Knowledge-Based Systems 251 (2022) 109185

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Content-aware Recommendation via Dynamic Heterogeneous Graph Convolutional Network

Tingting Liang [a], Lin Ma [b], Weizhong Zhang [c], Haoran Xu [a], Congying Xia [d], Yuyu Yin [a,*]

[a] College of Computer Science & Technology, Hangzhou Dianzi University, Hangzhou, China
[b] Meituan, Beijing, China
[c] The Hong Kong University of Science and Technology, Hong Kong
[d] Salesforce AI Research, San Francisco, CA, United States of America

ARTICLE INFO

Article history:
Received 27 November 2021
Received in revised form 12 May 2022
Accepted 30 May 2022
Available online 3 June 2022

Keywords:
Content-aware recommendation
Graph neural network
Dynamic heterogeneous graph
Content integration

ABSTRACT

With the explosive growth of products and multimedia contents on the Internet, the desire of users to find these online resources matching their interests makes it imperative to develop high quality recommendation systems. In this paper, we propose one novel neural network, namely Dynamic Heterogeneous Graph Convolutional Network (DHGCN), for item recommendation. Specifically, our proposed DHGCN consists of two components, namely the graph learner and heterogeneous graph convolution. The graph learner considers not only the user–item interactions but also user–user and item–item interactions, which are dynamically established during the graph evolution process. The heterogeneous graph convolution relies on a novel cross gating strategy to aggregate the representations yielded by the convolution over the learned heterogeneous graph and the item content information. Through comprehensive experiments on two real-world datasets, the proposed model is demonstrated to be effective on item recommendation task, outperforming the existing state-of-the-art models.

1. Introduction

With the big surge of Web applications and mobile devices, it becomes the norm for more and more people to consume plenty of online products and multimedia contents, such as movies and books. While this trend brings users multiple choices and wonderful consumption experience, it has also made the burden of users to find items matching their interests dramatically increase. To alleviate such an information overload problem, much research effort has been devoted to develop effective recommender systems. Conventional models for recommendation can be roughly categorized into three classes: collaborative filtering (CF) based methods [1,2], content-based methods [3–5], and the hybrid ones. The CF-based methods rely on the historical interactions to predict users' preference. Although CF has been widely accepted for its effectiveness and universality, it is weak in modeling side information (*e.g.*, user profiles, item attributes, context information) which is favorable in sparse scenarios where users and items have few interactions. To effectively exploit the side information, content-based models are extensively developed in recent years. A content-based model incorporating visual features is developed for the task of personalized ranking on implicit feedback datasets [6]. [5] presents a kernel to capture the contextual information in the news and then integrate the kernel into the recommendation framework. [7] extracts the textual and visual features from the user review texts and item images to learn the users' multimodal preferences for recommendation. The hybrid approaches unify the CF-based and content-based methods into one whole framework [8,9]. Although much progress has been achieved by leveraging content information in recommender systems, there is a lot of space for improvement due to the difficulties in content understanding and the user–item interaction modeling.

Recently, deep learning based models have been introduced for the recommendation task, due to their abilities on capturing the complicated nonlinear user–item relationships and learning more effective representations of both the users and items. Multilayer perceptron (MLP) [10,11] and auto-encoder [12,13] are incorporated into the recommendation systems to learn the complex interaction patterns undergoing user–item interactions. Convolutional neural networks (CNNs) are utilized for feature extraction from multiple sources [6,14]. In order to further model the temporal dynamics and sequential information, recommendation models realized in recurrent neural networks (RNNs) are considered [15,16]. Based on these powerful neural networks, content-aware recommender systems get more developed as content information can be modeled in a more effective way [17,18]. Although with significantly improved performances, these deep

* Corresponding author.
E-mail address: yinyuyu@hdu.edu.cn (Y. Yin).

https://doi.org/10.1016/j.knosys.2022.109185
0950-7051/

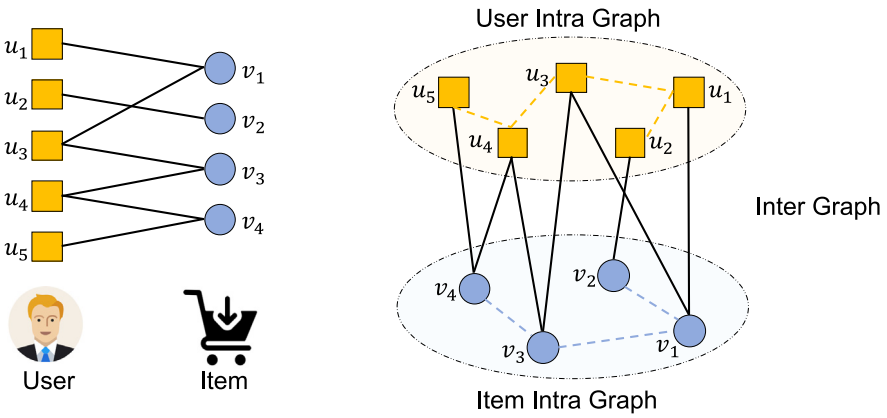

Fig. 1. (a) A traditional bipartite graph for modeling user–item interactions. (b) Our proposed dynamic heterogeneous graph for recommendation. Besides the traditional user–item interactions, which we name as the inter graph (the solid lines), the user–user and item–item interactions, which we regard as the intra graphs (the dashed lines), are also considered. Please note that the intra graph consists of two different connections, namely the static and dynamic connections, which will be dynamically established during the graph evolution process. Please check Section 3 for more detailed information.

learning based methods do not consider the local structure information of the users or items, which thereby neglect their fine-grained interactions. However, such interactions actually plays an essential role for the recommendation systems. Generally, the interactions between users and items in recommendation scenario can be viewed as a bipartite graph. For such a non-Euclidean data structure, classical CNN models are unable to be directly applied to learn effective patterns and extract meaningful information. Recent advances of graph neural network [19] gain extensive attention for their strength in the various fields [20–23], among which graph convolutional network (GCN) demonstrates its potential on the recommendation task [24,25], which aims to learn semantic representations of the users and items through the information propagation over local neighborhood structures and thereafter predict the interaction links.

As shown in Fig. 1(a), most existing GCN based recommendation models [24,25] mainly consider the user–item interactions, which we name as the inter graph, while neglect the user–user and item–item connections, which we regard as the intra graph. However, such connections within intra graphs are able to explicitly bridge the relationships in a collaborative manner, which is the essence of conventional CF-based methods. As shown in Fig. 1(a), user u_1 and u_4 are connected through four paths via the inter graph, which means the message passing between them needs at least 3-hop local convolutions. However, previous work shows that the best performance is obtained with a 2- or 3-hop convolutions for classification [26] and a 1-hop convolution for recommendation [24]. The reasons may be attributed to that the deeper graph convolution may introduce unexpected local noise, and increasing model depth may cause overfitting. The traditional GCN based recommendation models with limited local convolutions are unable to capture the global information, which contains the direct preference relevance between users and semantic similarity between items. Moreover, these models did not realize that the graph should evolve with the instantly learned node representations.

