Journal of Southeast University (English Edition) Vol. 38, No. 2, pp. 110 – 117 June 2022 ISSN 1003—7985

Graph-enhanced neural interactive collaborative filtering

Xie Chengyan[1] Dong Lu[2]

([1]School of Automation, Southeast University, Nanjing 210096, China)

([2]School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China)

Abstract: To improve the training efficiency and recommendation accuracy in cold-start interactive recommendation systems, a new graph structure called item similarity graph is proposed on the basis of real data from a public dataset. The proposed graph is built from collaborative interactions and a deep reinforcement learning-based graph-enhanced neural interactive collaborative filtering (GE-ICF) model. The GE-ICF framework is developed with a deep reinforcement learning framework and comprises an embedding propagation layer designed with graph neural networks. Extensive experiments are conducted to investigate the efficiency of the proposed graph structure and the superiority of the proposed GE-ICF framework. Results show that in cold-start interactive recommendation systems, the proposed item similarity graph performs well in data relationship modeling, with the training efficiency showing significant improvement. The proposed GE-ICF framework also demonstrates superiority in decision modeling, thereby increasing the recommendation accuracy remarkably.

Key words: interactive recommendation systems; cold-start; graph neural network; deep reinforcement learning

DOI: 10.3969/j.issn.1003 – 7985.2022.02.002

Personalized recommendation systems have become ubiquitous in the information industry, and they have been applied to classic online services. Traditional recommendation systems have been widely studied under the assumption of a stationary environment, where user preferences are assumed to be static[1–2]. However, such models fail to explore users' interests when few reliable user-item interactions are provided, such as that in a cold-start scenario. They also fail to model the dynamics of user preferences, thus leading to poor performance. Therefore, the research into interactive recommendation systems (IRSs) has flourished in recent years. IRSs consider recommendations as sequential interactions between systems and users. The main idea in modeling IRSs is to capture the dynamic nature of user preferences and achieve optimal recommendations in a time period T[3]. IRS research has two directions: contextual bandit and reinforcement learning (RL). Although contextual bandit algorithms have been used in different recommendation scenes, such as collaborative filtering[4–5] and e-commerce recommendation[6], they are usually invalid in nonlinear models and demonstrate too much pessimism toward recommendations. RL is a suitable learning framework for interactive recommendation tasks as it does not suffer from such issues. In the study of applying RL to IRSs, the themes include large action spaces, off-policy training[7–8], and online model framework[9].

The interactive recommendation problem in the current study is set in a cold-start scenario, which provides nothing about items or users other than insufficient observations of user-item ratings. A deep RL framework[10], which can be regarded as a generalized neural Q-network, is adopted to tackle the above problem. As for the representation of items, an embedding lookup table $\boldsymbol{X} \in \mathbf{R}^{N \times d_e}$ is adopted, with each item e being represented as a vector $\boldsymbol{x}_e \in \mathbf{R}^{d_e}$. The embedding lookup table is trained end to end in the framework. However, because such an embedding layer is optimized by user-item interactions in interactive recommendations and lacks an explicit encoding of crucial collaborative signals, an item similarity graph is proposed, and an embedding propagation layer constructed by graph neural networks (GNNs) is devised in this work to refine items' embeddings by aggregating the embeddings of similar items.

Given the graph structure from the data in recommendation systems, designing a proper graph and utilizing GNNs in recommendation systems are appealing.

User-item bipartite graphs are constructed in traditional recommendation methods for improved performance in rating prediction tasks[11], while sequence graphs are transformed in sequential recommendation methods to capture sequential knowledge[12]. Knowledge graphs[13] are utilized for additional information. By introducing an item similarity bipartite graph in the recommendation framework, we make interactive recommendations effective because of the deep exploitation in structural item similarity information inferred from user-item interactions. A user-item bipartite graph is suggested in the RL

Received 2021-12-03, **Revised** 2022-03-10.

Biographies: Xie Chengyan (1996—), female, graduate; Dong Lu (corresponding author), female, doctor, associate professor, ldong90@seu.edu.cn.

Foundation items: The National Natural Science Foundation of China (No. 62173251), the Guangdong Provincial Key Laboratory of Intelligent Decision and Cooperative Control, the Fundamental Research Funds for the Central Universities.

Citation: Xie Chengyan, Dong Lu. Graph-enhanced neural interactive collaborative filtering[J]. Journal of Southeast University (English Edition), 2022, 38(2): 110 – 117. DOI: 10.3969/j.issn.1003 – 7985.2022.02.002.

framework for interactive recommendations[14].

