

RESEARCH ARTICLE

Research on Neural Graph Collaborative Filtering Recommendation Model Fused With Item Temporal Sequence Relationships

DEWEN SENG¹, MENG FAN LI¹, XUEFENG ZHANG^{1,2}, AND JINGCHANG WANG¹¹Key Laboratory of Complex Systems Modeling and Simulation, Hangzhou Dianzi University, Hangzhou 310018, China²School of Information Engineering, College of Science and Technology, Ningbo University, Ningbo 315300, China

Corresponding author: Xuefeng Zhang (xuefeng0321@163.com)

This work was supported in part by the Fundamental Public Welfare Research Program of Zhejiang Province under Grant LGF22F020036, in part by the Educational Science Planning Project of Zhejiang Province under Grant GH2021562, and in part by the National Experimental Teaching Demonstration Center of Computer Science in China.


ABSTRACT Graph neural network-based recommender systems are blossoming recently, and it can explicitly express user-item high-order connectivity information, so it can significantly improve the recommendation performance. However, the existing methods usually assume that the users' interests are invariant, but temporal relationships is left insufficient exploration to characterize the user's dynamic interest. In this paper, we propose a Neural Graph Collaborative Filtering recommendation model fused with Item Temporal Sequence relationships (NGCF-ITS), that is, a hybrid recommendation model that fuses with user-item interactions information and item temporal sequence relationships. It divides the item temporal sequences into several groups of subsequences through the sliding window strategy, then constructs the item temporal sequence relationships graph and aggregates the characteristics of item temporal sequences information. At the last, deeply depicts the dynamic changes of users' interests, and uses the bipartite graph neural network to map the high-dimensional information of user-item and item-item into the low-dimensional space. The hybrid embedding of user-item historical interactions and item temporal sequence relationships are realized, and the expression of user-item interactions is enhanced. In this way, the heterogeneous multi-relational graphs are fused for the feature propagation, which largely refines the user and item representation for model prediction. Extensive experiments demonstrate the our proposed model significantly improve the recommendation performance compared to the state-of-the-art GNN-based models both in accuracy and training efficiency.

INDEX TERMS Neural graph collaborative filtering, item temporal sequence, sequential recommendation.

I. INTRODUCTION

Recommender systems have been widely applied to daily life such as online shopping, social networking, advertising recommendation and Web search. It can effectively avoid information overload and provide users with accurate personalized information recommendation services. For each user, the goal of the recommender systems is to predict what goods he or she may buy in the future [1], [2], [3]. Collaborative Filtering

(CF) [4], [5], [6] is a classic recommend method to dig users' interests by users' historical behaviors. The core idea of the traditional CF is to recommend users' favorite items with the same preferences [7], [8], such as User-Based CF [9] and Item-Based CF [10], [11]. However, efficient CF-based methods largely depend on the richness degree of the user-item interactions. Therefore, they have poor performance in some scenarios where item information or interactive records are sparse and insufficient, which often leads to the cold start problems in practical marketing [12]. To address the above problems, some researchers have proposed some

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Jiang .

hybrid recommendation models [13], [14], [15] to fuse with multi-source heterogeneous auxiliary information, which can alleviate the problems of cold start and data sparsity. However, the auxiliary information is one of the main challenges building such hybrid recommender methods, such as multimodal, heterogeneous, large-scale, sparse and uneven distribution. On these basis, most CF-based recommender methods mapped user-item interactions into two low-dimensional spaces, for example, Matrix Factorization (MF) [16] and Neural Network [17]. To obtain high-quality embedding representations, most researchs apply graph neural networks (GNNs) to embed user-item interactions, which significantly improves the overall recommendation performance. Tang et al. proposed the unsupervised network embedding algorithm Line [18] and node2vec [19]. Ying et al. [20] proposed a recommendation model PinSage based on GNNs, by adopting methods such as random walk and graph convolution to capture the features of graph structure and nodes to generate embedded representations of nodes, which greatly improves the quality of embedded representation. Wang et al. [21] developed a graph neural collaborative filtering recommender method NGCF, which embedded and encoded the user-item historical interaction information into a bipartite graph, and explicitly considered the high-order connectivity between users-items to further improve the embedded presentation ability. He et al. [22] proposed a light weight graph convolutional network model, LightGCN, for recommender systems, which abandoned the feature transformation and nonlinear activation of traditional GNNs, through propagation to learn user and item embeddings, and finally the weighted summation of the embeddings learned by all layers is used as the final embedding representation. Zhang et al. [23] argued a hybrid recommendation method based on the combination of paragraph embedding and neural network. This method uses neural network to collaborative filter the user's item score, and has high nonlinearity to capture the complex structure of user interaction score. Moreover, the problem of cold start is relieved by using the content characteristics of item embedding to obtain auxiliary information. Wang et al. proposed KGCF [24], which combined knowledge graph and recommendation with GNNs to dig the relationship of goods on knowledge graph, and effectively capture the internal relationship of goods and alleviate data sparsity. Sang et al. [25] proposed a knowledge graph-enhanced neural collaborative filtering model KGCF-RRN, which utilized knowledge graphs and long-term relationships of user-item interactions for recommendation, which effectively solved the sparsity problem of recommender systems. Tang et al. [26] proposed a multi-graph collaborative filtering model (DMGCF) based on a dynamic evolution mechanism, constructed user graph and item graph through user-item bipartite graph and embedding, simulated the edge information in the latent space, and proposed a new dynamic evolution mechanism for collaborative updating during learning and improved embedding and graph structure for better embedded content.

However, most researchs on GNNs only focus on the user-item binary relationships, without considering internal logic relationships between the user-item interactions and the item temporal sequence relationships. The existing methods usually assume that the users' interests are invariant, but temporal relationships is left insufficient exploration to characterize the user's dynamic interest. While intuitively useful to integrate user-item interactions into the embedding function, it is non-trivial to do it well. In particular, the scale of interactions can easily reach millions or even larger in real applications, making it difficult to distill the desired collaborative signal. Also, most methods use the descriptive characteristics of users or items to build embedded functions. When the relationships between users and items becomes greatly complex, these methods will lead to over-fitting, data sparsity and lower recommendation performance. On the other hand, the existing researches focused on users' short-term interests, ignored the impact of item temporal sequences on users' long-term interests or implicit interests. To solve these problems, this paper proposes a hybrid recommendation model, NGCF-ITS, which integrates user-item interactions and item temporal sequence relationships. The illustration of our recommendation model is shown in Figure 1. Our contributions can be summarized as the following:

- 1) We propose a top-N hybrid recommendation model that integrates user-item interaction information and item temporal sequence relationships. We adopt the sliding window strategy to divide the item temporal sequences into several groups of subsequences, build the item temporal sequence relationships graph, and aggregate the information features of the item sequence to describe deeply the dynamic interests.