In order to handle the aforementioned shortcomings, we propose a dynamic heterogeneous graph convolutional network (DHGCN) for recommendation, which consists of two components, namely the graph learner and the heterogeneous graph convolution. Different from the traditional GCN only considering inter graph for recommendation, our proposed graph learner further incorporates the intra graph, which directly considers the user–user and item–item interactions. As shown in Fig. 1(b), with two intra graphs considered, u_1 and u_4 can be directly connected, which is conducive to directly obtain global information complementary to the local messages. More importantly, the intra graph dynamically varies during the graph evolution process. As such, the relationships between the users and items can be more comprehensively exploited. Our proposed heterogeneous graph convolution aggregates the latent representations yielded by convolutions over the dynamic heterogeneous graph by the graph learner. Additionally, the item content information is integrated with the item node representations learned from graph convolutions. Finally, the item recommender relies on the learned node latent representations to predict links between users and items. The main contributions are summarized as follows:

- We propose a novel network for implementing recommendation, namely the dynamic heterogeneous graph convolutional network (DHGCN), which consists of one graph learner to construct one dynamic heterogeneous graph and one heterogeneous graph convolution to perform the convolutions over the constructed graph and yield the corresponding semantic representations of the users and items.
- Besides the inter-graph, our graph learner constructs the intra graph to consider the user–user and item–item relationships, which is dynamically established during the heterogeneous graph evolution process. We propose one novel cross gating strategy to aggregate the latent representations yielded by the convolution over the learned heterogeneous graph and the item content information.
- Extensive experiments conducted on real-world datasets demonstrate that superiority of our proposed DHGCN, which outperforms the state-of-art models.

The paper is organized as follows: Section 2 provides preliminary concepts. Section 3 introduces the proposed model in detail. We show empirical evaluations in Section 4. Section 5 presents a review of the related works. Finally, conclusions are given in Section 6.

2. Preliminary

GCNs [24,25] have been recently employed to construct recommender systems. In this section, we briefly review GCN as well as its application on recommendation.

2.1. Graph Convolutional Network

GCN is introduced in [26] as an effective graph representation model that conducts convolutions over a graph structure. GCN

targets to learn and update the graph node representations by iteratively aggregating feature information of the nodes within local graph neighborhoods, which makes it capable of considering both the graph structure information and feature representation. Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N graph nodes $v_i \in \mathcal{V}$, edge $(v_i, v_j) \in \mathcal{E}$, and an adjacency matrix $A \in \mathbb{R}^{N\times N}$, the convolution operation of GCN performs as:

$$Z^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}Z^{(l)}W^{(l)}\right), \tag{1}$$

where $\tilde{A} = A + I_N$ denotes the adjacency matrix of the undirected graph $\mathcal{G}$ with added self-connections. I_N is an identity matrix. $\tilde{D}$ is a degree matrix with diagonal entries $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. And $W^{(l)}$ is a layer-specific matrix of convolutional filter parameters. $\sigma(\cdot)$ is a nonlinear activation function (e.g., Sigmoid or ReLU). With such graph convolution process, the node representation matrix $Z^{(l)} \in \mathbb{R}^{N\times D}$ of the lth layer is thereby updated as $Z^{(l+1)} \in \mathbb{R}^{N\times D}$. $Z^{(0)}$ can be initialized as X, which denotes the node feature matrix.

2.2. GCN based recommendation

The user–item interactions can be represented by a bipartite graph $\mathcal{G}_B = (\mathcal{U}, \mathcal{V}, \mathcal{E})$, where $\mathcal{U}$ and $\mathcal{V}$ are two disjoint vertex sets of users and items with $|\mathcal{U}| = N_u$ and $|\mathcal{V}| = N_v$ [27]. Each edge $(u, v) \in \mathcal{E}$ indicates that user u has one direction interaction with the item v. For instance, in the video recommendation scenario, the edge (u, v) means that user u watches/clicks/reposts the video v. Such user–item interactions can be represented as one feedback matrix $A_{UV} \in \mathbb{R}^{I_{N_u}\times I_{N_v}}$ with $A_{uv} = 1$ if edge (u, v) exists. As such, the adjacent matrix of the undirected bipartite graph $\mathcal{G}_B$ is obtained by:

$$A = \begin{bmatrix} \mathbf{0} & A^{UV} \\ A^{VU} & \mathbf{0} \end{bmatrix}, \tag{2}$$

where $A^{VU} = (A^{UV})^{\mathrm{T}}$. The one-layer graph convolutional encoder for recommendation performs as:

$$\begin{bmatrix} U \\ V \end{bmatrix} = \sigma\left(\begin{bmatrix} Z_u \\ Z_v \end{bmatrix} W\right), \tag{3}$$

with

$$\begin{bmatrix} Z_u \\ Z_v \end{bmatrix} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}XW^{(c)}\right), \quad \tilde{A} = A + I_{(N_u+N_v)}. \tag{4}$$

Here, U and V are matrices of user and item embeddings obtained during the graph convolution process, which can be thereafter used to predict the links between users and items. X is the input feature matrix of users and items. W_c is the graph convolutional filter parameters, while W denotes the parameter of the fully connected layer. Please note that multiple graph convolutional layers can be stacked together to more comprehensively exploit the interactions between users and items and finally yield the semantic representations of the users and items.

One of the weaknesses of the GCN based recommendation model is that each node is only allowed to receive the messages passed by its counterpart neighborhoods, which limits its ability of capturing global structures in the graph. Moreover, applying traditional GCN for recommendation cannot effectively integrate the item content, which is particularly important to content-based recommendation.

3. The proposed Dynamic Heterogeneous Graph Convolutional Neural Network

In order to handle the drawbacks of the traditional GCN, we propose one novel neural network, namely dynamic heterogeneous graph convolutional neural network (DHGCN), for content-aware recommendation. As shown in Fig. 2, our proposed DHGCN mainly consists of two components, specifically the graph learner and the heterogeneous graph convolution.

Besides the user–item interactions (inter graph) considered in traditional GCN models, our proposed graph learner is responsible for constructing a dynamic heterogeneous graph, which directly incorporates the user–user and item–item interactions (intra graphs). Specifically, the intra graph consists of two different connections, with one being the static connections and the other one being the dynamic connections. The static connections (dashed black lines) are established based on the fixed global characteristics that refer to the content information (*e.g.,* user profiles or item content) or the historical interactions (*i.e.,* users that consume an item or items that being consumed by a user). The dynamic connections (dashed red lines) are constructed by referring to the updated user or item representations and vary during the graph evolution process.

Based on the heterogeneous graph produced by the graph learner, we propose the heterogeneous graph convolution to update the user or item representation by aggregating feature information passed from local neighborhoods with convolutions over the heterogeneous graph. Moreover, in order to absorb the item content information, one novel cross gating strategy is proposed to further update the item representation.

3.1. Graph learner

We rely on a dynamic heterogeneous graph to tackle the recommendation task, which not only considers the user–item connections, but also the user–user and item–item connections. We define a heterogeneous graph $\mathcal{G} = (\mathcal{O}, \mathcal{E}, \mathcal{R})$, with the entities $\mathcal{O} = \mathcal{U} \cup \mathcal{V}$ consisting of N_u user nodes $u_i \in \mathcal{U}$ and N_v item nodes $v_j \in \mathcal{V}$. The edges within the heterogeneous graph is denoted by $(o_i, r, o_j) \in \mathcal{E}$, where r is the connection types between different entities, namely the inter connection, static intra connection, and dynamic intra connection.