In sum, a new graph called an item similarity graph is built in this study to alleviate the computational burden while showing comparative structural information as a user-item bipartite graph. Then, a graph-enhanced neural interactive collaborative filtering (GE-ICF) framework, which devises an embedding propagation layer into an RL framework, is proposed for interactive recommendation tasks. Empirical studies on a real-world benchmark dataset are conducted, and the results show that the proposed GE-ICF framework outperforms baseline methods.

1 GE-ICF Framework's Method

1.1 Preliminaries

A typical recommendation system has a set of m users $U = \{1, 2, \ldots, m\}$ and n items $I = \{1, 2, \ldots, n\}$ with an observed feedback matrix $\boldsymbol{Y} \in \mathbf{R}^{m \times n}$, where y_{ij} represents the feedback from user i to item j. Here feedback can be explicit (e.g., rating, like/dislike choice) or be implicitly inferred from watching time, clicks, reviews, etc. For cold-start users in this study, an interactive recommendation process is conducted in a time period T between the recommender agent and a certain user. At each time step t in the interaction time period $\{0, 1, 2, \ldots, T\}$, the recommendation agent calculates item i_t to be recommended by policy π: $s_t \rightarrow I$ and suggests it to certain user u, where s_t represents the user state at time step t. Then, the user gives feedback y_{u,i_t} on the recommended item to the recommender agent, and this feedback guides the agent in updating the user's state and making next-round recommendations. The goal of designing an interactive recommendation system is to design a policy π that maximizes $G_\pi(T)$ as

$$G_\pi(T) = E_{i_t \sim \pi(s_t)} \left(\sum_{t=1}^{T} y_{u,i_t} \right) \tag{1}$$

where $G_\pi(T)$ is the expected cumulative feedback in a time period T. Although exploiting the user state at the current time step facilitates the derivation of accurate recommendations and maximization of immediate user feedback y_{u,i_t}, the exploration of items for recommendation is necessary for completing user profiles and maximizing cumulative user feedback $G(T)$, which is regarded as the delayed reward for a recommendation. RL is a sequential decision learning framework that is aimed at maximizing the sum of delayed rewards from an overall aspect[10]. Therefore, RL is applied in our system to balance exploitation and exploration during interactive recommendations.

The essential underlying model of RL is a Markov decision process (MDP). An MDP occurs between agents and the environment. In this study, the agent is the proposed recommendation system, and the environment is equivalent to the users of the system, as well as all the movies recorded in the system. The MDP is defined with five factors (S, A, P, D, γ). These factors are introduced and instantiated in the IRS for cold-start users. Fig. 1 illustrates the interactive recommendation in the MDP formulation.

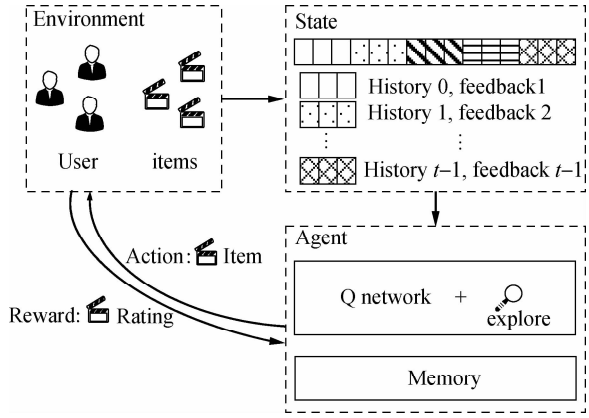

Fig. 1 Interactive recommendation process in MDP formulation

State space S contains a set of states s_t. In this study, a state at time t: $s_t = \{i_0, y_{u,i_0}, \ldots, i_{t-1}, y_{u,i_{t-1}}\}$ denotes the browsing history and corresponding feedback of a user u before time t. To reflect the change of user interests with time, the items in s_t are sorted in chronological order.

Action space A is equivalent to item set I in a recommendation. An action at time t: $a_t \in A$ denotes the item recommended to a user by the recommender system according to current user state s_t.

Reward D is a set the recommender system receives depending on user feedback.

Feedback y_{u,i_t} on the recommended item i_t is returned by user u, and it may be explicit or implicit depending on certain systems. The recommendation system receives immediate reward r_{s_t,a_t} according to the feedback. Rewards may not be the same as feedback; that is, reward shaping technology may be used to improve algorithm performance.

Transition probability $p(s_{t+1} \mid s_t, a_t)$ defines the probability of state transition from s_t to s_{t+1} after an item is recommended as an action. An MDP is assumed to have a Markov property; that is, it satisfies $p(s_{t+1} \mid s_t, a_t, \ldots, s_1, a_1) = p(s_{t+1} \mid s_t, a_t)$. $p(s_{t+1} \mid s_t, a_t) = 1$ is set at any time step, which means that the state at the next time step $t+1$ is determined once state s_t and action a_t are fixed. In this work, the state at $t+1$ is updated by appending action a_t and corresponding feedback y_{u,i_t} to state s_t; that is, it is accumulative.

