left(I_{u,t,-}\right) \| GRU_+\left(I_{u,t,+}\right) \tag{4}$$

where $\|$ is the concatenation operator, note that other operators of merging two vectors could also be used.

Therefore, when the target user continually interacts with the recommender agent, the user's dynamic interests can be updated by incorporating the newly interacted item's embedding into two GRUs. We use θ^{GRUs} to represent the learnable parameters in two GRUs.

4.3 GAT based multiple-hop social influence diffusion

In addition to the user's dynamic interests extracted from the historically interactive items, the social influence among users in social networks plays a key role in the decision-making of users. Specifically, an individual user's decision will be affected by socially connected users (e.g., friends, friends of friends, etc.), therefore it is imperative to actively exploit the multiple-hop social influence among socially connected users for an accurate recommendation. Moreover, each user has distinctive social relationships in the social network, so characterizing the multiple-hop social relationships among users is helpful to make personalized recommendations.

Let $\boldsymbol{q}^k \in f$ represents the f-dimensional aggregated social influence vector of any user after k-hop social diffusion, so-called k-hop social influence vector, and the user's embedding $\boldsymbol{q}$ is regarded as 0-hop social influence vector $\boldsymbol{q}^0$.

The spread of each user's social influence on social networks is implemented by feeding the user's social influence vector into the social influence diffusion module. The entire social influence diffusion module consists of K social influence diffusion layers and over time the user social influence in multi-layer structures spreads throughout the social networks.

Technically, social influence diffusion module is GAT with the learnable parameters θ^{GAT}, which is used to communicate and aggregate users' social influences in social network. And social influence diffusion layer is graph attention layer, in which a weighted sum is used as the sum-aggregation of a user and the user's 1-hop neighboring users.

Specifically, let $\boldsymbol{q}_u^k$ represent the aggregated k-hop social influence vector of the user u, and by feeding it into the k-th social influence diffusion layer, the updated social influence vector $\boldsymbol{q}_u^{k+1}$ can be characterized as aggregating the social influence vector of the user u and the k-hop social influence vectors of his/her neighbors,

$$\boldsymbol{q}_u^{k+1} = \mathrm{ReLU}\left(\sum_{u' \in N(u) \cup u} \beta_{u,u'}^k \boldsymbol{W}^k \boldsymbol{q}_{u'}^k\right) \tag{5}$$

where $\mathrm{ReLU}(x) = \max(0, x)$, $N(u)$ represents the set of the user u's 1-hop social neighbors in the social network, $\beta_{u,u'}^k$ is the aggregation factor between users u and u', $\boldsymbol{W}^k$ is the learnable parameter of the k-th social influence diffusion layer.

Considering the user u's different neighbors may have varying degrees of influence on the target user's decisions, a single-layer neural network is used to represent the aggregation factor $\beta_{u,u'}^k$,

$$\beta_{u,u'}^k = \mathrm{softmax}(e_{u,u'}^k) \tag{6}$$

$$e_{u,u'}^k = \mathrm{LeakyRelu}(\boldsymbol{a}^{\mathrm{T}}[\boldsymbol{W}\boldsymbol{q}_u^k \| \boldsymbol{W}\boldsymbol{q}_{u'}^k]) \tag{7}$$

where $\mathrm{LeakyRelu} = \max(0, x) + 0.01 \times \min(0, x)$, $\boldsymbol{a}$ is a trainable vector, and $\boldsymbol{W}$ is a trainable matrix. After being processed by K social influence diffusion layers, the final social influence vector $\boldsymbol{q}_u^K$ can be obtained, which represents the social influence of K hops social neighbors on the decisions of user u.

4.4 State fusion

After getting the target user's dynamic interests $\boldsymbol{p}_{u,t}$ and the K hops social influence vector $\boldsymbol{q}_u^K$, we need to combine two vectors to get the user's state representation $\boldsymbol{s}_{u,t}$, which is abbreviated as $\boldsymbol{s}_t$. To facilitate state fusion, let $f = 2 \times d$. This paper utilizes two common state fusion methods: addition and concatenation, which are illustrated in the following Eqs. (8) and (9), respectively:

$$\boldsymbol{s}_{u,t} = \boldsymbol{p}_{u,t} + \boldsymbol{q}_u^K \tag{8}$$

$$\boldsymbol{s}_{u,t} = \boldsymbol{W}'(\boldsymbol{p}_{u,t} \| \boldsymbol{q}_u^K) \tag{9}$$

where $\boldsymbol{W}'$ is an $f \times 2f$ matrix that should be learned. Note that our paper named the SGNR using additive fusion and SGNR using concatenation as SGNR-add and SGNR-con, respectively. As we mentioned in Section 4.3, the social influence vector $\boldsymbol{q}_u^K$ captures the social relationships which is unique for each user. So, SGNR-add and SGNR-con can make more personalized recommendations than recommendation schemes that do not consider user social relationships and use user social influence vector to solve user cold-start problem.

