Computers and Education: Artificial Intelligence 2 (2021) 100010

Contents lists available at ScienceDirect

Computers and Education: Artificial Intelligence

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

Top-N personalized recommendation with graph neural networks in MOOCs

Jingjing Wang [a], Haoran Xie [b], Fu Lee Wang [a,*], Lap-Kei Lee [a], Oliver Tat Sheung Au [a]

[a] School of Science and Technology, The Open University of Hong Kong, Ho Man Tin, Hong Kong
[b] Department of Computing and Decision Sciences, Lingnan University, Tuen Mun, Hong Kong

ARTICLE INFO

Keywords:
Recommender Systems
MOOCs
Graph neural networks
Personalized learning

ABSTRACT

Top-N personalized recommendation has been extensively studied in assisting learners in finding interesting courses in MOOCs. Although existing Top-N personalized recommendation methods have achieved comparable performance, these models have two major shortcomings. First, these models seldom learn an explicit representation of the structural relation of items. Second, most of these models typically obtain a user's general preference and neglect the recency of items. This paper proposes a Top-N personalized Recommendation with Graph Neural Network (TP-GNN) in the Massive Open Online Course (MOOCs) as a solution to tackle this problem. We explore two different aggregate functions to deal with the user's sequence neighbors and then use an attention mechanism to generate the final item representations. The experiments on a real-world course dataset demonstrated that TP-GNN could improve the performances. Furthermore, the system developed based on our method obtains positive feedback from the participants, which denotes that our method effectively predicts learners' preferences and needs.

1. Introduction

The emergence and rapid development of Internet and education technology have made online learning increasingly popular. Millions of users register on Massive Open Online Courses (MOOCs) annually, such as Coursera, edX and Udacity (Gupta, Jain, Goyal, & Lalit, 2020). As such, a vast number of courses have been accumulated on these MOOC platforms, leading to information overload. Technology-supported learning, such as Recommendation system (RS), has attracted much interest in the personalized learning area over the past decades. RS is a powerful tool to alleviate information overload and advertise potential items to users. Two widely used Top-N recommendation algorithms are the content-based RS (Pazzani & Billsus, 2007, pp. 325–341) and the collaborative filtering RS (He et al., 2017; Schafer, Frankowski, Herlocker, & Sen, 2007, pp. 291–324). These algorithms capture learners' preferences and provided personalized learning guidance based on the historical interactions like enrolls and views, to improve the learning performance of learners. However, most of these approaches focus on the user's general preferences, while ignoring their recent preferences. General preferences represent a user's long-term and static behaviors, while recent preferences present a user's short-term and dynamic behaviors. For example, in an online learning scenario, a learner might want to start a career in database operation technology. In this instance, he or she might go online to search for SQL database operation and then eventually turn to computer language programming. This is where conventional RS models would fail to adapt to the dynamic change of user preferences, thus negative recommendations would be generated by employing these models (Qiu, Li, Huang, & Yin, 2019). Because in recent preferences, the next item or action is more dependent on the user's most recent items or actions.

Using sequence of session events can solve the shift of a user's preferences and obtain promising results in many domains (Figueiredo, Ribeiro, Almeida, & Faloutsos, 2016; Hidasi, Karatzoglou, Baltrunas, & Tikk, 2015). Many types of proposals for sequential recommendations have been developed. Most of the existing research (Hidasi et al., 2015; Li et al., 2017; Tan, Xu, & Liu, 2016; Tuan & Phuong, 2017) employ Recurrent Neural Networks (RNNs) for establishing session-based RS. Hidasi et al. (2015) proposes to utilize Gated Recurrent Units (GRU) based on RNN to improve the performances of the session-based recommendations, known as GRU4Rec. Furthermore, they provide a data augmentation technique to ensure adequate user behavior training data. Recently, NARM (Li et al., 2017) developed an encoder-decoder model based RNN to simultaneously obtain users' sequential behaviors and general preferences. STAMP (Liu, Zeng, Mokhosi, & Zhang, 2018) also

[*] Corresponding author.
E-mail addresses: wangj3573@gmail.com (J. Wang), hrxie2@gmail.com (H. Xie), pwang@ouhk.edu.hk (F.L. Wang), lklee@ouhk.edu.hk (L.-K. Lee), oau@ouhk.edu.hk (O.T.S. Au).

https://doi.org/10.1016/j.caeai.2021.100010
Received 30 September 2020; Received in revised form 19 January 2021; Accepted 19 January 2021

aims to capture users' long-term and short-term interests. They propose to explicitly model the user's current interests based on last clicks into the recommendation.