Discount factor $\gamma \in [0, 1]$ defines the discount factor measuring the importance of future reward in the present

state value. Specifically, $\gamma = 0$ means that the recommender agent only considers the immediate reward while $\gamma = 1$ means that all future rewards are thought to be as important as the immediate reward.

Solving the RL task is to find an optimal policy $\pi_\theta: S \mapsto A$ that maximizes the expected cumulative rewards from a global view. The expected cumulative rewards can be presented by a value function $V(s) = E_{\pi_\theta}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s \right)$ or an action-value function $Q(s,a) = E_{\pi_\theta}\left(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s, a_t = a \right)$. Note that E_{π_θ} is the expectation under policy π_θ, t is the current time step, and r_{t+k} presents the immediate reward at a future time step $t + k$. A variant of neural network $Q(s, a; \theta)$ (i.e., Q-network)[15] is adopted to estimate the policy π_θ. A Q-network adopts the experience replay mechanism and a periodically updated target network to ensure the coverage of the model. A finite-size memory called a replay buffer is applied, and transition samples represented by (s_t, a_t, r_t, s_{t+1}) are stored there for sampling and model training.

In the recommendation procedure, the state space and action space are represented by item vectors. In practice, building item vectors by one-hot encoding is not efficient enough because of the one-hot encoding's extremely high dimension and sparsity, especially in problems with a large action space. Instead, we train dense, low-item vectors end to end in the RL framework. GNNs are integrated into the embedding process because of their superiority in representation learning.

1.2 Item similarity bipartite graph construction

Although a user-item interaction bipartite graph is widely used in collaborative filtering, it suffers from huge data size and a high calculational burden. Therefore, we propose to build an item similarity bipartite graph with the assumption that a user's interest does not change frequently. On the basis of this assumption, we count the frequency of two items simultaneously existing in one user's history. Assume that two items exist in n users' histories; they are thought to be similar if $n \geqslant g$, where g denotes an item similarity coefficient. An edge exists between two similar item nodes in the item similarity graph. We set all edges to have equal weights initially and learn the contribution of each neighbor to central nodes with an attention network. A toy sample of a user-item interaction bipartite graph and an item similarity bipartite graph is illustrated in Fig. 2.

Through the design of the item similarity graph, structural information among items is modeled with the graph size sharply decreasing because user nodes are no longer built in it.

1.3 Model architecture

We now present details of the proposed GE-ICF framework, the architecture of which is illustrated in Fig. 3. The framework is structured with four components: 1) an embedding layer that initializes all item embeddings in the system; 2) an embedding propagation layer that refines the item embeddings by injecting structural item similarity relations; 3) a stacking self-attention block that takes item embeddings and a user's corresponding feedback as input to generate a user profile; 4) a policy layer that predicts the most preferable item for the user. The framework is trained end to end with Q-learning[15].

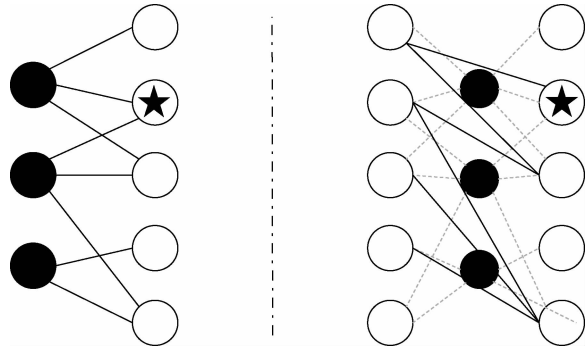

Fig. 2 Illustration of a user-item interaction bipartite graph and an item similarity bipartite graph. (a) User-item interaction bipartite graph; (b) Item similarity bipartite graph