- 2) Apply the bipartite graph neural network to embed the user-item historical interactions and item temporal sequence relationships, and map the user-item and item-item high-order information into the low-dimensional space to enhance the user-item interactions and improve the recommendation performance.

- 3) Construct a neural graph collaborative filtering recommendation model, the model mainly includes four layers. The four layers refer to the embedding layer, aggregation layer, propagation layer and prediction layer.

- 4) We conduct empirical studies on three standard million-size datasets, LastFM, Ciao and Douban. Extensive results demonstrate the state-of-the-art performance of NGCF-ITS.

II. PRELIMINARIES

This section introduces the problem definition and flowchart of the NGCF-ITS. To facilitate discussion, some symbols are introduced with specific definition as shown in Table 1. The flowchart of the NGCF-ITS is illustrated in Figure 2.

A. PROBLEM DEFINITION

Some symbols are introduced with specific definition as shown in Table 1. Similar to other recommendation algorithms, let $P = [p_1, p_2, p_3, \dots, p_M]$ and $Q = [q_1, q_2, q_3, \dots, q_N]$ be user embedding matrix and item

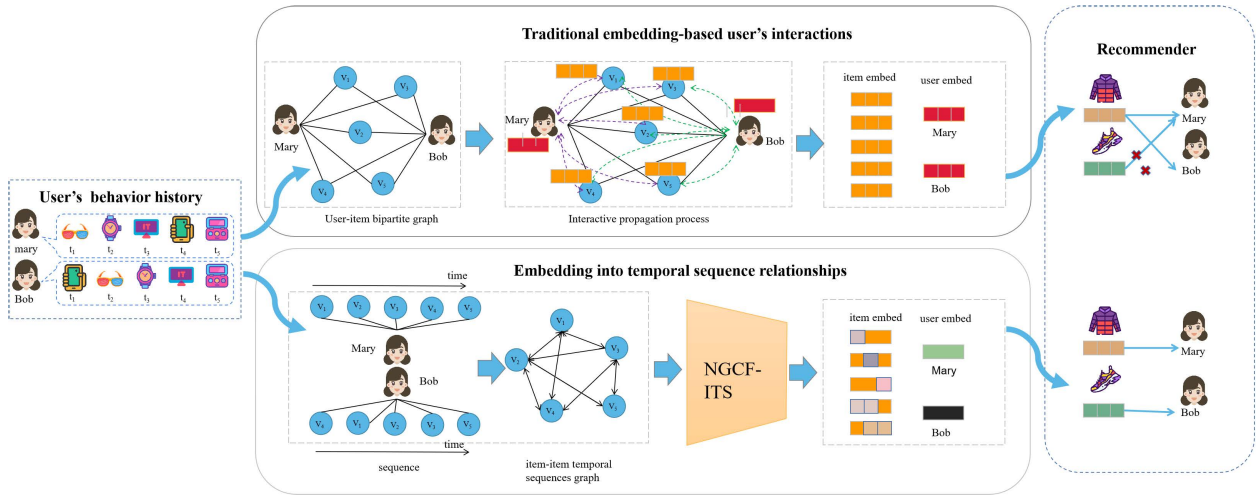


FIGURE 1. Illustration of the proposed NGCF-ITS model.

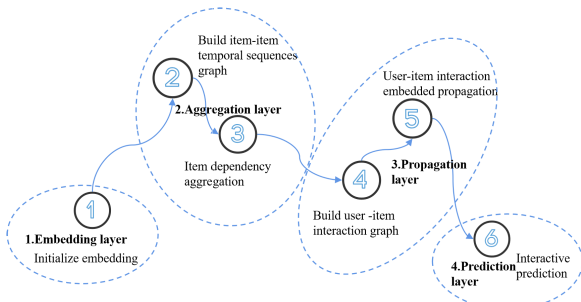


FIGURE 2. The flow chart of NGCF-ITS.

embedding matrix respectively, with $p_u \in R^d$, ($q_i \in R^d$) is embedding vector of user u (item i), d is the dimension in NGCF-ITS, $P \in R^{d \times M}$, $Q \in R^{d \times N}$.

In addition, most existing neural graph network recommendation algorithms describe the user-item interactions by establishing the bipartite graph, and capture the embedding of users and items by utilizing the graph structure methods. The traditional aggregation methods play a crucial role in the message transmission mechanism. However, previous studies only aggregated adjacent information from the perspective of spatial structure information. These methods usually assumed that users' interests are static, and less considered users' dynamic interests changing through the item temporal sequences. Therefore, we construct item-item temporal sequence relationships graph to capture users' dynamic interests changing. Here, we define the item-item directed graph $G < V, E >$, where V is item set, E shows edge set of items. $< V_x, V_y >$ represents an edge between item x and item y . Finally, we create user-item interaction graph $G < u, r, i >$, where u represents user set, i denotes item set, r denotes user-item interaction situation, Where $r = 1$ means this interaction is considered to be a success, or otherwise that is a failure.

TABLE 1. Symbol description.

Symbol	Description
P	User embedding matrix
Q	Item embedding matrix
p_u	User u embedding vector
q_i	Item i embedding vectort
d	Embedding dimension
A	Item-item temporal sequences matrix
E	Item-item temporal sequences unit matrix
\bar{A}	Item-item temporal sequences normalization matrix
$LeakyRelu()$	Activate function
W	Weight matrix
$\varphi()$	Normalization function
$m_{u \rightarrow v}^l$	Embedding propagation from item i to user u
$m_{v \rightarrow u}^l$	Embedding propagation from user u to item v
$m_{u \rightarrow u}^l$	User self-connection
l	layer number
$\sigma()$	Sigmoid activate function
\hat{y}	Predicted value
Θ	All trainable parameters
λ	Control the regular intensity L_2 to prevent over-fitting

B. NGCF-ITS

We now present the proposed NGCF-ITS model, the flowchart of which is illustrated in Figure 2. There are four layers in the flowchart. The detailed working mechanism of each layer is presented as follows.

1) Embedding layer: An embedding layer that offers and initialization of user embeddings and item embeddings. After being initialized in the embedding layer, the user-item potential collaborative signals will be captured in the subsequential training process of the recommendation model Moreover, the node feature in graph will be mapped into low-dimensional space, the embeddings will be iteratively optimized. (step 1)

2) Aggregation layer: To describe users' dynamically interests changing, we design a sliding window strategy which to split the item sequence into fine-grained sub-sequences.