Our proposed graph learner aims to construct different adjacent matrices based on the defined three types of relations between the entities (user and item nodes). For simplicity, we take the construction of user adjacent matrices as an example, while the item adjacent matrices can be constructed in the same way.

Inter Graph. The adjacent matrix A^{UV} of inter graph illustrates the implicit interactions between users and items, such as the user's clicking/viewing/reposting behaviors of the video, which is the same as the interactions defined in the traditional bipartite graph. A^{UV}_{ij} is set as 1, if the ith user interacts with jth item.

Static Intra Graph. Besides the user–item connections, the proposed graph learner also exploits the user–user and item–item interactions. We build a static intra graph based on the global characteristics of the users or items. For example, the user profiles and item content representations can be used to describe the global characteristics of the user and item, respectively. For users, we first compute the distance or similarity between any two users based on their profiles or historical interactions. There are lots of options for distance calculation and we choose the cosine similarity in this work for simplicity. Thus, $distance(\mathbf{x}_i^{(u)}, \mathbf{x}_j^{(u)}) = 1 - \frac{\mathbf{x}_i^{(u)}\cdot\mathbf{x}_j^{(u)}}{\|\mathbf{x}_i^{(u)}\|\|\mathbf{x}_j^{(u)}\|}$, where $\mathbf{x}_i^{(u)}$ denotes the profile or historical interactions of the ith user. Then we set a threshold t, if the distance between two users is less than the t, the two users are connected as follows:

$$(A_S^{UU})_{ij} = \begin{cases} 1, & distance(\mathbf{x}_i^{(u)}, \mathbf{x}_j^{(u)}) \le t, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

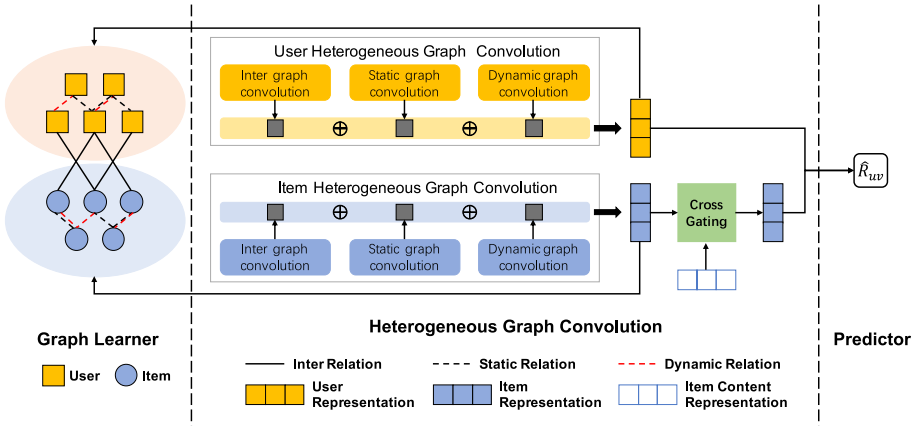

Fig. 2. The proposed DHGCN for content-aware recommendation. Our proposed graph learner aims to establish the connections between the user and item nodes. Specifically, three types of graphs are established, namely the inter graph, static intra graph, and dynamic intra graph. Our constructed heterogeneous graph is dynamically established during the graph evolution process. Afterwards, the heterogeneous graph convolution is performed over the constructed heterogeneous graph on the users and items, respectively. Moreover, a novel cross gating strategy is used to incorporate the item content representation for updating the item representation. During the inference stage, the yielded semantic representations of users and items are used to predict their links and appropriately recommend items to the related users.

Dynamic Intra Graph. More importantly, our proposed graph learner leverages the newly learned node representation to further construct a dynamic intra graph between users or items, which evolves with heterogeneous graph convolution process and is further reused for message passing to update the following node representations. Same as the static intra graph, the dynamic connections $(A_D^{UU})_{ij}$ is also determined by the distance $distance(\mathbf{u}_i, \mathbf{u}_j)$, where $\mathbf{u}_i$ denotes the currently learned node embedding for the ith user. Please note that during the graph evolution, each node representation will be updated, which thereby results in a dynamically varied intra graph.

For the users, our proposed graph learner will generate three different adjacent matrices, namely A^{UV}, A_S^{UU}, and A_D^{UU}. For the items, three different adjacent matrices A^{VU}, A_S^{VV}, and A_D^{VV} can be generated in the same manner.

3.2. Heterogeneous Graph Convolution

Based on the learned graphs (the adjacent matrices for users and items), we rely on the heterogeneous graph convolution process to yield the semantic embedding for each user and item node. Specifically, our proposed heterogeneous graph convolution consists of two steps: heterogeneous graph convolution encoding and item content integration.

3.2.1. Heterogeneous graph convolution encoder.

The core idea of conventional GCN based recommender systems [24,25] is to learn the representations of users and items via iteratively aggregating messages from local neighborhoods over a graph, which usually denotes the inter graph. However, for our proposed heterogeneous graph constructed by the graph learner, each node is connected with different types of neighbor nodes through different connections (inter or intra connections), and two nodes may be connected through different types of edges (static and dynamic connections). Therefore, each node can receive feature information and thereby perform updating through multiple types of local neighboring nodes with different type connections. As such, the proposed heterogeneous graph convolution encoder assigns different processing channels according to various relation types. Formally, the process of message passing for item nodes across the learned heterogeneous graph can be represented as follows:

$$\begin{aligned} Z_{v_{inter}}^{(l+1)} &= \hat{A}^{VU} U^{(l)} W_{vu}^{(l)}, \\ Z_{v_{static}}^{(l+1)} &= \hat{A}_S^{VV} V^{(l)} W_{vv}^{(l)}, \\ Z_{v_{dynamic}}^{(l+1)} &= \hat{A}_D^{VV} V^{(l)} \tilde{W}_{vv}^{(l)}, \end{aligned} \tag{6}$$

where $\hat{A}^{VU}, \hat{A}_S^{VV}, \hat{A}_D^{VV}$ are the normalized matrices [28]. $U^{(l)}$ and $V^{(l)}$ are the user and item representations learned in the lth graph convolution layer, and $\{W_{vu}, W_{vv}, \tilde{W}_{vv}\}$ are the matrices of convolution filter parameters for different relation types, namely inter, static intra, and dynamic intra connections respectively. $Z_{v_{inter}}^{(l+1)}, Z_{v_{static}}^{(l+1)}, Z_{v_{dynamic}}^{(l+1)}$ are convolved node representation over the inter graph, static intra graph, and dynamic intra graph, respectively, which are jointly considered to generate the item node representation at the $(l+1)$th graph convolution layer:

$$Z_v^{(l+1)} = \sigma\{W_v \sigma\{\phi(Z_{v_{inter}}^{(l+1)}, Z_{v_{static}}^{(l+1)}, Z_{v_{dynamic}}^{(l+1)})\}\}, \tag{7}$$

$\phi(\cdot)$ denotes an accumulation operation, such as concatenation and summation. $\sigma(\cdot)$ denotes the element-wise nonlinear activation function, such as ReLU.