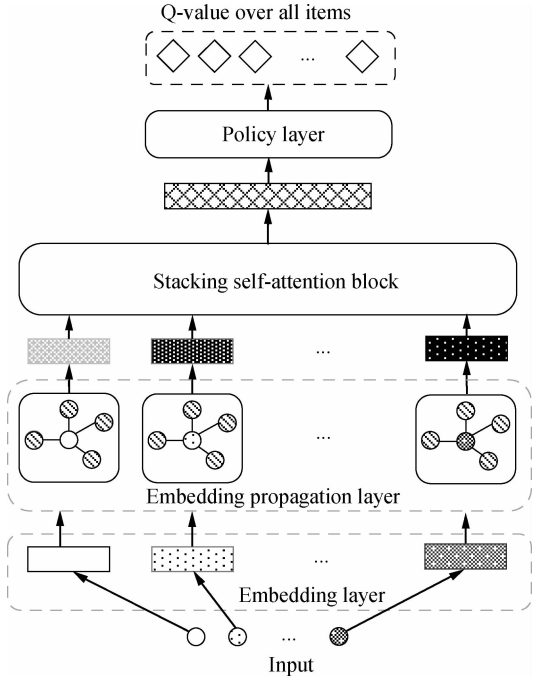

Fig. 3 Architecture of the GE-ICF framework

1.3.1 Embedding layer

Given a user state $s_t = \{i_0, y_{u,i_0}, \ldots, i_{t-1}, y_{u,i_{t-1}}\}$, we first represent items i_t with embedding vectors. We build an embedding lookup table $\boldsymbol{X} \in \mathbf{R}^{N \times d_e}$ for the initialization of all N items' embeddings in the system, with d_e denoting the embedding size. The embedding lookup table is initialized randomly and optimized in an end-to-end style.

In contrast to traditional collaborative filtering methods, which take these ID embeddings as items' final embeddings, they are refined by propagating the information of similar items on an item similarity graph in the GE-ICF framework, thus leading to effective item representations.

1.3.2 Embedding propagation layer

We develop an embedding propagation layer from the idea of GAT. This layer aims to aggregate similar items' features to refine the central nodes' embedding vectors. It takes the embedding lookup table $\boldsymbol{X} \in \mathbf{R}^{N \times d_e}$ and item similarity bipartite graph as input and outputs a graph-aware embedding lookup table $\boldsymbol{X}' \in \mathbf{R}^{N \times d'_e}$, thus transforming an item i's embedding vector from $\boldsymbol{x}_i \in \mathbf{R}^{d_e}$ to $\boldsymbol{x}'_i \in \mathbf{R}^{d'_e}$.

A shared weight matrix $\boldsymbol{W} \in \mathbf{R}^{d_e \times d'_e}$ is necessary in the first step for transforming inputted embedding vectors into high-order features. This step allows the framework to obtain sufficient expressive power. Then, an attention mechanism attention: $\mathbf{R}^{d'_e} \times \mathbf{R}^{d'_e} \rightarrow \mathbf{R}$ is adopted to measure different importance levels of neighbor nodes for central nodes in the form of attention coefficients:

$$e_{ij} = \text{attention}(\boldsymbol{W}\boldsymbol{x}_i, \boldsymbol{W}\boldsymbol{x}_j) \tag{2}$$

where e_{ij} is an attention coefficient calculated to measure the contribution of a neighbor node $j \in N_i$ for the central node i_{central} and N_i denotes all the one-hop neighbors of node i_{central} in the graph as well as the node i_{central} itself. A softmax function is then applied to all attention coefficients e_{i*} as

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})} \tag{3}$$

where α_{ij} is a normalized attention coefficient that makes all the importance levels of nodes in N_i comparable.

We adopt a single-layer feed forward neural network for the attention mechanism, in which the normalized attention coefficient can be expanded as

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(\text{LeakyReLU}(\boldsymbol{a}^{\mathrm{T}}[\boldsymbol{W}\boldsymbol{x}_i \parallel \boldsymbol{W}\boldsymbol{x}_j]))}{\sum_{k \in N_i} \exp(\text{LeakyReLU}(\boldsymbol{a}^{\mathrm{T}}[\boldsymbol{W}\boldsymbol{x}_i \parallel \boldsymbol{W}\boldsymbol{x}_k]))} \tag{4}$$

where $\boldsymbol{a}^{\mathrm{T}} \in \mathbf{R}^{2 \times d'_e}$ is a parameter vector for linear transformation; $\parallel$ is the concatenation operation; LeakyReLU $(\cdot)$ is a function for nonlinearity modeling.

As the central node i_{central} is already contained in the node set N_i, the message propagation process and the message aggregation process can be regarded to be conducted simultaneously by a linear combination of the features corresponding to related nodes and the nonlinearity transformation on the combined embedding vector:

$$\boldsymbol{x}'_i = \sigma\left(\sum_{j \in N_i} \alpha_{ij} \boldsymbol{W}\boldsymbol{x}_j\right) \tag{5}$$

where $\boldsymbol{x}'_i$ is a graph-aware embedding vector of item i_{central}.