Then, building the item temporal sequence relationships graph, we aggregate the characteristics of item temporal sequences in the graph. Aggregation layers that refine the embeddings by injecting high-order connectivity relations, the aggregation method can find latent features for both users and items similarly. (step2, 3)

3) Propagation layer: We devise an embedding propagation layer, which refines a user's (or an item's) embedding by aggregating the embeddings of the interacted items (or users). By stacking user-item interaction graph and the item-item temporal sequence relationships graph, we can enforce the embeddings to capture the collaborative signal in high-order connectivities. (step 4, 5)

4) Prediction layer: The prediction layer that aggregates the refined embeddings from different propagation layers and outputs the affinity score of a user-item pair, and output prediction results of recommendation. (step 6)

III. METHODS

In this section, We will describe our proposed model research on neural graph collaborative filtering recommendation model fused with item temporal sequence relationships (NGCG-ITS). As shown in figure 2, there are four layers: Embedding layer, Aggregation layer, Propagation layer, Prediction layer.

A. EMBEDDING LAYER

In particular, the scale of user-item interactions can easily reach millions or even larger in real application, it is difficult to distill the desired collaborative signal. In this work, we tackle the change by employing embedding methods based on graph neural network, P and X can be initiated randomly, the user embedding matrix $P = [p_1, p_2, p_3, \dots, p_M]$, where M is the number of users and the item embedding matrix $Q = [q_1, q_2, q_3, \dots, q_N]$ with N is the number of items, $P \in R^{d \times M}$, $Q \in R^{d \times N}$, where d is embedding dimension, we describe a user u (an item i) with an embedding vector $p_u \in R^d$ ($q_i \in R^d$).

We apply the bipartite graph neural network to embed the user-item historical interactions and item temporal sequence relationships, and map the user-item and item-item high-order information into the low-dimensional space to enhance the user-item interactions and improve the recommendation performance.

B. AGGREGATION LAYER

Prior works of GNN-based recommendation only focus on static information of user's interest but ignore the item temporal sequence information. Thus, We describe users' dynamic interests by aggregated embed user-item history interactions and the item temporal sequence relationships to relieve gradient extinction. On the one hand, users' short-term interests and long-term interests should be considered, which can help better reflect a user's interest and alleviate the sparsity issue of user-item interactions. On the other hand, to relieve the computing pressure from users' historical interactions and item temporal sequence relationships, we adopt sliding

window strategy to construct temporal sequence relationships graph. Firstly, we conduct a sliding window strategy to split the item sequence into fine-grained sub-sequences. Then, design a item-item temporal sequence relationships graph for learning the sub-sequences of the same item. At the last, we describe users' interest changing by aggregating historical interactions and temporal sequence relationships. The main description is as follows:

1) CONSTRUCT ITEM-ITEM TEMPORAL SEQUENCE RELATIONSHIPS GRAPH

We incorporate the item-item sequence relationships graph to better indicate the user's interests and alleviate the user-item sparsity issue. Hence, we define the item-item directed graph $G < V, E >$, where $V = \{V_1, V_2, V_3, \dots, V_M\}$ denotes the sequence set of items interacted by users. By splitting V into sub-sequences $T = \{T_1, T_2, T_3, \dots, T_H\}$. Then, we construct item-item temporal sequence relationships graph for learning the sub-sequences of the same items. To overcome the sampling simply, which will lose some important features of user's interest and diversity of item, we use β to denote the length of a subsequence. In the section, to demonstrate the effectiveness of β , we conduct comparative examples with different kinds of parameters setting, including $\beta = 2$ and $\beta = 3$, the detailed process is shown in Figure 3. For example, in Figure 3(a), while $\beta = 3$, the sequence set of user-item historic interactions is $V = \{V_1, V_2, V_3, V_4, V_5\}$, the set of sub-sequence is $T = \{T_1, T_2, T_3, T_4\}$. Where $T_1 = \{V_1, V_2, V_3\}$ represents the 1-th sub-sequence. Meanwhile, $T_2 = \{V_2, V_3, V_4\}$, $T_3 = \{V_3, V_4, V_5\}$, $T_4 = \{V_4, V_5\}$.

Next, Figure 3(b) illustrates the connection process of item-item temporal sequence relationships graph. where N_1 means the number of connected edges, and N_2 represents the number of unconnected edges. As shown in the Figure 3, to analyze the benefits of sliding window method, we vary the number of β in the range of 2, 3, the obvious observation is that the performance tends to better during the increase of β . For example, $\beta = 2$, $N_1 = 4$, $N_2 = 6$, and $\beta = 3$, $N_1 = 7$, $N_2 = 3$. The detailed analysis process will be discussed in the experiment section.

2) AGGREGATE ITEM-ITEM TEMPORAL SEQUENCE RELATIONSHIPS

With the item temporal sequence relationships explosion, the proposed model has become important for embedding user-item interactions. Graph Convolutional Network (GCN) can not only process topological graph structure data, but also iteratively aggregate feature information from neighbor nodes and excavate deeper potential features and internal laws. Therefore, we utilize GCN to aggregate item temporal sequence relationships and capture the higher-order relationships of items by multiple aggregation embedding. The l -th item embedding vector of aggregation layer is:

$$\hat{q}_v^l = f(\hat{q}_v^{l-1}, \sum_{v' \in N_v} m_{v \rightarrow v'}) \quad (1)$$