Analogously, the representation $Z_u^{(l+1)}$ of the user nodes at the $(l+1)$th graph convolution layer can be also updated by a similar heterogeneous graph convolution encoder.

3.2.2. Item content integration

With the aforementioned heterogeneous graph convolution encoder, each item node receives the messages from its neighboring (or connected) user and item nodes, which does not consider the item content information. However, for recommendation task, the item content information plays an essential role, as it contains rich semantic information and directly attracts the users' interests. Therefore, in order to effectively incorporate the item content into the item node representation, we propose a novel cross gating mechanism to emphasize the useful information meanwhile gate out the irrelevant parts. As shown in Fig. 3, the cross gating mechanism is formulated as follows:

$$\begin{aligned} \mathbf{g}_v &= \sigma(W_f^g \mathbf{f}_j + b_f^g), \\ \mathbf{g}_f &= \sigma(W_v^g \mathbf{z}_{v,j} + b_v^g), \\ \tilde{\mathbf{z}}_{v,j} &= \mathbf{z}_{v,j} \odot \mathbf{g}_v, \\ \tilde{\mathbf{f}}_j &= \mathbf{f}_j \odot \mathbf{g}_f, \end{aligned} \tag{8}$$

where $\{W_v^g, W_f^g, b_v^g, b_f^g\}$ are the learnable parameters. $\mathbf{f}_j$ is the feature representation characterizing the content of jth item. $\mathbf{z}_{v,j}$

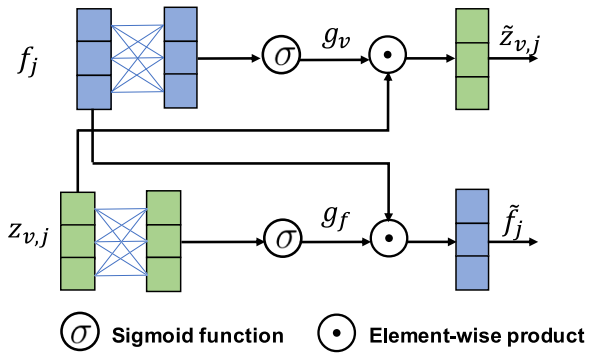

Fig. 3. The proposed cross gating mechanism, which adaptively incorporates the item content information to update the item representation.

is the representation of item j learned during the heterogeneous graph convolution encoder. $\sigma(\cdot)$ denotes the non-linear sigmoid activation function, and $\odot$ denotes the element-wise multiplication. The final representation of jth item in the $(l+1)$th layer is obtained by:

$$\mathbf{v}_j^{(l+1)} = \sigma(W^h \cdot [\tilde{\mathbf{z}}_{v,j}, \tilde{\mathbf{f}}_j]). \tag{9}$$

The latent representations of all items can be expressed by $V^{(l+1)} = [\mathbf{v}_1, \ldots, \mathbf{v}_{N_v}]^{\mathrm{T}}$, which can thereby be used to update the dynamic intra connections between items:

$$\hat{A}_D^{VV} \leftarrow distance(V^{(l+1)}, V^{(l+1)}). \tag{10}$$

Such constructed dynamic intra graph can further make the heterogeneous graph evolve by adaptively incorporating the item content information.

The cross gating mechanism aims to control the extent to which the item node representation interacts with the item content information. Specifically, if the item node representation is irrelevant to item content, the corresponding feature will be filtered to alleviate their effects on the subsequent processes. If the two are closely related, the cross gating strategy is expected to further emphasize their interactions.

3.3. Training

In order to train our proposed DHGCN, we use the widely applied Bayesian Personalized Ranking (BPR) loss [29], which is a pair-wise loss function. Compared to point-wise based loss, BPR compares the score of a positive and a sampled negative item and maximizes the preference difference between them. The loss function is defined as:

$$\mathcal{L} = \sum_{(i,j,k)\in\mathcal{D}} -\ln\sigma(\mathbf{u}_i^{\mathrm{T}}\mathbf{v}_j - \mathbf{u}_i^{\mathrm{T}}\mathbf{v}_k) + \lambda(\|U\|_2^2 + \|V\|_2^2), \tag{11}$$

where $\mathbf{u}_i$ and $\mathbf{v}_j$ denote the representations of ith user and jth item. λ represents the weight of regularization terms. The training set $\mathcal{D}$ is defined as follows:

$$\mathcal{D} = \{(i,j,k)|i \in \mathcal{U} \wedge j \in \mathcal{V}_i^+ \wedge k \in \mathcal{V}_i^-\}, \tag{12}$$

where $\mathcal{V}_i^+$ and $\mathcal{V}_i^-$ are the positive and negative item sets of user j.

3.4. Inference

With sufficient training, we leverages the learned semantic representations of users and items to measure their similarities and finally predict their links, based on which we can appropriately recommend items to the related users.

Table 1
Statistics of the evaluation datasets.

Dataset	Interactions	Items	Users	Sparsity
MovieLens-20M	330,231	8,587	7,061	99.46%
CiteULike	204,986	16,980	5,551	99.78%

4. Experiment

In this section, we evaluate the effectiveness of our proposed DHGCN for content-aware recommendations. We begin by describing the dataset used for the performance comparisons and the experimental settings, followed by a brief description of competitor models. Afterwards, we conduct the performance comparisons and also demonstrate the corresponding ablation studies to illustrate the effectiveness of our proposed DHGCN.

4.1. Dataset

We choose two publicly available datasets for evaluating content-aware recommendation which is the scenario studied in this work. Table 1 summaries the statistics of the evaluation datasets.

- **MovieLens-20M** [30]: The MovieLens datasets have been widely used to evaluate recommendation models. In this work, we use the MovieLens 20M dataset with a YouTube Trailers dataset[1] which links the videos on Youtube with the user-movie interaction data. As the feedbacks in this dataset is explicit, we transform them into implicit feedback following [10]. We select the recent two years data of MovieLens-20M and filter the users with interactions less than 10. For video content extraction, we first extract a sequence of images from each video at one image per second. Then, we use inception-V3 [31] to extract features from images in the sequence one by one. The dimension of the extracted features is 1024. We use the averaged feature as the final representation for the videos.
- **CiteULike**[2]: CiteULike is user for registered users to create scientific article libraries and save them for future reference. The goal is to leverage these libraries to recommend relevant new articles to each user. A subset of the CiteULike data with observed user-article pairs is used in our experiment. CiteULike contains article content information in the form of title and abstract. We use a vocabulary of the top 8,000 words selected by tf–idf [32].

4.2. Experimental setting

4.2.1. Evaluation protocols

To compare the performance of the proposed DHGCN with the above mentioned baselines models, we use Recall, Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG) as evaluation metrics. Recall measures the fraction of relevant items retrieved out of all relevant items, and MAP and NDCG also evaluate the ranking performance of models besides the accuracy. We report the average results for all users in the testing set. For both datasets, we select the first 80% items associated with each user in time order to form the training set and the latest 20% items as the testing set. In the training set, we use the latest 10% items as the validation set for model selection. We use the early stopping strategy based on the performance on validation set. Considering the randomness of deep learning model, we repeat each experiment 5 times and report the average performance.