We employ multihead attention to stabilize the learning process of self-attention. The final item vectors can be represented by the concatenation or average of K independent attention outputs. We find that concatenation is more sensible to capture graph-aware item representations in this work:

$$\boldsymbol{x}'_i = \parallel_{k=1}^{K} \sigma\left(\sum_{j \in N_i} \alpha_{ij}^k W^k x_j\right) \tag{6}$$

where k is the serial number of each attention head.

1.3.3 Stacking self-attention block

A user profile is then generated by stacking self-attention blocks with user history and the corresponding feedback, and user history is represented with refined item embeddings.

The numbers of items with different user feedback items in a user's history show extreme imbalance; that is, positive feedback items are much fewer than negative feedback items with the assumption that unexposed items are negative samples for users. As we use a dataset within an explicit rating in this work, the items in user history are classified with ratings y_{u,i_t} in user state, and different rated items are processed independently by stacked self-attentive neural networks[10].

1.3.4 Policy layer

With the generated user profile, we apply a two-layer-perceptron (MLP) to extract useful information and model corresponding action-value function $Q_\theta(s_t, \cdot)$ for all items under the current state:

$$Q_\theta(s_t, \cdot) = \text{ReLU}(\boldsymbol{u}_t^{\mathrm{T}} \boldsymbol{W}^{(1)} + \boldsymbol{b}^{(1)})^{\mathrm{T}} \boldsymbol{W}^{(2)} + \boldsymbol{b}^{(2)} \tag{7}$$

where $\boldsymbol{u}_t$ is the user profile vector at timestamp t; $\boldsymbol{W}^{(1)}$, $\boldsymbol{W}^{(2)}$ are the weight matrixes of each perceptron layer; $\boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}$ are the biases of each perceptron layer; ReLU $(\cdot)$ is a function for nonlinearity modeling.

The policy $\pi_\theta(s_t)$ is to recommend item i with maximal Q-value for user u at time t:

$$\pi_\theta(s_t) = \text{argmax}_i Q_\theta(s_t, i) \tag{8}$$

1.4 Model training

We utilize Q-learning[15] to train the whole GE-ICF framework (see Fig. 3). The adopted datasets are divided into training set Γ_{train} and test set Γ_{test} by users. Before the interaction, an item similarity graph is constructed with training users' interactive data Γ_{train} and item similarity coefficient g and is applied to the framework. In the t-th trial, a user state $s_t = \{i_0, y_{u,i_0}, \cdots, i_{t-1}, y_{u,i_{t-1}}\}$ is observed, and the item with the largest value calculated by the approximated value function $Q_\theta(s_t, \cdot)$ is chosen as corresponding recommendation i_t. The ζ-greedy policy is used for exploration during training to enrich the learning sam-

ples. Then, the recommender agent receives the user's feedback y_{u,i_t} on i_t and maps it into reward r_{s_t,i_t}. At the same time, the user state is updated into $s_{t+1} = \{i_0, y_{u,i_0}, \cdots, i_t, y_{u,i_t}\}$. Therefore, a new transition sample $(s_t, i_t, r_{s_t,i_t}, s_{t+1})$ is generated and stored in the memory buffer for batch learning.

We train the weights in the framework in each episode by minimizing the mean squared error:

$$\text{error}(\theta) = E_{(s_t,i_t,r_{u,i_t},s_{t+1})\sim M}[(y_t - Q_\theta(s_t,i_t))^2] \quad (9)$$

where

$$y_t = r_{u,i_t} + \gamma \max_{i_{t+1}\in I} Q_{\theta^-}(s_{t+1}, i_{t+1}) \quad (10)$$

is the target value from the optimized Bellman equation, and the target network[15] is applied to improve system robustness. γ is a discount factor, and $Q_{\theta^-}(s_{t+1}, i_{t+1})$ is the Q-value calculated by the target network. Efficient learning is adopted[10] in this study, with γ set to be dynamic for improved model training. M is a transition sample set stored in the memory buffer.

2 Experiments

We conduct extensive experiments to answer the following questions:

1) Does the application of GNNs refine the item embeddings and improve the recommendation efficiency?

2) Does the designed item similarity graph achieve comparable results to user-item bipartite graphs while sharply decreasing training time?

3) How does the depth of GNNs influence the final recommendation efficiency?

The experimental settings are reviewed first in the following subsection. Thereafter, the questions are discussed in the Results and Analysis section.

2.1 Experimental setting

2.1.1 Datasets

Experiments on recommendation systems should be conducted online to determine their interactive performance. However, online experiments are not always possible as they require a platform and could possibly sacrifice user experience. Therefore, a stable benchmark dataset, MovieLens 100K, is adopted for the experiment in this work. The statistics of the dataset are summarized in Tab. 1.