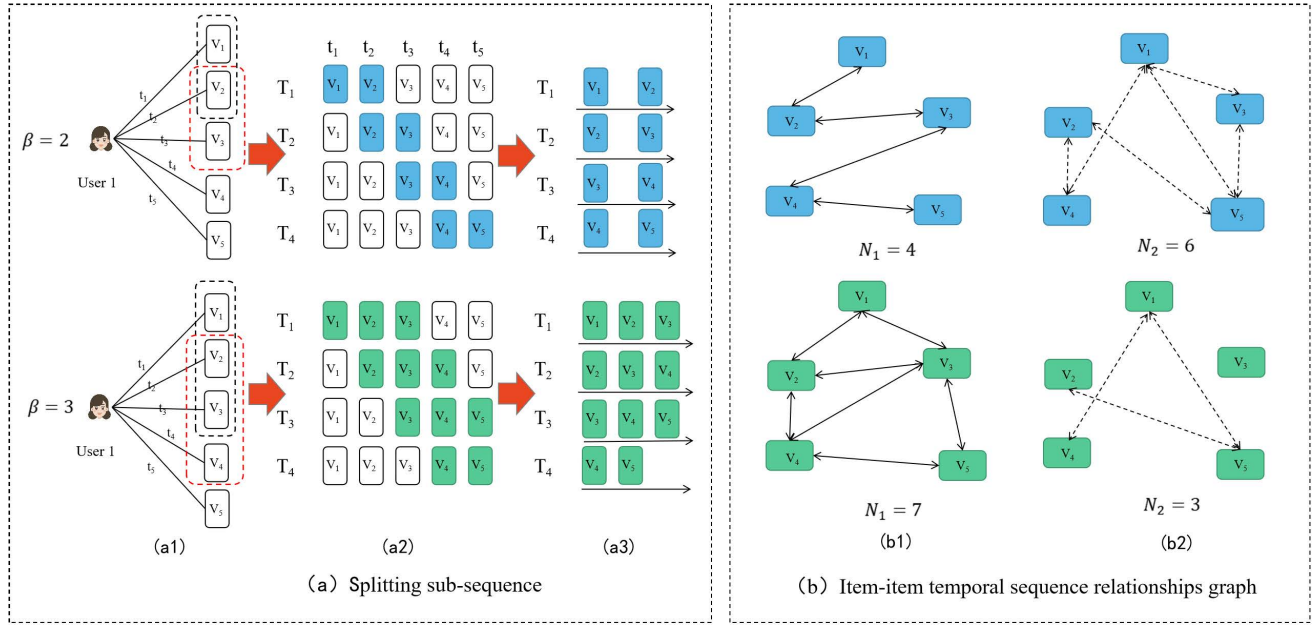


FIGURE 3. The flow chart of NGCF-ITS.

where \hat{q}_v^l is item v embedding vector, when $l = 0$, \hat{q}_v^l is item v initialization embedding vector. $m_{v \rightarrow v'}$ is the propagation of temporal sequence relationships from neighbor item v' , $v' \in N_v$, where N_v is the set of item v neighbor item, $f(\cdot)$ is coding function. In this way, we implement $m_{v \rightarrow v'}$ as:

$$m_{v \rightarrow v'} = \text{LeakyReLU}(W_0^l \hat{q}_v^{l-1} \bar{A}_{v,v'}) \quad (2)$$

$\text{LeakyReLU}(\cdot)$ represents activate function. The normalized matrix $\bar{A} = \varphi(A + E)$, where A and E are item connection matrix and unit matrix, respectively. $\varphi(\cdot)$ is normalized functions. $\bar{A}_{v,v'}$ denotes the normalized node weight value of item v regarding item v' . Item embedding vector of aggregation layer is:

$$\hat{q}_v^l = \text{LeakyReLU}((W_0^l \sum_{v' \in N_v} \hat{q}_{v'}^{l-1} \bar{A}_{v,v'}) + W_0^l \hat{q}_v^{l-1} \bar{A}_{v,v} + b_0^l) \quad (3)$$

W_0^l and d_0^l represent the trainable weight matrices. We consider not only the temporal sequence relationships of items from neighbor node, but also the characteristics of items.

C. PROPAGATION LAYER

In recommender systems, the user-item historical interactions are dominant in model prediction. Hence, we define the user-item interaction graph as $G \langle u, r, i \rangle$, where u and i denote the user and item set respectively, r denotes the situation of user-item inter-action, Where $r = 1$ means success interaction, otherwise that is a failure. Therefore, We apply user-item connection to capture cooperative signals and obtain high order similarity. The user embedding vector of l -th is:

$$p_u^l = \text{LeakyReLU}(m_{u \leftarrow u}^l + \sum_{v \in N_u} m_{u \leftarrow v}^l) \quad (4)$$

where p_u^l represents embedding vector of user u during l -th propagation. Note that in addition to the messages propagated from neighbors U_v , we take the self-connection of u into consideration: $m_{u \leftarrow u}^l = W_{u,1}^l p_u^{l-1}$, which retains the information of original features. $m_{u \leftarrow u}^l$ and $m_{u \leftarrow v}^l$ denote weight matrix. $\text{LeakyReLU}(\cdot)$ is activate function, which allows messages to encode both positive and negative signals. $m_{u \leftarrow v}^l$ is:

$$m_{u \leftarrow v}^l = \frac{1}{\sqrt{|N_u||N_v|}} W_{u,1}^l q_v^{l-1} + \frac{1}{\sqrt{|N_u||N_v|}} W_{u,2}^l (\hat{q}_v^{l-1} \oplus p_u^{l-1}) \quad (5)$$

We define graph Laplacian norm as $\frac{1}{\sqrt{|N_u||N_v|}}$, where N_u, N_v denote the first-hop neighbors of user u and item v , respectively. From the view point of representation learning, it reflects how much contributes of the historical item to the user preference. From the view point of message passing, it can be interpreted as a discount factor, considering the messages being propagated should decay with the path length. To increase the expressiveness of the model, we utilize embedding information \hat{q}_v^{l-1} and use-item interactive information $\hat{q}_v^{l-1} \oplus p_u^{l-1}$. In addition, $W_{u,1}^l, W_{u,2}^l \in R^{d \times d}$ is the weigh matrix.

Similar to user embedding vector, the embedding vector of the item can be written as:

$$p_v^l = \text{LeakyReLU}(W_{v,1}^l \hat{p}_v^{l-1} + \sum_{v \in N_u} \frac{1}{\sqrt{|N_u||N_v|}} W_{v,1}^l p_v^{l-1} + \frac{1}{\sqrt{|N_u||N_v|}} W_{v,2}^l p_v^{l-1} \oplus q_u^{l-1}) \quad (6)$$

D. PREDICTION LAYER

When the aggregation layer and propagation layer are aggregated and propagated for many times, we will find that

multiple operations can not only learn the dependence of item temporal sequence relationships, but also explore the higher-order connectivity. We observe how the learning results are influenced in experiment, since the representations obtained in different layers emphasize the messages passed over different connections, they have different contributions in reflecting user preference. So, we connect $p_u^1, p_u^2, p_u^3, \dots, p_u^l, q_v^1, q_v^2, q_v^3, \dots, q_v^l$ and get final vector of user embedding p_u and vector of item embedding q_v :

$$\begin{aligned} p_u &= p_u^1 \parallel p_u^2 \parallel p_u^3 \parallel \dots \parallel p_u^l \\ q_v &= q_v^1 \parallel q_v^2 \parallel q_v^3 \parallel \dots \parallel q_v^l \end{aligned} \quad (7)$$

where \parallel is concatenation operation, and we define a l -th vector of the user or item embedding. The range of l can not only effectively obtain the embedding vectors of users and items, but also avoid insufficient information extraction. A larger value indicates that the space and time complexity are higher, while a smaller value indicate the opposite. The final precision result is:

$$\hat{y}_{u,v} = p_u^T q_v \quad (8)$$

We regard the observed user-item interactions as positive samples. otherwise, it is regarded as negative samples. The proposed model will be optimized by minimizing the Bayesian Personalized Ranking (BPR) loss. The BPR loss is based on the assumption that a user-item interaction that has already existed in the datasets should be assigned a higher ranking score than an interaction that does not exist in the datasets. The objective function is as follows.

$$Loss = \sum_{(u,v,z) \in O} -\ln(\sigma(\hat{y}_{u,v} - \hat{y}_u, z)) + \lambda \parallel \Theta \parallel_2^2 \quad (9)$$

where $(u, v, z) \in O$, (v, z) represents the training sample sets, which indicates item v interactive with user u , (u, z) is the unobserved interaction. $\sigma(\cdot)$ is the sigmoid function, and $\parallel \cdot \parallel_2^2$ denotes the L_2 regularization term, which is to prevent over-fitting and Θ represents model training parameters.