4.5 DDDQN based item recommendation

Using the fused state, a specific DRL method, DDDQN is used to conduct item recommendations to optimize the long-term return for IR. Specifically, given a user state representation $\boldsymbol{s}_t$ and an item embedding $\boldsymbol{i}_t$, return of recommending the item to the user, i.e., the Q-value $Q(\boldsymbol{s}_t, \boldsymbol{i}_t; \theta_V, \theta_A)$, will be estimated by state value function $V(\boldsymbol{s}_t)$ and advantage function $A(\boldsymbol{s}_t, \boldsymbol{i}_t)$,

$$Q\left(\boldsymbol{s}_t, \boldsymbol{i}_t; \theta_V, \theta_A\right) = V\left(\boldsymbol{s}_t; \theta_V\right) + A\left(\boldsymbol{s}_t, \boldsymbol{i}_t; \theta_A\right) \tag{10}$$

Here, approximation of both two functions is accomplished by two-layer neural network, where θ_V and θ_A are the parameters of the state value function and advantage function, respectively, denoted as $\theta = \{\theta_V, \theta_A\}$ in our paper.

In the IR process, at time step t, the recommender agent continually observes the user's interaction records $I_{u,t}$ and the social network G. Then, the user's state representation $\boldsymbol{s}_t$ can be obtained by Eq. (8). And then the agent recommends an item i_t via the ε-greedy policy, i.e., with the probability $1 - \varepsilon$ recommending the max-value item and the probability of ε randomly choosing an item. The agent then observes an immediate reward R_t from user's feedback, updates user's interaction records $I_{u,t+1}$ and user's state s_{t+1}, and stores the experience (s_t, i_t, R_t, s_{t+1}) into replay buffer D. In the replay buffer D, the mini-batch experience is sampled, and the components in the proposed SGNR framework are updated by minimizing the mean-square loss function, which is defined in the following:

$$L(\theta, \theta^*) = E_{(s_t, i_t, R_t, s_{t+1}) \sim D}[(y_t - Q(\boldsymbol{s}_t, \boldsymbol{i}_t; \theta, \theta^*))^2] \tag{11}$$

where y_t is the target value of Berman's optimal equation[44], and θ^* is the set of all learnable parameters in our scheme, i.e., θ^I, θ^Q ,θ^{GRUs}, and θ^{GAT}. Parameter θ is updated by gradient descent $\theta = \theta - \eta \nabla_\theta L\left(\theta, \theta^*\right)$, where η is the learning rate, and the gradient is calculated by

$$\nabla_\theta L(\theta,\theta^*) = E_{(s_t,i_t,R_t,s_{t+1})\sim D}\times$$
$$[(y_t - Q(\boldsymbol{s}_t,\boldsymbol{i}_t;\theta,\theta^*))\nabla_\theta Q(\boldsymbol{s}_t,\boldsymbol{i}_t;\theta,\theta^*)] \quad (12)$$

and the updated process of parameters θ^* is similar to parameters θ.

To avoid optimistic estimations, using the double-Q[45] technique, Double DDQN is adopted to compute the target value. That is, y_t is defined as follows:

$$y_t = R_t +$$
$$\gamma Q(\boldsymbol{s}_{t+1}, \text{argmax}_{i_{t+1}} Q(\boldsymbol{s}_{t+1},\boldsymbol{i}_{t+1};\theta,\theta^*);\theta',\theta^*) \quad (13)$$

where R_t represents the observed instantaneous reward of the current decision, γ is the discount factor that balances the current reward and future expected reward. Note that θ' denotes the parameters of the target network. After each iteration, the target network is updated using the following sliding average: $\theta' = \tau\theta + (1-\tau)\theta'$, where $\tau \in [0, 1]$, in our experiments, τ is set as 0.01.