[1] https://grouplens.org/datasets/movielens/20m-youtube/.
[2] http://www.citeulike.org.

Table 2
Performance comparison on MovieLens-20M.

Methods	R@10	R@20	R@40	M@10	M@20	M@40	N@10	N@20	N@40
VBPR	0.0667	0.1107	0.1767	0.0296	0.0325	0.0367	0.0597	0.0756	0.0994
FM	0.0667	0.1099	0.1788	0.0279	0.0309	0.0353	0.0587	0.0744	0.0988
CDML	0.0676	0.1112	0.1785	0.0286	0.0311	0.0355	0.0630	0.0785	0.1025
NCF	0.0788	0.1331	0.2171	0.0330	0.0367	0.0423	0.0720	0.0912	0.1207
SPCF	0.0543	0.1017	0.1750	0.0199	0.0238	0.0284	0.0457	0.0632	0.0882
GC–MC	0.0688	0.1168	0.1899	0.0312	0.0346	0.0394	0.0637	0.0813	0.1076
NGCF	0.0794	0.1320	0.2112	0.0351	0.0392	0.0447	0.0713	0.0908	0.1192
DHGCN	**0.0834**	**0.1427**	**0.2238**	**0.0369**	**0.0415**	**0.0471**	**0.0756**	**0.0974**	**0.1262**

4.2.2. Implementation details

We implement the DHGCN model with Tensorflow. We apply one-layer DHGCN model for a fair comparison with GC–MC. The dimension of all hidden layers are set as 16. We choose ReLU as the non-linear activation function. During the training process, Adam optimizer is adapted to minimize our BPR loss with a learning rate of 0.01. For the intra graph construction, we apply the cosine similarity to measure the connectivity.

4.3. Baseline methods

To evaluate the performance of the proposed approach, we compare it with the following representative recommendation models:

- **VBPR** [6] is a visual personalized ranking model that uncovers visual and latent dimensions simultaneously.
- **FM** [33] is a widely used content-aware recommendation method which explores pairwise interactions between item features.
- **CDML** [17] is an embedding learning approach that embeds videos using visual content onto a metric space, which can be used for video recommendation.
- **NCF** [10] is a deep learning based model combining MF and MLP to learn the latent representations from user–item interactions. The MLP endows NCF with the ability of modeling non-linear relationships between users and items.
- **NGCF** [34] is a graph neural network based framework which encodes the collaborative signal through high-order embedding propagation in the form of high-order connectivities through embedding propagation.
- **SpectralCF** [27] learns the deep connections between users and items with the rich information of connectivity in the spectral domains of the bipartite graph.
- **GC–MC** [24] proposes a graph auto-encoder framework to learn user–item interactions through message passing on the bipartite graph.

We select these seven baseline methods as: (1) VBPR, FM, and CDML are three typical content-based recommendation models, comparing to which can verify the strength of content modeling and integration of our proposed DHGCN. (2) As DHGCN is based on the graph convolutional network, for a fair comparison, we select three of the state-of-the-art models that successfully combine recommendation and graph neural network, namely NGCF, SpectralCF, and GCMC. (3) NCF is selected because it shows the impressive performance compared to those neural network based recommendation methods proposed in recent years. NCF is widely recognized effective and robust as it well utilizes the collaborative signal between users and items. Basically, most of the works of recommendation choose NCF as a baseline method.

For those methods that are originally designed for explicit recommendation, such as FM and GC–MC, for a fair comparison, we use the BPR loss to adapt them for implicit data.

4.4. Model comparison

Tables 2 and 3 compare DHGCN with baseline methods in terms of Recall@N, MAP@N, and NDCG@N ($N = 10, 20, 40$) based on two datasets. The best results are listed in bold. Overall, the proposed DHGCN consistently outperforms baseline methods across all cases. The content-based baselines (*i.e,* VBPR, FM, and CDML) obtain comparable performance, among which CDML performs slightly better, especially on CiteULike, as the embedding network is capable of learning item representations preserving item-to-item relationships. Among the graph-based baselines, SpectralCF gives the worst performance. One reason may be attributed to the lack of content information, another reason is that SpectralCF is mainly proposed for the cold-start problem and does not fit the dataset. GC–MC generally obtains a better result even compared to the content-based models as it benefits from the local message passing in the bipartite graph structure while content-based models do not consider the structure information. However, GC–MC is limited as each node in the graph only can learn information from its neighborhoods and misses the global characteristics and extra feature information. NGCF performs best compared to the other two graph-based models as it is capable of explicitly exploiting the collaborative signal and effectively exploring the high-order connectivity. NCF yields a performance comparable to NGCF based on the dataset of MovieLens-20M, benefiting from its ability of capturing non-linear user–item relationships. However, its performance on the dataset of CiteULike is not quite satisfactory, which indicates the unstable performance of NCF depends on datasets. However, none of above methods considers the heterogeneous graph structure involving both local and global structure information, and the feature transition through dynamic connections. As a result, DHGCN greatly outperforms the best baseline NCF by 7.21%, 13.08%, and 6.80% with metrics of Recall@N, MAP@N, and NDCG@N ($N = 20$) on MovieLens-20M. For CiteULike, compared to the best baseline NGCF, DHGCN obtains an improvement of 6.07%, 3.26%, and 5.93% in terms of Recall@N, MAP@N, and NDCG@N ($N = 20$).

4.5. Ablation study

Our DHGCN model contains three essential components including static intra graph, dynamic intra graph, and item feature integration. To verify the effectiveness of each component, we implement five variants of DHGCN based on the dataset of MovieLens-20M. Note that all these variants utilizes the user behaviors, namely the videos have been interacted by the users, to form the static connections between users.

- **Static-F**: This variant removes the dynamic intra graph and item feature integration of our DHGCN model, and uses item features to build the static intra graph for items.
- **Static-F(CG)**: This variant adds the cross gating mechanism into Static-F to integrate representations learned from heterogeneous graph convolution with item features.

Table 3
Performance comparison on CiteULike.

Methods	R@10	R@20	R@40	M@10	M@20	M@40	N@10	N@20	N@40
VBPR	0.0388	0.0676	0.1155	0.0127	0.0146	0.0166	0.0248	0.0339	0.0471
FM	0.0352	0.0634	0.1061	0.0114	0.0132	0.0151	0.0224	0.0312	0.0432
CDML	0.0743	0.1155	0.1799	0.0275	0.0309	0.0341	0.0469	0.0600	0.0773
NCF	0.0666	0.1105	0.1704	0.0257	0.0294	0.0336	0.0480	0.0621	0.0799
SPCF	0.0877	0.1387	0.2095	0.0355	0.0411	0.0455	0.0629	0.0789	0.0986
GC–MC	0.1099	0.1620	0.2350	0.0460	0.0524	0.0574	0.0790	0.0957	0.1161
NGCF	0.1177	0.1799	0.2577	0.0473	0.0539	0.0587	0.0818	0.1011	0.1227
DHGCN	**0.1212**	**0.1908**	**0.2797**	**0.0480**	**0.0557**	**0.0616**	**0.0852**	**0.1071**	**0.1321**

Table 4
Differences among the variant models of DHGCN.