IV. EXPERIMENTS

We perform experiments on three real-world datasets to evaluate our proposed model, especially the aggregation layer. We aim to answer the following research questions:

1) How does different embedding dimension d affect NGCF-ITS?

2) How do the layer number l of the aggregation layer and the sliding window parameter β affect the recommendation performance?

3) How does NGCF-ITS perform as compared with state-of-the-art models?

A. EXPERIMENTAL SETTINGS

1) DATASETS

To verify the unbiasedness of the model, we conduct experiments on three real-world datasets: LastFM, Douban, and Ciao. These datasets are all publicly accessible, independent

TABLE 2. Specification of the datasets.

Datasets	Users	Items	Interaction	Sparsity
LastFM	1892	17632	92834	0.27%
Douban	2848	39586	894887	0.794%
Ciao	7375	105114	284086	0.0365%

TABLE 3. Analysis of different d on datasets in HR@10.

Datasets	8	16	32	64
LastFM	0.67857	0.76546	0.77346	0.79104
Douban	0.64061	0.66300	0.66148	0.67666
Ciao	0.13823	0.14284	0.14107	0.14243

of each other, and vary in application domain, data size, and data sparsity.

The attributes of the three datasets are shown in Table 2, Ciao datasets obtains higher data sparsity. The obvious observation is that the recommendation performance of the model under different sparsity levels. The details of the datasets are as follows:

1) LastFM: LastFM is an online radio and music community set in the UK. It provides developers with a wealth of APIs which including data of various sizes. We apply the datasets of user listening sequence with implicit feedback of context information released at the 2nd international seminar of the 5th ACM recommendation system conference.

2) Douban: Douban is a well-known theme social networking site in China. The topics include movies, books, music, etc. Users can conduct social discussions on topics of interest, and the content of the discussions can generate cross-information feedback for other users. We conduct experiments on a subset of movie ratings.

3) Ciao: Ciao is a DVD category datasets crawled from <http://dvd.ciao.co.uk> website in December 2013, which covering user ratings for various items and their personal reviews. The scoring mechanism is the scoring interval of [1], [5], and the average of the effective scores of the comprehensive user shows that the score and evaluation will affect the user's willingness to purchase the item.

2) EVALUATION PROTOCOLS

In this section, we take all-ranking protocol and adopt four evaluation metrics: $Precision@N$, $Recall@N$, $NDCG@N$. The detailed introduction is shown as follow:

$$Precision@N = \frac{1}{\bar{U}} \sum_{u \in U} \frac{|R_u \cap I_u|}{N} \quad (10)$$

$$Recall@N = \frac{1}{\bar{U}} \sum_{u \in U} \frac{|R_u \cap I_u|}{I_u} \quad (11)$$

$$DCG@N = \frac{1}{\bar{U}} \sum_{i=1}^N \frac{2^{rel_i} - 1}{\log_2(i + 1)} \quad (12)$$

$$NDCG@N = \frac{1}{\bar{U}} \sum_{u \in U} \frac{DCG@N}{IDCG@N} \quad (13)$$

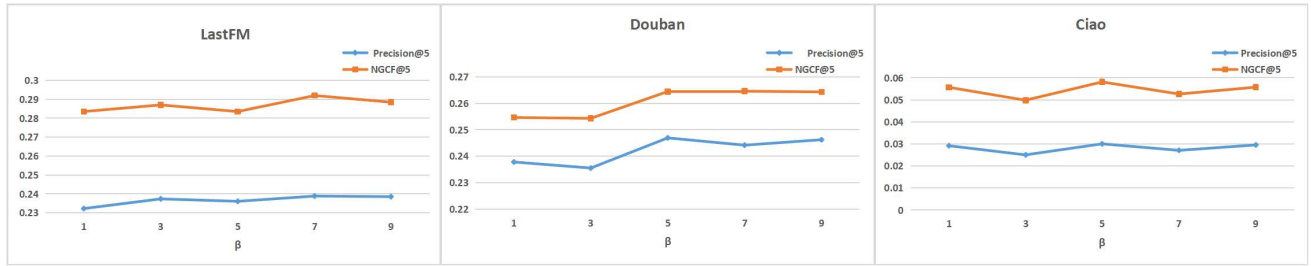


FIGURE 4. Different sliding window sizes were analyzed in three datasets (LastFM, douban, ciao) Precision@5 as well as NDCG@5.

In our experiments, the length of the recommendation N is in 5, 10, 20, 40. \bar{U} represents all users of test set. R_u denotes all item sets which were recommended by user u . I_u means item sets which have interacted with user u . The precision, recall rate and NDCG under N recommended items, respectively denoted as $Precision@N$, $Recall@N$, $NDCG@N$.

3) PARAMETER SETTING

The experiment was conducted on tensorflow platform. In the initialization phase, the embedding dimension of the model is set to $d = 64$, and the batch processing is set to 1024. We initialize the parameters in the model with Xavier, set the learning rate and L_2 regularization to 0.0001 and 10^5 , respectively, and use the Adam optimizer for model optimization training. If the value of $Recall@20$ does not continue to increase for 50 consecutive epochs, the model stops training.

4) BASELINES

In order to test the performance improvement of the NGCF-ITS model, we conduct comparative experiments with some current advanced recommendation models. The models are introduced as follows:

1) APR [27]: Considering that when BPR [28] optimizes MF, the model parameters are susceptible to adversarial disturbances, APR combines BPR with adversarial training methods to make the recommendation model more robust, thus improving the accuracy of model recommendation.

2) CFGAN [29]: A novel CF model based on GAN, which can achieve higher recommendation system accuracy. By adjusting the score vector of the generated user of the G network in the adversarial network, the value range is 0 to 1, which solves the problem of discrete generation in the traditional IRGAN, and also optimizes for collaborative filtering.

3) CUNE_BPR [30]: The CUNE_BPR model extracts implicit social information from user feedback on items, and identifies top-N semantic meta-information for each user, incorporating top-N semantic meta-information into the BPR framework to solve the problem of project sequencing.

4) NeuMF [31]: The NeuMF model combines the linear features of MF and the nonlinear features of DNNS [32], [34], [35] to build latent structures between users and items,

through multiple hidden layers on the element level and the embedded connections between users and items.