5 Eeperimental Setting and Result

5.1 Description of datasets

Two real datasets are used for experiments: Delicious and LastFM which were widely used in social recommendation[9, 46]. Each dataset contains a user-item feedback matrix and social network. To ensure that there are enough data for training and testing, we remove users and items with less than 5 feedbacks. The basic statistics of the obtained dataset are shown in Table 2.

5.2 Valuation methodology

Due to the interactive nature of IR, an interactive experimental setting is appropriately built with an environment simulator using the offline datasets available[47, 48]. Following Ref. [34], we regard the observable feedbacks of users as user's positive feedbacks, and randomly choose fixed number unobserved user-item pairs as user's negative feedbacks. Specially, in Delicious, 1000 unobserved (u, i) pairs are randomly chosen for each user u as negative feedbacks, and we randomly select 200 unobserved (u, i) pairs in LastFM. In the whole recommendation process, the available item pool for each user consists of the observed positive feedbacks and selected negative feedbacks, and to reduce computational complexity at each time step, we randomly select 100 items in the available item pool as our candidate items.

Following the experiment protocol proposed in Ref. [48], our environment simulator considers the instantaneous feedback and the sequential nature of user behavior. In each episode, the environment simulator samples a user u, and then the recommendation agent interacts with the user u until the end of the episode, the instantaneous reward of the recommended item i to the target user u at time step t is given as follows:

$$R_t = r_{u,i} + \alpha \times (c_p - c_n) \quad (14)$$

where $r_{u,i}$ represents the observed instantaneous feedback of the target user u to the recommended item i, c_p and c_n represent the number of the continuous positive and negative feedbacks of the target user u, respectively. α is a non-negative parameter that controls the balance of instantaneous feedback and sequential behavior. In this paper, following Ref. [48], we conducted experiments in two cases that $\alpha = 0.0$ and $\alpha = 0.1$.

We divide the datasets into two parts, 80% of users are randomly and evenly selected as the training set to train model parameters and the rest 20% of users are used as the test set to test the performance of models. And in each episode, we remove the recommended items in the available item pool to prevent duplicated recommendations.

5.3 Evaluation metrics

To evaluate the performance of our proposed SGNR schemes and compare with other benchmark schemes, three evaluation metrics are adopted: average reward, average precision, and average recall.

Average reward@*T*:

$$\text{Reward@}T = \frac{1}{M_{\text{test}} \times T}\sum_u \sum_{t=1}^{T} R_t,$$

Average precision@*T*:

$$\text{Precision@}T = \frac{1}{M_{\text{test}}}\sum_u \frac{\text{num}_p(T)}{T},$$

Average recall@*T*:

$$\text{Recall@}T = \frac{1}{M_{\text{test}}}\sum_u \frac{\text{num}_p(T)}{\text{totalnum}_p},$$

Table 2 Statistics of datasets.

Item	Number of users	Number of items	Number of observed user feedbacks	Average items per user	Number of observed social relations	Average friends per user
Delicious	1781	6558	178 526	100.2	14 150	7.9
LastFM	1874	2828	71 411	38.1	25 173	13.4

where num_p is the positive feedback number of each user u in a T step episode, totalnum_p is the total positive feedback number of the user, and M_{test} represents the total number of users in the test set.

5.4 Baseline schemes for comparison

We compare SGNR-con with the following six representative baselines in IR environment.

Greedy MF[27] is a well-known collaborative filtering method that trains the MF model interactively using the user-system interactions.

LinUCB[7] is an MAB-based interactive recommendation algorithm, that selects items using the upper confidence interval of estimated rewards based on the item's context information.

hLinUCB[8] is an improved LinUCB method that accommodates the unknown hidden features of items.

DEERS[31] is a DRL-based interactive recommendation method that models user preference considering both positive and negative feedback via GRUs.

NICF[32] is a DRL-based interactive recommendation method that combines self-attention block and DQN to complete IR.

SADQN[34] is a DRL-based method that only considers the single-hop social relationships.

5.5 Hyper-parameter settings

For SGNR, we set the social influence diffusion module layer K to 2 for all datasets. We have tried larger layers and found that the model with larger layers brings lots of computational cost with only limited performance improvement. For all models, we set the item embedding dimension d as 30. All parameters are randomly initialized, and the neural networks in all DRL-based methods take two fully connected layers with an activation function as $ReLU$. The hyper-parameters of all models are chosen by grid search, including learning rate, $L2$ norm regularization, and discount factor γ. In the training set, we set the episode length T to 20, which is consistent with the maximum number of rounds in testing. All trainable parameters are optimized in an end-to-end fashion by the Adam optimizer.