To make the experiments reasonable, we assume that each item in a user's history in the dataset is the user's instinctive action and is not biased by recommendations. In addition, the ratings from users for items not in their records are assumed to be 0 following existing studies.

Tab. 1 Summary statistics of datasets

Dataset	MovieLens 100K
Users	943
Items	1 682
Interactions	100 000
Interactions per user	106.04
Interactions per item	59.45

2.1.2 Comparison methods

To verify the efficiency of our proposed GE-ICF framework, we select five baselines among different types of recommendations for comparison.

1) Random: A policy uniformly samples items to recommend to users. It is a baseline to output the worst performance, in which no algorithms are used for recommendations.

2) Popular: An algorithm that orders items with the number of ratings and recommends items accordingly. Before the popularity of personalized recommendations, Popular was a most widely adopted policy because of its surprisingly excellent performance on recommendations.

3) Thompson sampling (TS)[5]: An interactive collaborative filtering algorithm achieved with the combination of probabilistic matrix factorization (PMF) and Thompson sampling. Thompson sampling can be replaced with other exploration techniques, such as GLM-UCB. We choose PMF with Thompson sampling as a representation of such techniques to compare it with our algorithm with the goal of balancing exploitation and exploration in recommendations.

4) NICF[10]: A state-of-the-art algorithm that applies RL to interactive collaborative filtering. We refer to its idea on the construction of the DQN-based framework and compare our work with it to verify whether the devised GNNs make sense.

5) GCQN[14]: A DQN-based recommendation that applies a user-item bipartite graph to detect the collaborative signal and uses GRU layers to generate the user profile.

6) GE-ICF: The proposed approach to the interactive recommendation with the item similarity bipartite graph devised.

7) GE-ICF-β: The same architecture as the GE-ICF, except that a user-item bipartite graph is devised in the framework.

We compare GE-ICF and GE-ICF-β to investigate whether the proposed item similarity graph achieves comparable performance in abstracting collaborative signals with the user-item bipartite graph while sharply reducing the burden on calculation.

We adopt cumulative precision during T interactions p_T to evaluate the accuracy of recommendations:

$$p_T = \frac{1}{n_{\text{users}}} \sum_{\text{users}} \sum_{t=1}^{T} b_t \quad (11)$$

where b_t is a parameter indicating whether the recommendation is satisfiable or not and n_{users} is the number of users. $b_t = 1$ if $y_{u,i_t} \geqslant 4$, and 0 otherwise. As we set the reward r_{s_t,i_t} under the same rule, the cumulative precision is

equivalent to the cumulative reward in T interactions.

The dataset is divided into three disjoint sets by users: 85% of the users and their interactions are set as a training set, 5% of the users and their interactions comprise the validation set, and the remaining 10% of the users are set as the test set. In our approach, the batch size of learning is set to be 128, and the learning rate is fixed to 0.001. The memory buffer to replay training samples is set as large as 1×10^6 for sufficient learning, and the exploration factor ζ decays from 1 to 0 during training. The optimizer is chosen to be the Adam optimizer. The item similarity coefficient g is set to be 10. The experiments are conducted on the same machine with a 4-core 8-thread CPU (i5-8300h, 2.30 GHz), Nvidia GeForce GTX 1050 Ti GPU, and 64 GB RAM. We run each model separately five times under five different seeds and average the outputs for the final results.

2.2 Results and analysis

2.2.1 Influence of GNNs

The results of p_T over different models on the dataset MovieLens 100K are reported in Tab. 2, where $T = 10$, 20, 40.

Tab. 2 p_T of different models on MovieLens 100K

Method	$T=10$	$T=20$	$T=40$
Random	0.292 8	0.629 6	1.308 0
Popular	3.282 0	6.124 0	10.660 0
TS	2.170 5	3.564 2	5.465 3
GCQN	2.869 5	4.766 3	6.724 2
NICF	4.656 8	8.162 1	13.762 1
GE-ICF	4.671 6	8.290 5	13.934 7

We compare our proposed framework with five baselines and find that when $T = 10$, 20, and 40, the proposed framework remarkably outperforms the other baselines in terms of recommendation accuracy. This result verifies that the embedding propagation layer we proposed indeed improves the model's capability of detecting collaborative signals and improves the recommendation accuracy in a cold-start scenario.

2.2.2 Efficiency of the proposed item similarity graph

The algorithms GE-ICF and GE-ICF-β are further compared on p_T and seconds per training step (SPT) with $T = 40$ in Tab. 3. Although the precision of GE-ICF-β is slightly higher than that of GE-ICF when T is small, the training time of GE-ICF-β is more than one and a half times as long as that of GE-ICF. This result means that the item similarity bipartite graph achieves comparable results to user-item bipartite graphs while the training efficiency is improved remarkably.