5) NGCF [21]: The NGCF model explicitly models the interaction information between users and items, constructs a user-item interaction graph, and on this basis, effectively embeds the collaboration signal into the user-item interaction graph propagation process to achieve high-order connectivity, which improves the embedded representation of users and items.

6) LightGCN [22]: LightGCN model simplifies NGCF model. It eliminates feature transformation and nonlinear activation in NGCF, because these designs have no positive impact on collaborative filtering, and reduces part of the noise of NGCF embedding. Compared with NGCF, LightGCN not only improves the operation efficiency, but also significantly improves the recommendation performance.

B. PERFORMANCE

1) IMPACT OF SLIDING WINDOW SIZE OF β

To investigate whether NGCF-ITS can benefit from sliding window size, we vary the β . In particular, we search the β in the range of 1, 3, 5, 7, 9. Figure 4 summarizes the experiment results, where indicates that the item-item sequence relationships and the users' dynamic interests are affected by sliding window β . We have the following observations in Figure 4:

1) Different sliding window size of NGCF-ITS influences the recommendation cases. NGCF-ITS generally achieves better performance in Douban ($\beta = 5$), Ciao ($\beta = 5$), and LastFM $\beta = 7$.

2) It can be observed that the optimal β is highly dependent on the average length of the neighbors of the datasets, so the optimal β can be set through the properties of the datasets.

2) IMPACT OF THE NUMBER OF LAYERS l

Parameter l controls the intensity of learning, and during the increase of l , the obvious observation is that the recommendation can capture users' dynamic interests accurately.

To verify the performance of l on recommendation model, we set l in range of 0, 1, 2, 3, 4. Figure 5 shows that the experiment results on $Recall@20$, $Precision@20$, $NDCG@20$. NGCF-ITS obtains the worst performance on three datasets with $l = 0$, in other words, the item temporal sequence relationships is not considered. In contrast, the temporal

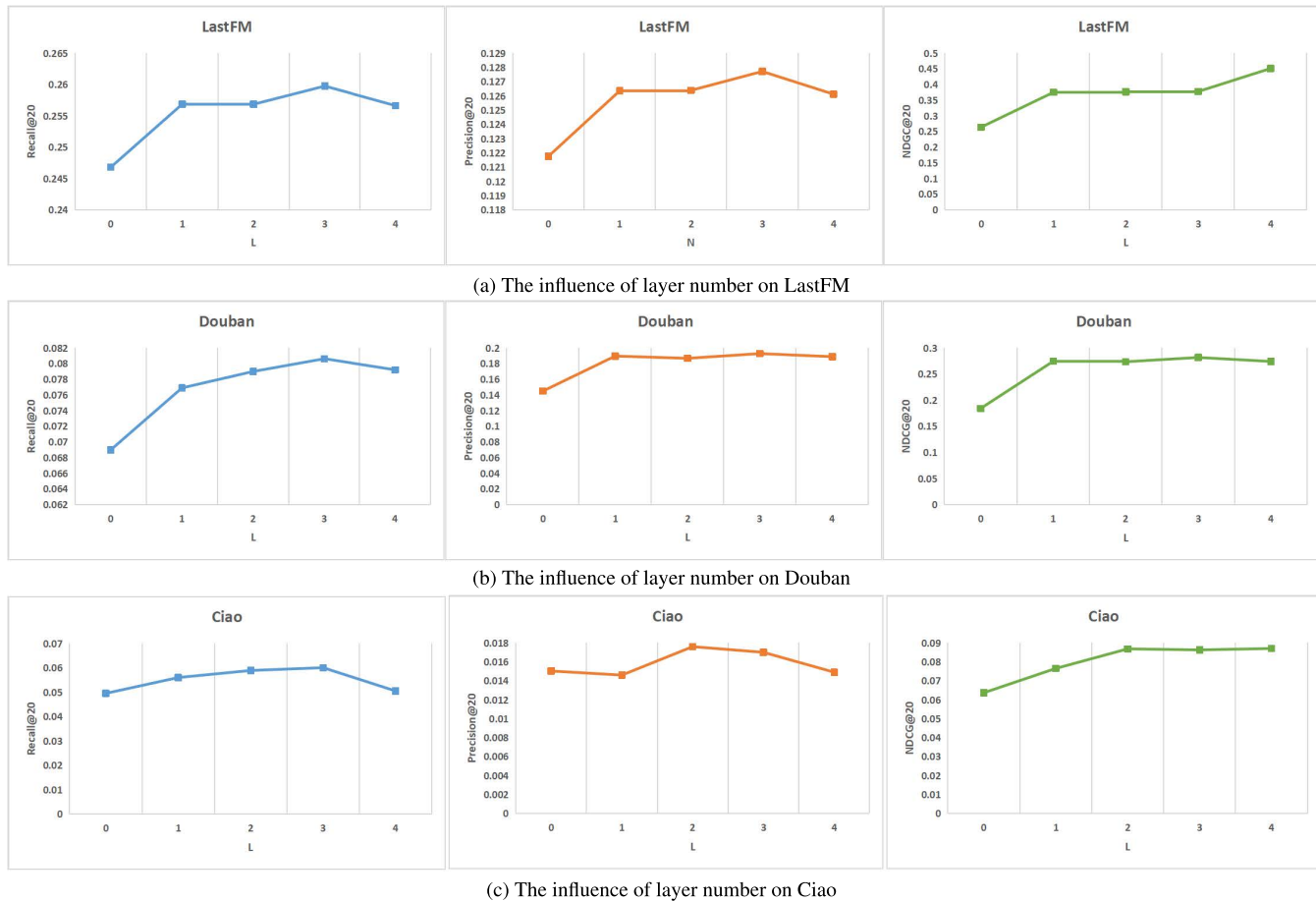


FIGURE 5. The influence of layer number.

sequence relationships can indeed improve recommendation performance.

Compared with LastFM and Douban, due to Ciao is very sparse, the influence of that is relatively modest, which will quickly speed up the convergence of the underlying model and offer better performance.

In addition, during the increase of l , the experiment results is relatively stable at the same time. For example, the result of LastFM and Douban is better with $l = 3$, Ciao is better with $l = 2$. Then, the experiment results declined, the reason might be that the value of l is higher, the feature learning of graph convolution networks on graph structure is too smooth. The learning of item temporal sequence relationships reaches the state of over-fitting, leading to a poor performance.

3) IMPACT OF EMBEDDING DIMENSION d

To analyze the importance of embedding dimension d on three datasets, The dimension d is a multiple of 8, such as $d = [8, 16, 32, 64]$. The experiment data are shown in Table 3, the experiment performance is worst on three datasets with $d = 8$. During the increase of d , the experiment data on LastFM and Douban tend to better, the effect of rising is especially obvious in $d = 8$ to $d = 16$, then the changing

is slowly in $d = 64$, the experiment results reach the best performance.