5.6 Overall performance

The performance comparison results are shown in Table 3. The best result in each case is highlighted with a bold number. For performance comparison of all methods, we can obtain the following results.

- First, SGNR (i.e., SGNR-con) has the best performance in all α settings of both datasets. For example, under the setting $\alpha = 0.0$ on LastFM and Delicious, SGNR improves Precision@20 over the best baseline SADQN 2.40% and 2.20%, respectively. This demonstrates that compared with schemes without exploiting social relationships or only utilizing single-hop social relationships, incorporating the multiple-hop social influence among users can provide more personalized recommendations and improve the recommendation performance (i.e., increase user benefit).
- Second, in all DRL-based schemes, schemes that consider the social relationships, i.e., SADQN and SGNR, perform better than other schemes without exploiting social relationships, i.e., DEERS and NICF. It implies that social relationships can improve interactive

Table 3 Overall performance of various IR schemes on Delicious and LastFM datasets.

Dataset	Method	$\alpha = 0$			$\alpha = 0.1$		
		Reward@20	Precision@20	Recall@20	Reward@20	Precision@20	Recall@20
Delicious	Greedy MF	−0.4573	0.2713	0.0545	−0.7871	0.2110	0.0376
	LinUCB	−0.4196	0.2901	0.0510	−0.7098	0.2641	0.0441
	hLinUCB	−0.1553	0.4223	0.0832	−0.2167	0.4129	0.0768
	DEERS	0.2494	0.6274	0.1350	0.3846	0.6192	0.1305
	NICF	0.2501	0.6250	0.1345	0.3472	0.6092	0.1298
	SADQN	0.2883	0.6441	0.1368	0.4363	0.6382	0.1338
	SGNR	**0.3171**	**0.6585**	**0.1429**	**0.4604**	**0.6534**	**0.1404**
LastFM	Greedy MF	−0.1639	0.4180	0.2049	−0.2383	0.3991	0.1939
	LinUCB	−0.0106	0.4946	0.2473	−0.0896	0.4560	0.2249
	hLinUCB	0.0465	0.5232	0.2533	0.0859	0.5128	0.2469
	DEERS	0.3339	0.6669	0.3336	0.5102	0.6627	0.3307
	NICF	0.2651	0.6325	0.3135	0.4279	0.6329	0.3142
	SADQN	0.3571	0.6785	0.3369	0.5535	0.6729	0.3317
	SGNR	**0.3898**	**0.6949**	**0.3472**	**0.6160**	**0.6982**	**0.3490**

recommendation performance.

- Third, in all cases, DRL-based schemes perform better than traditional recommendations, i.e., Greedy MF and MAB-based schemes. The reasons are twofold. On the one hand, the capacity of other non-DRL methods is limited by linear reward mapping functions which are difficult to model complicated user-item interactions. On the other hand, they focus on the immediate item reward, and the effect of the current action on the entire sequence is not considered at all.

5.7 Ablation study

In this section, we further analyze the role of different modules in SGNR via an ablation study. To study the role of different parts, we tested the performance of four different SGNR variants, called SGNR-dp (i.e., DEERS), SGNR-si, SGNR-add, and SGNR-con. The relationship between the four variants is shown in Table 4. Note that, in Table 4, "dp" represents the user dynamic interests module; "si" denotes the social influence diffusion module; "add" and "con" denotes that the user state is fused by addition and concatenation, respectively.

Figures 3 and 4 show the performance in terms of Reward@20 and Precision@20 of these four variants under different α settings in Delicious and LastFM respectively. The following results can be observed.

- First, SGNR-con and SGNR-add perform better than SGNR-dp (i.e., DEERS) in all cases, which demonstrates the effectiveness of the social influence diffusion module of SGNR. This illustrates that the user's social relationships do influence the user's decision, and characterizing the multiple-hop user influence is beneficial for IR tasks.

- Second, SGNR-con and SNGR-add have a better performance than SGNR-si in all cases. This illustrates the effectiveness of the user dynamic interests module of SGNR. The result shows that the users dynamic interests and the social influence among users together determine the user's choices in IR. Considering both user interests and social influence in an IR environment will perform better than that considering only one factor.

Table 4 Structure of various SGNR variants.