Methods	Item static graph		Feature integration	Dynamic graph
	Item feature	Historical interaction		
Static-F	✓			
Static-F(CG)	✓		✓	
DHGCN-F	✓		✓	✓
Static-B		✓		
Static-B(CG)		✓	✓	
DHGCN		✓	✓	✓

Table 5
Performance comparison with five variants.

Methods	R@10	R@20	R@40	M@10	M@20	M@40	N@10	N@20	N@40
GC–MC	0.0688	0.1168	0.1899	0.0312	0.0346	0.0394	0.0637	0.0813	0.1076
Static-F	0.0832	0.1376	0.2181	0.0360	0.0400	0.0455	0.0757	0.0955	0.1241
Static-F(CG)	**0.0843**	0.1374	0.2209	0.0360	0.0400	0.0458	0.0749	0.0943	0.1240
DHGCN-F	0.0817	0.1341	0.2171	0.0338	0.0377	0.0433	0.0721	0.0912	0.1206
Static-B	0.0748	0.1274	0.2058	0.0318	0.0359	0.0413	0.0667	0.0860	0.1139
Static-B(CG)	0.0841	0.1406	0.2209	0.0357	0.0401	0.0456	0.0749	0.0957	0.1243
DHGCN	0.0834	**0.1427**	**0.2238**	**0.0369**	**0.0415**	**0.0471**	**0.0756**	**0.0974**	**0.1262**

- **DHGCN-F**: This variant adds the dynamic intra graph into Static-F(CG), which only has difference in the construction of item static graph compared to our DHGCN model.
- **Static-B**: This variant only differs from Static-F in the way to construct the static intra graph. It applies the historical rated records to build the graph.
- **Static-B(CG)**: The difference between Static-B(CG) and Static-F(CG) are the same with that between Static-F and Static-B.
- **DHGCN**: This is our DHGCN model, which adds a dynamic intra graph based on Static-B(CG).

Table 4 summarizes the differences among these variant models for a better understanding.

Results and Analysis: Table 5 shows the comparison performance of DHGCN and its five variants. Overall, all the components make significant contributions compared to the original GCN based model (GC–MC), which demonstrates the necessity the user–user and item–item connections as the only consideration of interactions between users and items would result in a loss of the global information. All these variants can be divided into two classes according to the information used for constructing the item static intra graph: the features extracted from item contents (the first three variants in Table 5) and the rated records of items (the last three variants).

For the former class, it can be observed that constructing item static intra graph with content information (Static-F) brings significant improvement. The items with similar features are connected and pass messages from each other, which means the updated representations of those items with similar features are similar as well. On this basis, integrating the similar item features would be much less impressive or even counterproductive. The fact that Static-F(CG) makes little improvement to Static-F proves this point. The extra dynamic graph also fails to make a progress as the construction of the dynamic intra graph over the items is based on the item representations learned in the last iteration. Since the representations of items with similar features are similar, the static intra graph and dynamic intra graph are constructed by the same feature information. DHGCN-F overlays the same item feature three times, which leads the worse performance as there are no complementary features only redundancy.

For the latter class, Static-B uses the item historical interactions, also known as the user behaviors on items to construct the static intra graph of items. There are message passing between the items with the similar interactions to generate the new representations. Static-B performs worse than Static-F, indicating that item features contains more representative information. On this basis, integrating the different kind of item features via the cross gating mechanism leads to a significant improvement. The obvious evidence is that Static-B(CG) gets at most 12.43%, 12.23%, and 12.29% promotion in three metrics compared to Static-B. The dynamic graph makes the further improvements as it incorporates the structure feature generated by the last updated representations which is the vector-level feature of historical interactions. The multiple types of global information (historical interactions and item features) makes our DHGCN achieve the best performance.

4.6. Dynamic network visualization

To further understand the proposed DHGCN model, we perform visualized analysis based on MovieLens-20M. Fig. 4 shows the static intra graph and the evolution of dynamic intra graph of videos. For clarity, we select the first 50 videos to show the connections (similarities) between any two of them. The higher the similarity, the redder the color, while the lower the similarity,

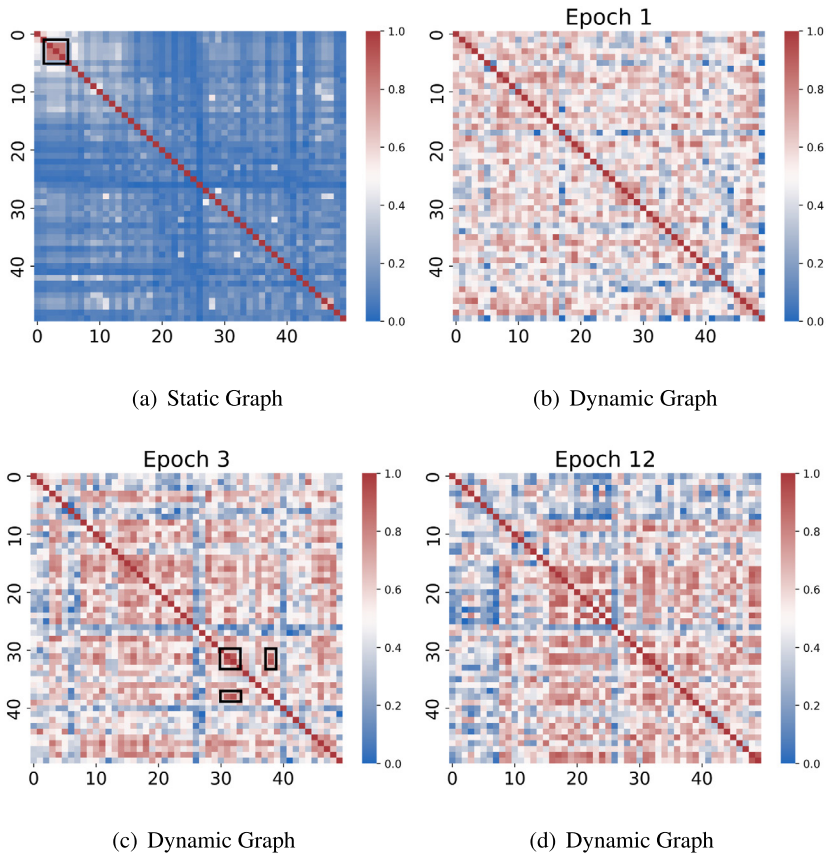

Fig. 4. Visualization of the evolution of the dynamic intra graph of videos. The higher the similarity, the redder the color, while the lower the similarity, the bluer the color.