Different from LastFM and Douban, the experiment results reach the best in Ciao with $d = 16$. the reason for this phenomenon is that Ciao is relatively sparse. In this case, compared with high-latitude embedding, better results can be obtained by training low-dimension embedding, while high-latitude embedding will lead to over-fitting of data and lower recommendation performance, which is also explained in the paper [19].

4) COMPARISON WITH REPRESENTATIVE MODELS

To evaluate our proposed model, the experiments are carried out on three real-world datasets, and to the accuracy performance of the propose model, we compare with other recommendation models, such as the improved CUNE_BPR and APR based on traditional BPRMF, NeuMF based on neural networks, CFGAN based on generative adversarial network, NGCF based on GNN and LightGCN model simplifies NGCF. The above model is described in BASELINES.

1) Table 4 shows the experimental results on LastFM, Douban and Ciao with $N = 5, 10, 20, 40$. we have the following observations: our proposed model NGCF-ITS typically

TABLE 4. Performance compared methods.

Datasets	N	Metrics	APR	CUNE_BPR	CFGAN	NeuMF	NGCF	LightGCN	NGCF-ITS
LastFM	5	Recall@N	0.1064	0.0380	0.0969	0.0839	0.1094	0.1098	0.1132
		Precision@N	0.2061	0.0724	0.1928	0.1653	0.2158	0.2207	0.2253
		NDGC@N	0.2506	0.0762	0.2474	0.2179	0.2638	0.2628	0.2720
	10	Recall@N	0.1621	0.0552	0.1495	0.1293	0.1665	0.1786	0.1766
		Precision@N	0.1576	0.0527	0.1475	0.1270	0.1644	0.1751	0.1725
		NDGC@N	0.2080	0.0677	0.2020	0.1786	0.2167	0.2262	0.2320
	20	Recall@N	0.2346	0.0890	0.2204	0.1909	0.2468	0.2582	0.2558
		Precision@N	0.1146	0.0427	0.1089	0.0941	0.1217	0.1225	0.1263
		NDGC@N	0.2514	0.0869	0.2449	0.2157	0.2639	0.2645	0.2752
	40	Recall@N	0.3222	0.3222	0.3052	0.2748	0.3420	0.3517	0.3501
		Precision@N	0.0788	0.0318	0.0753	0.0676	0.0843	0.0824	0.0863
		NDGC@N	0.2936	0.1061	0.2854	0.2559	0.3101	0.3209	0.3275
Douban	5	Recall@N	0.0232	0.0204	0.0079	0.0203	0.0239	0.0265	0.0254
		Precision@N	0.1000	0.2103	0.0757	0.1503	0.1776	0.2177	0.2255
		NDGC@N	0.1154	0.0801	0.0849	0.1698	0.2001	0.2103	0.2143
	10	Recall@N	0.0389	0.0425	0.0143	0.0346	0.0403	0.0426	0.0436
		Precision@N	0.0928	0.1917	0.0634	0.1413	0.1617	0.1851	0.2049
		NDGC@N	0.1141	0.0796	0.0765	0.1661	0.1913	0.2039	0.2045
	20	Recall@N	0.0652	0.0689	0.0252	0.0605	0.0689	0.0712	0.0709
		Precision@N	0.0865	0.1682	0.0555	0.1298	0.1449	0.1825	0.1896
		NDGC@N	0.1169	0.0818	0.0715	0.1634	0.1842	0.2008	0.1944
	40	Recall@N	0.1116	0.1102	0.0438	0.1062	0.1148	0.1181	0.1211
		Precision@N	0.0780	0.1409	0.0517	0.1165	0.1268	0.1444	0.1510
		NDGC@N	0.1260	0.0945	0.0731	0.1651	0.1820	0.2043	0.1924
Ciao	5	Recall@N	0.0143	0.0168	0.0190	0.0180	0.0175	0.0198	0.0201
		Precision@N	0.0161	0.0138	0.0201	0.0210	0.0209	0.0208	0.0221
		NDGC@N	0.0197	0.0190	0.0212	0.0257	0.0252	0.0284	0.0266
	10	Recall@N	0.0255	0.0251	0.0284	0.0292	0.0299	0.0333	0.0322
		Precision@N	0.0141	0.0109	0.0174	0.0174	0.0180	0.0185	0.0194
		NDGC@N	0.0225	0.0210	0.0246	0.0279	0.0280	0.0291	0.0297
	20	Recall@N	0.0416	0.0381	0.0494	0.0455	0.0495	0.0510	0.0526
		Precision@N	0.0113	0.0084	0.0139	0.0146	0.0150	0.0167	0.0161
		NDGC@N	0.0271	0.0251	0.0281	0.0326	0.0336	0.0343	0.0351
	40	Recall@N	0.0630	0.0575	0.0732	0.0658	0.0759	0.0801	0.0812
		Precision@N	0.0085	0.0063	0.0097	0.0103	0.0115	0.0118	0.0121
		NDGC@N	0.0333	0.0306	0.0364	0.0387	0.0409	0.0428	0.0439

outperformed approaches, which reflects the accuracy and effectiveness. In addition, based GCN model use graph structures for message propagation to achieve performance improvements.

2) NGCF-ITS aggregates item temporal sequence relationships, this not only increases the model representation ability, but also boosts the performance for recommendation, and NGCF-ITS generally achieves better performance than LightGCN in most cases.

3) NGCF-ITS consistently yields the best performance on all the datasets. In particular, NGCF-ITS improves over LightGCN is 4.0% on LastFM, over CUNE_BPR is 6.8% on Douban and over NeuMF is 5.2% on Ciao in Table 4. This phenomenon indicates that aggregation of item temporal sequence relationships into GNN-based recommendations has sufficient development prospects.

V. CONCLUSION

In this paper, we proposed a neural graph collaborative filtering recommendation model fused with item temporal sequence relationships (NGCF-ITS). Specifically, we utilized the sliding window strategy to divide the item temporal sequence relationships into several groups of sub-sequences, and constructed the item temporal sequence relationships

graph to aggregate the item-item temporal sequence relationships and dig users' dynamic interests. We use bipartite graph neural network to map high-dimensional information of user-item and item-item into low-dimensional space to realize the hybrid embedding of user-item historical interaction information and item-item temporal sequence relationships information, expression of information can be enhanced by user-item interaction sequence. Moreover, make use of the user-item interaction graph and item temporal sequence relationships graph to construct a four-layer neural graph collaborative filtering recommendation model framework, including the embedding layer, the aggregation layer, the propagation layer and the prediction layer. The results of comparative experiments on the three real-world datasets of LastFM, Ciao and Douban, which shows that NGCF-ITS model has brought significant improvements compared with other models, meaning that the item temporal sequence relationships is valuable in providing more accurate recommendations.