Module	SGNR-dp	SGNR-si	SGNR-add	SGNR-con
dp	√	×	√	√
si	×	√	√	√
add	×	×	√	×
con	×	×	×	√

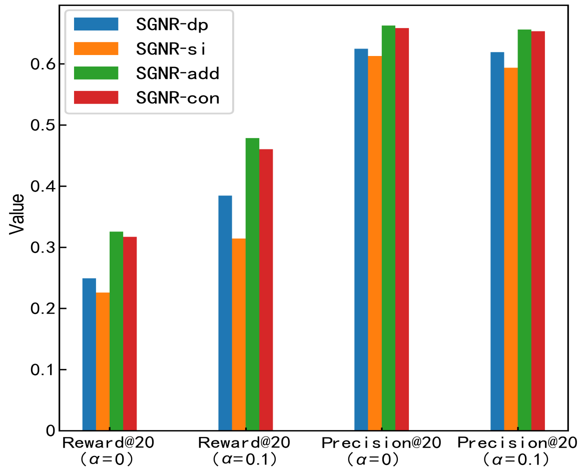

Fig. 3 Performance of four SGNR variants on Delicious.

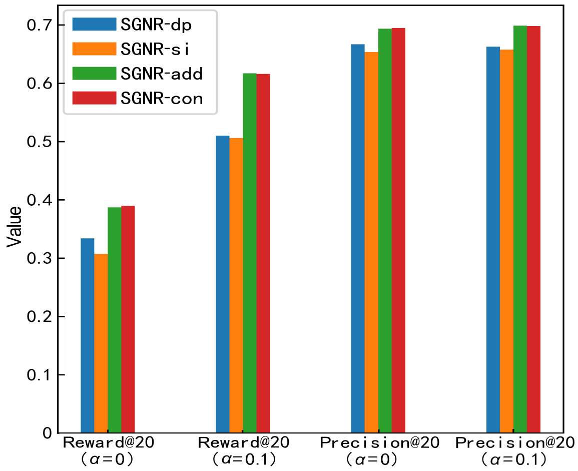

Fig. 4 Performance of four SGNR variants on LastFM.

5.8 Impact of social influence on user cold-start issue

To illustrate the impact of social influence diffusion on recommendation performance in a fine-grained way, we present the precision of SGNR-dp, SGNR-add, and SGNR-con in different test rounds, i.e., $T = 5, 10$, and 20 at $\alpha = 0.1$. The results on Delicious and LastFM are shown in Tables 5 and 6. Note that, in Table 5, the relative improvements of SGNR-add and

Table 5 Precision varying with *T* on Delicious.

T	SGNR-dp	SGNR-add	SGNR-con
5	0.6120	0.7736(26.4%)	0.7773(27.0%)
10	0.6666	0.7545(13.1%)	0.7524(12.8%)
20	0.6627	0.6989(5.4%)	0.6982(5.3%)

Table 6 Precision varying with *T* on LastFM.

T	SGNR-dp	SGNR-add	SGNR-con
5	0.5432	0.6764(24.5%)	0.6771(24.6%)
10	0.5991	0.6731(12.3%)	0.6742(12.5%)
20	0.6192	0.6562(5.9%)	0.6534(5.5%)

SGNR-con against SGNR-dp are shown in the brackets, which imply the effect of social influence on different recommendation rounds.

From Tables 5 and 6, the following results can be observed.

- First, on both datasets, the Precision@5 of SGNR-dp is the lowest among all schemes and is lower than the Precision@10 of SGNR-dp, which is caused by the fact that too few interactions (i.e., $T = 5$) cannot accurately establish the user's interests. That is, without exploiting the social influence existed among users, SGNR-dp suffers from the user cold-start problem.
- Second, interestingly, we can observe that, on both datasets, at the initial recommendation stage, that is, when the number of recommendation rounds is small, i.e., $T = 5$, against SGNR-dp, the Precision@5 improvement of SGNR-con and SGNR-add, is much larger than Precision@10 improvement. It demonstrates that SGNR with fully utilizing the multiple-hop social influence among users, i.e., SGNR-con and SGNR-add, can effectively alleviate the user cold-start issue in DRL-based IR.