Tab. 3 Performance comparison between GE-ICF and GE-ICF-β on MovieLens 100K

Method	$T=10$	$T=20$	$T=40$	SPT
GE-ICF	4.671 6	8.290 5	13.934 7	60.61
GE-ICF-β	4.717 9	8.317 9	13.903 1	95.73

2.2.3 Influence of GNN depth

To investigate the influence of the GNN layers in the proposed framework, we vary the depths of the GNN layers in the range of $\{1, 2, 3\}$. Tab. 4 summarizes the experimental results, and the results of the framework without GNN layers are presented for reference.

Tab. 4 p_T of the GE-ICF framework with different GNN depths on dataset MovieLens 100K

Layer depth	$T=10$	$T=20$	$T=40$
0	4.656 8	8.162 1	13.762 1
1	4.671 6	8.290 5	13.934 7
2	4.722 1	8.240 0	13.837 9
3	4.602 1	8.027 4	13.467 4

The results in Fig. 4 indicate that although the application of GNN layers improves the recommendation precis-

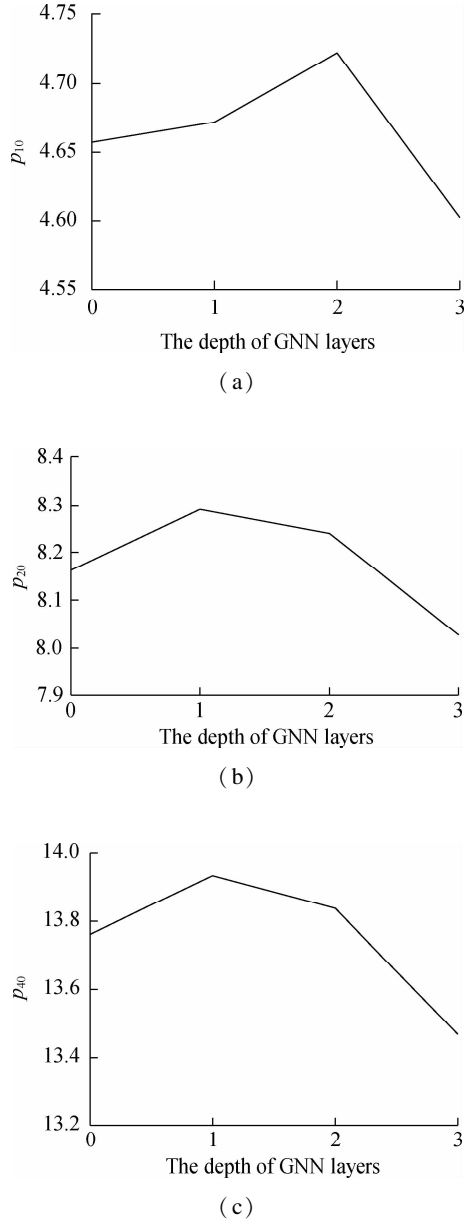

Fig. 4 p_T comparison among GE-ICF framework with different layer depths. (a) p_{10}; (b) p_{20}; (c) p_{40}

ion during time period T, the recommendation performance worsens as the depth of the GNN layers increases. p_{10} achieves the best performance when the GNN layer depth is equal to 1, and the GE-ICF framework with two GNN layers works the best in the time period $T=20$, 40. When the layer depth is up to 3, the recommendation efficiency decreases more sharply, even becoming worse than that of the framework without GNN layers. The reason might be that applying an excessively deep architecture would introduce noise to representation learning. Moreover, the multistacking of GNN layers might bring about an over smoothness issue.

3 Conclusions

1) A GE-ICF framework is proposed in this work to enhance neural interactive filtering performance by recommending GNNs to capture collaborative signals. Extensive experiments are conducted on a benchmark dataset in this work. The results indicate that the recommended GNNs indeed make sense for the training of item embeddings and that the proposed GE-ICF framework outperforms others in interactive recommendation tasks.

2) The proposed item similarity graph is of great significance because it contains as much collaborative information as user-item bipartite graphs while sharply decreasing graph size and shortening training time.

3) Our future work involves several possible directions. Firstly, we would like to investigate how to extend the model by incorporating rich user information (e. g., age, gender, nationality, occupation) and context information (e. g., location, dwell time, device) in a heuristic way. Secondly, we are interested in the effective utilization of RL in IRSs under the guidance of the diversity of recommendations, which is the key indicator of model exploration degree.