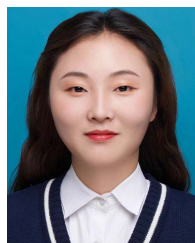
REFERENCES

- [1] T. L. Huang, Y. Dong, M. Ding, Z. Yang, and W. Feng, "MixGCF: An improved training method for graph neural network-based recommender systems," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2021, pp. 665–674.

- [2] S. Zhao, W. Wei, D. Zou, and X. Mao, "Multi-view intent disentangle graph networks for bundle recommendation," in *Proc. 36th AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 4, pp. 4379–4387.
- [3] J. W. Jeong, J. Choi, H. Cho, and S. Chung, "FPAdaMetric: False-positive-aware adaptive metric learning for session-based recommendation," in *Proc. 36th AAAI Conf. Artif. Intell.*, 2022, pp. 4039–4047.
- [4] 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, Dec. 2018.
- [5] J. F. Chen, G. Zhu, H. Hou, C. Yuan, and Y. Huang, "AutoGSR: Neural architecture search for graph-based session recommendation," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2022, pp. 1694–1704.
- [6] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web (WWW)*, 2001, pp. 285–295.
- [7] J. F. Xia, D. Li, H. Gu, J. Liu, T. Lu, and N. Gu, "FIRE: Fast incremental recommendation with graph signal processing," in *Proc. 31st Int. Conf. World Wide Web*, 2022, pp. 2360–2369.
- [8] X. F. Zhang, X. L. Chen, and D. W. Seng, "Top-N recommendation algorithm combining user trust and influence," *J. Zhejiang Univ. (Eng. Sci.)*, vol. 54, no. 2, pp. 311–319, 2022.
- [9] J. W. Qin et al., "A collaborative filtering algorithm based on trust and user's emotional preference," *J. Softw.*, vol. 24, no. S2, pp. 61–72, 2013.
- [10] W. Liu, Y. Xi, J. Qin, F. Sun, B. Chen, W. Zhang, R. Zhang, and R. Tang, "Neural re-ranking in multi-stage recommender systems: A review," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022.
- [11] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.
- [12] J. Guo, Y. Zhou, P. Zhang, B. Song, and C. Chen, "Trust-aware recommendation based on heterogeneous multi-relational graphs fusion," *Inf. Fusion*, vol. 74, pp. 87–95, Oct. 2021.
- [13] F. Q. Zhou and W. Q. Cheng, "Sequence recommendation based on global enhanced graph neural network," *Comput. Sci.*, vol. 49, no. 9, pp. 55–63, 2022.
- [14] C. H. Tsai, P. Brusilovsky, and B. Rahdari, "Exploring user-controlled hybrid recommendation in a conference context," in *Proc. Joint Workshops Co-Located 24th Conf. Intell. User Interface*, 2019, pp. 1–6.
- [15] Y. Qian, Y. Zhang, Y. Ma, H. Yu, and L. Peng, "EARS: Emotion-aware recommender system based on hybrid information fusion," *Inf. Fusion*, vol. 46, pp. 141–146, Mar. 2019.
- [16] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the β -divergence," *Neural Comput.*, vol. 23, no. 9, pp. 2421–2456, 2011.
- [17] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12.
- [18] J. Tang, M. Qu, M. Wang, M. Zhang, and J. Yan, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web Int. World Wide Web Conf. Steering Committee*, 2015, pp. 1067–1077.
- [19] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 855–864.
- [20] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 974–983.
- [21] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2019, pp. 165–174.
- [22] X. N. He, K. Deng, X. Wang, Y. Li, and Y. Zhang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2020, pp. 639–648.
- [23] Y. H. Zhang, Z. Liu, and C. Sang, "Unifying paragraph embeddings and neural collaborative filtering for hybrid recommendation," *Appl. Soft Comput.*, vol. 106, Jul. 2021, Art. no. 107345.
- [24] H. W. Wang et al., "Knowledge graph convolutional networks for recommender systems," in *Proc. World Wide Web Conf.*, no. 7, 2019, pp. 3307–3313.
- [25] L. Sang, M. Xu, S. Qian, and X. Wu, "Knowledge graph enhanced neural collaborative filtering with residual recurrent network," *Neurocomputing*, vol. 454, pp. 417–429, Sep. 2021.
- [26] H. Tang, G. Zhao, X. Bu, and X. Qian, "Dynamic evolution of multi-graph based collaborative filtering for recommendation systems," *Knowl.-Based Syst.*, vol. 228, Sep. 2021, Art. no. 107251.
- [27] X. He, Z. He, X. Du, and T.-S. Chua, "Adversarial personalized ranking for recommendation," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 315–364.
- [28] S. Rendle et al., *BPR: Bayesian Personalized Ranking From Implicit Feedback*. UAI, 2009, pp. 452–461.
- [29] D. K. Chae, J. S. Kang, S. W. Kim, and J. T. Lee, "CFGAN: A generic collaborative filtering framework based on generative adversarial networks," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manag.*, 2018, pp. 137–146.
- [30] C. Z. Zhang et al., "Collaborative user network embedding for social recommender systems," *SDM*, 2017, pp. 381–389.
- [31] X. N. He, L. Z. Liao, H. W. Zhang, L. Nie, and X. Hu, "Neural collaborative filtering," in *Proc. 27th Int. Conf. World Wide Web*, 2017, pp. 173–182.
- [32] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 160–167.
- [33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [34] R. Hong, Z. Hu, L. Liu, M. Wang, S. Yan, and Q. Tian, "Understanding Blooming human groups in social networks," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 1980–1988, Nov. 2015.
- [35] H. W. Zhang et al., "Start from scratch: Towards automatically identifying, modeling, and naming visual attributes," *ACM Multimedia*, pp. 187–196, 2014.



DEWEN SENG was born in February 1977. He received the Ph.D. degree from the University of Science Technology, Beijing, China, in 2005. He is currently an Associate Professor and the Vice Director with the Software Engineering Institute, Hangzhou Dianzi University. He has (co)authored more than 40 papers. His research interests include artificial intelligence and data mining technology.



MENGFAN LI was born in March 1997. He is currently pursuing the master's degree with the School of Computer, Hangzhou Dianzi University. His current research interests include recommender systems and computer vision.



XUEFENG ZHANG was born in March 1980. He received the Ph.D. degree from Kyushu University, Fukuoka, Japan, in 2012. He is currently an Associate Professor with the College of Science and Technology, Ningbo University. He is also a Teacher with Hangzhou Dianzi University. His research interests include artificial intelligence and recommender systems.



JINGCHANG WANG was born in March 1999. He is currently pursuing the master's degree with the School of Computer, Hangzhou Dianzi University. His current research interests include recommender systems and attention mechanism.

...