6 Discussion

In this work, we show that utilizing user multi-hop social relationship can effectively mitigate the user cold-start problem in IRS. That is, our scheme achieves a higher recommendation performance than other IR method, especially in the early recommendation stage. This is consistent with our previous descriptions. Our scheme can model user preference as well as user multi-hop social relationship to conduct IR, which can achieve a high-quality recommendation even the target user has no or few interactions record. So, in e-commerce, it is very necessary to model multi-hop social relationship by GAT to get a high-quality recommendation, especially in the early recommendation stage.

7 Conclusion and Future Work

In the work, we proposed a multiple-hop social relationships-enhanced DDDQN-based framework SGNR for IR, which combines sequential decision-making processes and social network learning to effectively solve user cold-start issue in IR. In detail, we model the user dynamic interests with the recurrent neural network by utilizing both users positive and negative feedback on items. And the multiple-hop influences among users are diffused and aggregated on the social network via graph attention network, which can effectively exploit the social relationships among users. Extensive experiments on two real-world datasets demonstrates that our scheme can lead to significantly better performance by effectively solving user cold-start problem, compared to other IR schemes.

In our work, it is assumed that the social network among users can be obtained through an exogenous method, e.g., mining the social network platform. However, in some cases, it is difficult to extract the social network prior, thus it is imperative to dynamically learn the social graph during the recommendation, and meanwhile, conduct the social network enhanced DRL-based IR. Generally, it is challenging to conduct graph neural network operations on the dynamically changed social graph and is deserved to be deeply investigated in the future.

References

[1] D. Mican, D. A. Sitar-Tut, and O. I. Moisescu, Perceived usefulness: A silver bullet to assure user data availability for online recommendation systems, *Decis. Support Syst.*, vol. 139, p. 113420, 2020.

[2] B. Pathak, R. Garfinkel, R. D. Gopal, R. Venkatesan, and F. Yin, Empirical analysis of the impact of recommender systems on sales, *J. Manag. Inf. Syst.*, vol. 27, no. 2, pp. 159–188, 2010.

[3] S. Zhou, X. Dai, H. Chen, W. Zhang, K. Ren, R. Tang, X. He, and Y. Yu, Interactive recommender system via knowledge graph-enhanced reinforcement learning, in *Proc. 43rd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, New York, NY, USA, 2020, pp. 179–188.

[4] N. Rubens, M. Elahi, M. Sugiyama, and D. Kaplan, Active learning in recommender systems, in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, eds. Boston, MA, USA: Springer, 2015, pp. 809–846.

[5] X. Zhao, W. Zhang, and J. Wang, Interactive collaborative filtering, in *Proc. 22nd ACM Int. Conf. on Information and Knowledge Management*, San Francisco, CA, USA, 2013, pp. 1411–1420.

[6] L. Li, W. Chu, J. Langford, and R. E. Schapire, A contextual-bandit approach to personalized news article recommendation, in *Proc. 19th Int. Conf. on World Wide Web*, Raleigh, NC, USA, 2010, pp. 661–670.

[7] H. Wang, Q. Wu, and H. Wang, Learning hidden features for contextual bandits, in *Proc. 25th ACM Int. on Conf. on Information and Knowledge Management*, Indianapolis, IND, USA, 2016, pp. 1633–1642.

[8] H. Wang, Q. Wu, and H. Wang, Factorization bandits for interactive recommendation, in *Proc. 31st AAAI Conf. on Artificial Intelligence*, San Francisco, CA, USA, 2017, pp. 2695–2702.

[9] K. Zhu and T. Zhang, Deep reinforcement learning based mobile robot navigation: A review, *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 674–691, 2021.

[10] Y. Xiao, G. Niu, L. Xiao, Y. Ding, S. Liu, and Y. Fan, Reinforcement learning based energy-efficient internet-of-things video transmission, *Intelligent and Converged Networks*, vol. 1, no. 3, pp. 258–270, 2020.

[11] W. Zhang, Z. Hou, X. Wang, Z. Xu, X. Liu, and F. Y. Wang, Parallel-data-based social evolution modeling, *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 878–885, 2021.

[12] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li, DRN: A deep reinforcement learning framework for news recommendation, in *Proc. 2018 World Wide Web Conf.*, Lyon, France, 2018, pp. 167–176.

[13] X. Zhao, L. Xia, L. Zhang, Z. Ding, D. Yin, and J. Tang, Deep reinforcement learning for page-wise recommendations, in *Proc. 12th ACM Conf. on Recommender Systems*, Vancouver, Canada, 2018, pp. 95–103.