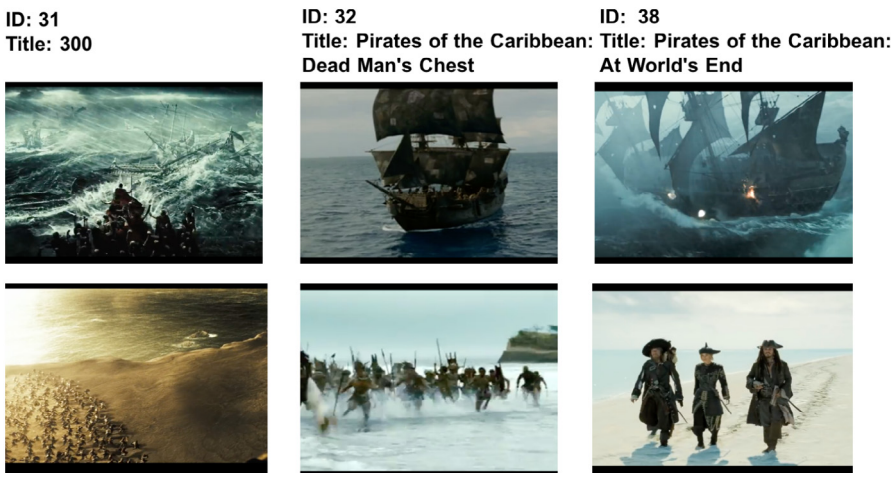

Fig. 5. Visualization of the similar cases in the dynamic intra graph.

the bluer the color. Fig. 4(a) is the static intra graph which is computed by the cosine similarity of the user vectors associated with videos. It can be seen that most nodes hold a weak connections with each other except for the three videos in the black box at the top left, which are the series of *Lord of the Rings*. The high similarities among the three videos indicates that the audiences for them are quite similar, which is consistent with the reality. Fig. 4(b–d) are the dynamic graphs obtained in epoch 1, 3, and 12. In the first epoch, the similarity values between any two videos (except for the similarity to itself) are relatively uniform. Most of the values are between 0.4 and 0.6 as the related colors are lighter. Along with the training process, the similarity values in epoch 3 appear polarization as the reds and blues cover more, which means the relationships between these videos become differentiable. This phenomenon is even more pronounced in epoch 12. The connections between a part of nodes get weaker while some connections become stronger, indicating the evolution of the intra graph is able to filter some irrelevant information and emphasis the valuable information.

The qualitative example in Fig. 5 can further verify the effectiveness of dynamic modeling with the frames in video trailers. The three movies corresponding to the videos in Fig. 4(c) with ids 31, 32, and 38 (the red points in the black box), and each two of them are connected with a high similarity. As shown in Fig. 5,

the NO. 32 and No. 38 movies belong to the series of *Pirates of the Caribbean* which are surely similar with each other. The trailer of No. 31 contains some similar frames with the other two, which demonstrates that the dynamic network has the ability to learn a precise node relationships during the training process.

5. Related work

From the conceptual perspective, two topics can be seen as closely related to our work: graph neural networks and deep learning based recommender systems.

5.1. Deep learning based recommender systems

Due to the capability to capture the nonlinear user–item relationships and learn representative user and item embeddings, deep learning techniques have been widely introduced for the recommendation tasks. Some works employ CNNs to extract features from multiple sources to facilitate recommendation [6,14]. A convolutional sequence embedding recommendation model called Caser [35] adopts both horizontal and vertical convolutional filters to capture sequential patterns at multiple levels including point-level and union-level, and general preferences including skip behaviors and long term user preferences. MLP [10] and auto-encoder [12,13] are incorporated into the recommendation systems to learn the complex interaction patterns undergoing user–item interactions. A MLP-based method is proposed for makeup recommendation [36], which utilizes two identical MLPs to model labeled examples and expert rules to guide the learning process of recommendation. [37] proposes an autoencoder framework with an attention mechanism for cross-domain recommendation, which is able to transfer information between different domains and integrate features for a more accurate recommendation. For session-based recommendation, RNNs are frequently used to model the temporal dynamics and sequential information [15,38]. A long- and short-term preference modeling framework is proposed for next-POI recommendation [39], consisting of a nonlocal network for modeling long-term preference and a geo-dilated RNN for learning short-term preference. [40] proposes to recommend items through the integration both of historical preferences and present user motivations with two components, namely Neural Item Embedding, and Discriminative Behaviors Learning. The latter develop a contextual LSTM to effectively learn the session behaviors and preference behaviors for improving the next-item recommendation.

Recently, in addition to the classical neural networks (*e.g.*, CNNs, RNNs, MLP) based recommendation models, GNN based recommendation methods begin to be developed, which would be specifically introduced in the next part.

5.2. Graph neural networks

Graph structure is widely adopt for representing complicated relationships of objects, *e.g.,* social networks, knowledge graph, and traffic connection. With the rapid development of deep learning techniques, many works try to build model with graph data by neural network, which explicitly facilitates the rise of Graph Neural Network (GNN). The key points of GNN are the iterative aggregation of features from neighboring nodes and the integration of the aggregated messages with the present central node features during the propagation process [19]. In recent years, plenty of graph neural network for learning over graphs have been proposed. The original GCN model proposed in [26] uses a layer-wise propagation rule that is based on a first-order approximation of spectral convolutions on graphs for semi-supervised classification. An inductive framework GraphSAGE is proposed to employ node feature information with different aggregator functions to learn an embedding function that generalizes to unseen nodes [41]. Based on the pure GCN model, [42] introduces an attention-based framework which assigns different importance to nodes of a same neighborhood to perform node classification of graph-structured data. GGNN [43] is a typical recurrent graph neural network which learns high-level node representations by adopting a gated recurrent unit in the node representation update step.

Considering the fact that most of the data in recommendation scenario has essentially a graph structure, some works start to apply the powerful GNN techniques for recommendation. [24] models the matrix completion for recommender systems as link prediction in bipartite graphs and proposes a graph convolutional encoder/decoder to predicts unseen ratings. A highly-scalable GCN based approach combining random walks and graph convolutions to generate node embeddings is proposed to improve the efficiency and robustness of recommender systems [25]. NGCF [34] is proposed to encode the collaborative signal through high-order embedding propagation in the form of high-order connectivities through embedding propagation. Considering that the interacted items are not equally representative to reflect user preferences, the GAT framework based MCCF [44] employs attention mechanism to differentiate the importance of neighbors. Compared to these GNN based recommendation models that only consider one type of connections (*i.e.*, user–item interactions), our model constructs two more types of connections (*i.e.*, user–user and item–item relationships) which are dynamically established during the graph evolution process.

6. Conclusion

In this paper, we proposed a novel neural network for content-based recommendation, namely the dynamic heterogeneous graph convolutional neural network (DHGCN). DHGCN consists of one graph learner to build both static and dynamic graphs for message passing, and one heterogeneous graph convolution to learn the representation of each node from its neighborhoods. A cross gating mechanism is designed for the feature integration in heterogeneous graph convolutions. Our experiments on two real-world datasets provide strong evidence for the superiority of the proposed DHGCN.

Promising as DHGCN might be in content and structure feature modeling, there still exists room for improvements. In the future work, we will pay more attention to the data sparsity and cold-start problems. Setting the appropriate recommendation scenario and designing the specific feature extraction strategy for new users or new items is a main idea. In the feature extraction part, especial for modeling user preference, we will use the datasets with timestamp information to model the temporal dynamics for a better recommendation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported in part by “Pioneer” and “Leading Goose” R&D Program of Zhejiang, China under grant 2022C03043, Natural Science Foundation of Zhejiang Province, China under grant LY22F020009, NSFC, China under grant 62002088, 61872119.

References

[1] J.B. Schafer, D. Frankowski, J. Herlocker, S. Sen, Collaborative filtering recommender systems, in: The Adaptive Web, Springer, 2007, pp. 291–324.