References

[1] Wu Y, DuBois C, Zheng A X, et al. Collaborative denoising auto-encoders for top-*n* recommender systems [C]// *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. Los Angeles, CA, USA, 2016: 153 - 162. DOI: 10.1145/2835776.2835837.

[2] Chen X, Xu H, Zhang Y, et al. Sequential recommendation with user memory networks [C]//*Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. Los Angeles, CA, USA, 2018: 108 - 116. DOI:10.1145/3159652.3159668.

[3] Zhao X, Xia L, Tang J, et al. Deep reinforcement learning for search, recommendation, and online advertising: A survey [J]. *ACM SIGWEB newsletter*, 2019: 1 - 15. DOI: 10.1145/3320496.3320500.

[4] Wang H, Wu Q, Wang H. Factorization bandits for interactive recommendation [C]// *Thirty-first AAAI Conference on Artificial Intelligence*. San Francisco, CA, USA, 2017: 2695 - 2702. DOI: 10.5555/3298483.3298627.

[5] Zhao X, Zhang W, Wang J. Interactive collaborative filtering [C]// *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*. Los Angeles, CA, USA, 2013: 1411 - 1420. DOI: 10.1145/2505515.2505690.

[6] Wu Q, Wang H, Hong L, et al. Returning is believing: Optimizing long-term user engagement in recommender systems [C]// *Proceedings of the* 2017 *ACM on Conference on Information and Knowledge Management*. Singapore, 2017: 1927 - 1936. DOI: 10.1145/3132847.3133025.

[7] Zou L, Xia L, Du P, et al. Pseudo Dyna-Q: A reinforcement learning framework for interactive recommendation [C]// *Proceedings of the 13th International Conference on Web Search and Data Mining*. Houston, TA, USA, 2020: 816 - 824. DOI: 10.1145/3336191.3371801.

[8] Chen H, Dai X,Cai H, et al. Large-scale interactive recommendation with tree-structured policy gradient [C]// *Proceedings of the AAAI Conference on Artificial Intelligence*. Honolulu, Hawaii, USA, 2019, **33**(1): 3312 - 3320. DOI: 10.1609/aaai.v33i01.33013312.

[9] Zheng G, Zhang F, Zheng Z, et al. DRN: A deep reinforcement learning framework for news recommendation [C]// *Proceedings of the* 2018 *World Wide Web Conference*. Lyon, France, 2018: 167 - 176. DOI: 10.1145/3178876.3185994.

[10] Zou L, Xia L, Gu Y, et al. Neural interactive collaborative filtering [C]// *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Xi'an, China, 2020: 749 - 758. DOI:10.1145/3397271.3401181.

[11] Wang X, He X, Wang M, et al. Neural graph collaborative filtering [C]//*Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Paris, France, 2019: 165 - 174. DOI: 10.1145/3331184.3331267.

[12] Ma C, Ma L, Zhang Y, et al. Memory augmented graph neural networks for sequential recommendation [C]// *Proceedings of the AAAI Conference on Artificial Intelligence*, New York, USA, 2020, **34**(4): 5045 - 5052. DOI: 10.1609/aaai.v34i04.5945.

[13] Wang H, Zhao M,Xie X, et al. Knowledge graph convolutional networks for recommender systems [C]// *The World Wide Web Conference*. Los Angeles, CA, USA, 2019: 3307 - 3313. DOI: 10.1145/3308558.3313417.

[14] Lei Y, Pei H, Yan H, et al. Reinforcement learning based recommendation with graph convolutional q-network [C]// *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Xi'an, China, 2020: 1757 - 1760. DOI: 10.1145/3397271.3401237.

[15] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning [J]. *Nature*, 2015, **518** (7540): 529 - 533. DOI: 10.1038/nature14236.

图神经网络增强交互协同过滤推荐算法

谢程燕[1] 董 璐[2]

([1] 东南大学自动化学院, 南京 210096)
([2] 东南大学网络空间安全学院, 南京 211189)

摘要:为提升冷启动场景下交互推荐系统的训练效率和推荐精度,基于一个公开数据集的真实数据,根据用户交互构建了一种商品相似度连接图,并设计了基于深度强化学习的图神经网络增强交互协同过滤模型(GE-ICF)来进行仿真实验. 该模型基于深度强化学习框架,采用图神经网络进行向量传播层设计,在商品相似度连接图中挖掘商品间关系,优化商品向量准确度. 结果表明:在冷启动交互推荐场景下,商品相似度连接图能够对大规模稀疏交互推荐数据关系进行高效建模,有效提升训练效率;GE-ICF 模型能够深入挖掘数据间关系,进行更精确地决策建模,有效提高了训练精度.

关键词:交互推荐系统;冷启动;图神经网络;深度强化学习
中图分类号:TP18