[14] Y. Liu, Y. Xiao, Q. Wu, C. Miao, J. Zhang, B. Zhao, and H. Tang, Diversified interactive recommendation with implicit feedback, in *Proc. 34th AAAI Conf. on Artificial Intelligence*, Palo Alto, CA, USA, 2020, pp. 4932–4939.

[15] J. Gu, J. Wang, L. Zhang, Z. Yu, X. Xin, and Y. Liu, Spotlight: Hot target discovery and localization with crowdsourced photos, *Tsinghua Science and Technology*, vol. 25, no. 1, pp. 68–80, 2019.

[16] C. Peng, C. Zhang, X. Xue, J. Gao, H. Liang, and Z. Niu, Cross-modal complementary network with hierarchical fusion for multimodal sentiment classification, *Tsinghua Science and Technology*, vol. 27, no. 4, pp. 664–679, 2022.

[17] K. Yang, J. Zhu, and X. Guo, POI neural-rec model via graph embedding representation, *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 208–218, 2021.

[18] S. Deng, L. Huang, G. Xu, X. Wu, and Z. Wu, On deep learning for trust-aware recommendations in social networks, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 5, pp. 1164–1177, 2017.

[19] J. Tang, X. Hu, and H. Liu, Social recommendation: A review, *Soc. Netw. Anal. Min.*, vol. 3, no. 4, pp. 1113–1133, 2013.

[20] G. Guo, J. Zhang, and N. Yorke-Smith, TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings, in *Proc. 29th AAAI Conf. on Artificial Intelligence*, Austin, TX, USA, 2015, pp. 123–129.

[21] L. Wu, P. Sun, R. Hong, Y. Ge, and M. Wang, Collaborative neural social recommendation, *IEEE Trans. Syst Man Cybernet Syst*, vol. 51, no. 1, pp. 464–476, 2021.

[22] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang, A neural influence diffusion model for social recommendation, in *Proc. 42nd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Paris, France, 2019, pp. 235–244.

[23] L. Wu, J. Li, P. Sun, R. Hong, Y. Ge, and M. Wang, DiffNet++: A neural influence and interest diffusion network for social recommendation, *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4753–4766, 2020.

[24] Y. Liu, C. Liang, 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, 2022.

[25] M. Jamali and M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks, in *Proc. Fourth ACM Conf. on Recommender Systems*, Barcelona, Spain, 2010, pp. 135–142.

[26] Y. Koren, R. Bell, and C. Volinsky, Matrix factorization techniques for recommender Systems, *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[27] Y. Koren, Factorization meets the neighborhood: A multifaceted collaborative filtering model, in *Proc. 14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Las Vegas, NV, USA, 2008, pp. 426–434.

[28] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, Neural collaborative filtering, in *Proc. 26th Int. Conf. on World Wide Web*, Perth, Australia, 2017, pp. 173–182.

[29] X. He, Z. He, J. Song, Z. Liu, Y. G. Jiang, and T. S. Chua, NAIS: Neural attentive item similarity model for recommendation, *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2354–2366, 2018.

[30] X. Zhao, L. Zhang, Z. Ding, L. Xia, J. Tang, and D. Yin, Recommendations with negative feedback via pairwise deep reinforcement learning, in *Proc. 24th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, London, UK, 2018, pp. 1040–1048.

[31] L. Zou, L. Xia, Y. Gu, X. Zhao, W. Liu, J. X. Huang, and D. Yin, Neural interactive collaborative filtering, in *Proc. 43rd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Xi'an, China, 2020, pp. 749–758

[32] Q. Wu, H. Wang, Q. Gu, and H. Wang, Contextual bandits in a collaborative environment, in *Proc. 39th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Pisa, Italy, 2016, pp. 529–538.

[33] X. Wang, S. C. H. Hoi, C. Liu, and M. Ester, Interactive social recommendation, in *Proc. 2017 ACM on Conf. on Information and Knowledge Management*, Singapore, 2017, pp. 357–366.

[34] Y. Lei, Z. Wang, W. Li, H. Pei, and Q. Dai, Social attentive deep Q-networks for recommender systems, *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2443–2457, 2022.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, Attention is all you need, in *Proc. 31st Int. Conf. on Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 6000–6010.

[36] S. Natarajan, S. Vairavasundaram, S. Natarajan, and A. H. Gandomi, Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data, *Exp. Syst. Appl.*, vol. 149, pp. 113–248, 2020.