[2] Y. Huang, B. Cui, J. Jiang, K. Hong, W. Zhang, Y. Xie, Real-time video recommendation exploration, in: Proceedings of the 2016 International Conference on Management of Data, ACM, 2016, pp. 35–46.

[3] T. Mei, B. Yang, X.-S. Hua, S. Li, Contextual video recommendation by multimodal relevance and user feedback, ACM Trans. Inf. Syst. (TOIS) 29 (2) (2011) 10.

[4] P. Cui, Z. Wang, Z. Su, What videos are similar with you?: Learning a common attributed representation for video recommendation, in: Proceedings of the 22nd ACM International Conference on Multimedia, ACM, 2014, pp. 597–606.

[5] Z. Lu, Z. Dou, J. Lian, X. Xie, Q. Yang, Content-based collaborative filtering for news topic recommendation, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 217–223.

[6] R. He, J. McAuley, VBPR: visual Bayesian personalized ranking from implicit feedback, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI Press, 2016, pp. 144–150.

[7] C. Xu, Z. Guan, W. Zhao, Q. Wu, M. Yan, L. Chen, Q. Miao, Recommendation by users' multimodal preferences for smart city applications, IEEE Trans. Ind. Inf. 17 (6) (2020) 4197–4205.

[8] A. Ferracani, D. Pezzatini, M. Bertini, A. Del Bimbo, Item-based video recommendation: An hybrid approach considering human factors, in: Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, ACM, 2016, pp. 351–354.

[9] B. Chen, J. Wang, Q. Huang, T. Mei, Personalized video recommendation through tripartite graph propagation, in: Proceedings of the 20th ACM International Conference on Multimedia, ACM, 2012, pp. 1133–1136.

[10] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.

[11] H. Guo, R. Tang, Y. Ye, Z. Li, X. He, Deepfm: a factorization-machine based neural network for ctr prediction, 2017, arXiv preprint arXiv:1703.04247.

[12] H. Wang, N. Wang, D.-Y. Yeung, Collaborative deep learning for recommender systems, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1235–1244.

[13] S. Sedhain, A.K. Menon, S. Sanner, L. Xie, Autorec: Autoencoders meet collaborative filtering, in: Proceedings of the 24th International Conference on World Wide Web, ACM, 2015, pp. 111–112.

[14] S. Wang, Y. Wang, J. Tang, K. Shu, S. Ranganath, H. Liu, What your images reveal: Exploiting visual contents for point-of-interest recommendation, in: Proceedings of the 26th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2017, pp. 391–400.

[15] B. Hidasi, A. Karatzoglou, L. Baltrunas, D. Tikk, Session-based recommendations with recurrent neural networks, 2015, arXiv preprint arXiv:1511.06939.

[16] C.-Y. Wu, A. Ahmed, A. Beutel, A.J. Smola, H. Jing, Recurrent recommender networks, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, ACM, 2017, pp. 495–503.

[17] J. Lee, S. Abu-El-Haija, B. Varadarajan, A.P. Natsev, Collaborative deep metric learning for video understanding, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2018, pp. 481–490.

[18] C. Ma, P. Kang, B. Wu, Q. Wang, X. Liu, Gated attentive-autoencoder for content-aware recommendation, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019, pp. 519–527.

[19] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S.Y. Philip, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. 32 (1) (2020) 4–24.

[20] X. Song, J. Li, Y. Tang, T. Zhao, Y. Chen, Z. Guan, JKT: A joint graph convolutional network based deep knowledge tracing, Inform. Sci. 580 (2021) 510–523.

[21] S. Min, Z. Gao, J. Peng, L. Wang, K. Qin, B. Fang, STGSN—A spatial–temporal graph neural network framework for time-evolving social networks, Knowl.-Based Syst. 214 (2021) 106746.

[22] X. Song, J. Li, Q. Lei, W. Zhao, Y. Chen, A. Mian, Bi-CLKT: Bi-graph contrastive learning based knowledge tracing, Knowl.-Based Syst. 241 (2022) 108274.

[23] D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, B. Xu, J. Bai, J. Tong, et al., Spectral temporal graph neural network for multivariate time-series forecasting, Adv. Neural Inf. Process. Syst. 33 (2020) 17766–17778.

[24] R.v.d. Berg, T.N. Kipf, M. Welling, Graph convolutional matrix completion, Stat 1050 (2017) 7.

[25] R. Ying, R. He, K. Chen, P. Eksombatchai, W.L. Hamilton, J. Leskovec, Graph convolutional neural networks for web-scale recommender systems, 2018, arXiv preprint arXiv:1806.01973.

[26] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.

[27] L. Zheng, C.-T. Lu, F. Jiang, J. Zhang, P.S. Yu, Spectral collaborative filtering, in: Proceedings of the 12th ACM Conference on Recommender Systems, 2018, pp. 311–319.

[28] M. Zitnik, M. Agrawal, J. Leskovec, Modeling polypharmacy side effects with graph convolutional networks, Bioinformatics 34 (13) (2018) i457–i466.

[29] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, in: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2009, pp. 452–461.

[30] F.M. Harper, J.A. Konstan, The movielens datasets: History and context, Acm Trans. Interact. Intell. Syst. (Tiis) 5 (4) (2016) 19.

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.

[32] C. Wang, D.M. Blei, Collaborative topic modeling for recommending scientific articles, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2011, pp. 448–456.

[33] S. Rendle, Factorization machines, in: 2010 IEEE International Conference on Data Mining, IEEE, 2010, pp. 995–1000.

[34] X. Wang, X. He, M. Wang, F. Feng, T.-S. Chua, Neural graph collaborative filtering, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 165–174.

[35] J. Tang, K. Wang, Personalized top-n sequential recommendation via convolutional sequence embedding, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, pp. 565–573.

[36] T. Alashkar, S. Jiang, S. Wang, Y. Fu, Examples-rules guided deep neural network for makeup recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, no. 1, 2017.

[37] S.-T. Zhong, L. Huang, C.-D. Wang, J.-H. Lai, S.Y. Philip, An autoencoder framework with attention mechanism for cross-domain recommendation, IEEE Trans. Cybern. (2020).

[38] C. Wu, J. Wang, J. Liu, W. Liu, Recurrent neural network based recommendation for time heterogeneous feedback, Knowl.-Based Syst. 109 (2016) 90–103.

[39] K. Sun, T. Qian, T. Chen, Y. Liang, Q.V.H. Nguyen, H. Yin, Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, no. 01, 2020, pp. 214–221.

[40] Z. Li, H. Zhao, Q. Liu, Z. Huang, T. Mei, E. Chen, Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1734–1743.

[41] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Advances in Neural Information Processing Systems, 2017, pp. 1024–1034.

[42] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: Proceedings the International Conference on Learning Representations, 2018.

[43] Y. Li, D. Tarlow, M. Brockschmidt, R. Zemel, Gated graph sequence neural networks, 2015, arXiv preprint arXiv:1511.05493.

[44] X. Wang, R. Wang, C. Shi, G. Song, Q. Li, Multi-component graph convolutional collaborative filtering, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, no. 04, 2020, pp. 6267–6274.