[37] J. Wei, J. H. He, K. Chen, Y. Zhou, and Z. Y. Tang, Collaborative filtering and deep learning based recommendation system for cold start items, *Exp. Syst. Appl.*, vol. 69, pp. 29–39, 2017.

[38] H. S. Sheu, Z. Chu, D. Qi, and S. Li, Knowledge-guided article embedding refinement for session-based news recommendation, *IEEE Trans. Neural Netw. Learn. Syst.*, doi: 10.1109/TNNLS.2021.3084958.

[39] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, A comprehensive survey on graph neural networks, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2021.

[40] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, Playing Atari with deep reinforcement learning, arXiv preprint arXiv: 1312.5602, 2013.

[41] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, Dueling network architectures for deep reinforcement learning, in *Proc. 33rd Int. Conf. on Machine Learning*, New York, NY, USA, 2016, pp. 1995–2003.

[42] Y. Yue and T. Joachims, Interactively optimizing information retrieval systems as a dueling bandits problem, in *Proc. 26th Ann. Int. Conf. on Machine Learning*, Montreal, Canada, 2009, pp. 1201–1208.

[43] K. Cho, B. Van Merriëboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in *Proc. 2014 Conf. on Empirical Methods in Natural Language Processing* (*EMNLP*), Doha, Qatar, 2014, pp. 1724–1734.

[44] R. S. Sutton and A. G. Barto, *Reinforcement Learning*: *An Introduction*, Boston, MA, USA: MIT Press, 2018.

[45] H. Van Hasselt, A. Guez, and D. Silver, Deep reinforcement learning with double Q-Learning, in *Proc. 30th AAAI Conf. on Artificial Intelligence*, Phoenix, AZ, USA, 2016, pp. 2094–2100.

[46] Z. Guo and H. Wang, A deep graph neural network-based mechanism for social recommendations, *IEEE Trans. Ind. Inf.*, vol. 17, no. 4, pp. 2776–2783, 2021.

[47] Y. Hu, Q. Da, A. Zeng, Y. Yu, and Y. Xu, Reinforcement learning to rank in e-commerce search engine: Formalization, analysis, and application, in *Proc. 24th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, London, UK, 2018, pp. 368–377.

[48] H. Chen, X. Dai, H. Cai, W. Zhang, X. Wang, R. Tang, Y. Zhang, and Y. Yu, Large-scale interactive recommendation with tree-structured policy gradient, in *Proc. 33rd AAAI Conf. on Artificial Intelligence*, Honolulu, HI, USA, 2019, pp. 3312–3320.

Dehua Ma received the BEng degree in communication engineering from Guizhou University, China in 2016. He is currently a master student in telecommunications and information engineering at Nanjing University of Posts and Telecommunications, China. His main research interests include deep reinforcement learning, graph neural network, and interactive recommendation system.

Qun Jin received the BEng degree from Zhejiang University, China in 1982, the MEng degree from Hangzhou Institute of Electronic Engineering (now Hangzhou Dianzi University) and the Fifteenth Research Institute of Ministry of Electronic Industry, China in 1984, and the PhD degree from Nihon University, Japan in 1992. He is currently a professor at the Networked Information Systems Laboratory, Department of Human Informatics and Cognitive Sciences, Faculty of Human Sciences, Waseda University, Japan. He has been extensively engaged in research works in the fields of computer science, information systems, and human informatics. His recent research interests cover human-centric ubiquitous computing, behavior and cognitive informatics, big data, personal analytics and individual modeling, cyber security, blockchain, intelligence computing and applications in healthcare, and computing for well-being.

Jianhua Ma received the BEng and MEng degrees from National University of Defense Technology (NUDT), China in 1982 and 1985, respectively, and the PhD degree from Xidian University, China in 1990. He is currently a professor at Digital Media Department, the Faculty of Computer and Information Sciences, Hosei University, Japan. His main research interest is ubiquitous computing.

Yufeng Wang received the BEng degree from Hefei Normal University, China in 1997, the BEng degree from Xidian University, China in 2000, and the PhD degree from Beijing University of Posts and Telecommunications, Beijing, China in 2004. He is currently a professor at Nanjing University of Posts and Telecommunications, China. From March 2008 to April 2011, he was an expert researcher at National Institute of Information and Communications Technology (NICT), Japan. He is a guest researcher at Advanced Research Center for Human Sciences, Waseda University, Japan. His research interests focus on cyber-physical-social systems, data sciences, etc.