These state-of-the-art methods obtain satisfactory results, however there are two major shortcomings with existing RNN based methods. First, these models rely on adequate user behavior data to estimate user representations. User behavior data that is implicated on session clicks is usually limited, especially for new enrollments via clicks in MOOCs. Due to session being anonymous, in session-based recommendation systems, item representations are generated from the hidden state of these RNN methods, (e.g. the global recommender of NARM). Therefore, it is difficult to generate user's representations from each session. Furthermore, methods based on RNN may perform poorly, when adjacent actions do not have dependency relationships, for example, accidental clicks on wrong items, or those enrolling on the course out of curiosity.

Second, the current recommendation literature typically considers only direct connectivity relations, which limiting the ability of representation, leading to poor recommend quality and a significant decay in performance. Since users often share information among friends, implying that users' preferences are easily influenced by their friends. As such, it is worth taking user-item interactions in the high-order connections into account (Wang, He, Wang, Feng, & Chua, 2019).

To tackle the above issues, we propose a novel graph neural network framework, namely a Top-N personalized recommendation with a Graph Neural Network in MOOCs (TP-GNN). To make the most of co-learning relationship and sequential patterns, we explore two different aggregate functions to deal with users' neighbors based on a user-item interaction graph, and then apply an attention mechanism to generate the final node embedding representations. The main contributions of this work are as follows:

- TP-GNN model both the user's sequential behaviors and general preferences into consideration.
- For general preferences, we apply GCN to capture the high-level implicit relations between nodes, particularly, for course category relationship and co-learning relationship. For serialization aggregation, we apply CNN to filter significant features of sequential patterns.
- We carried out these experiments on a real-world MOOC dataset. The results show that TP-GNN outperforms baselines in terms of Recall and NDCG. we also develop the learning system based our method to evaluate the effectiveness.

2. Related work

2.1. Conventional recommendation

The core idea of conventional recommendations is based on the similarities between items and users obtained from the user-item interaction data (Wang, Xiong, Huang, & Li, 2018). One of the main approaches is Matrix Factorization (MF) (Koren, Bell, & Volinsky, 2009), which is centered on reducing the problems from data sparsity. MF methods decompose the user-item rating matrix into two low-ranking matrices for identifying latent features (Jishan & Wang, 2017). Singular Value Decomposition (SVD) is one of the most widely MF-based models employed in RS (Luo, Xie, Rao, & Wang, 2019). SVD methods achieved comparable results but the complexity of calculation limited the development. Koren proposed SVD++ (Koren, 2008), compared to SVD, this method incorporates implicit feedback information to achieve a better accuracy. Factorization machines (Rendle, 2010) provide a mechanism to model the pairwise features interactions as the addition and inner product operations to obtain a promising result. MF-based methods focus on inner-product mechanism, which impose fundamental limitations on item embeddings and lead to a poor result.

Recently, a powerful approach in recommendation systems is incorporated neural networks into collaborative filtering methods (He et al., 2017). Salakhutdinov, Mnih, and Hinton (2007) are the first to use neural network for Collaborative Filtering (CF) in movie recommendation. In this study (Salakhutdinov et al., 2017), Restricted Boltzmann Machines (RBM) is adopted to model user-item interactions for recommendations. Other work (Wang, Wang, & Yeung, 2015) introduces a hierarchical Bayesian model based on encoder-decoder mechanism, this method can jointly leverage content information and collaborative filtering for the next basket recommendation. Furthermore, some works such as models proposed by Seo (Seo, Huang, Yang, & Liu, 2017) and TARMF (Lu, Dong, & Smyth, 2018) integrates attention mechanism into deep neural network to weigh the contributions of items. Even in cross-domain recommendations (Liu, Li, Du, Chang, & Gao, 2019), deep neural networks also show its powerful ability to map items into a joint latent space.

Another line of these works aims to solve sequence recommendations. Recurrent neural networks (RNN) are commonly used for session-based recommendations, which can deal with variable-length sequence data. Hidasi et al. (2015) was the first use RNN to recommendation and achieves significant improvements over traditional methods, even on a short session-based dataset. Tan et al. (2016) proposes to incorporate two techniques to further improve the accuracy of recommendations for session-based recommendation, one is data augmentation to reduce overfitting and the other is to adjust the distribution of the input data through pre-training. Zhang et al. (2014) employ RNN for predicting advertisement click, and they point out that on ad click prediction domain, users' next click highly depends on the current behaviors on sessions. Hidasi et al. (2015) incorporates rich item features such as image and text to further improve the performance of RNN based session recommendations. Instead of single RNN, they propose using several parallel RNNs to model different nature of the input data respectively. Some works have focused on using convolution neural networks (CNN) for recommendations. Kim, Park, Oh, Lee, and Yu (2016) proposed incorporating auxiliary information from the textual data like reviews and opinions based on CNN, to improve the accuracy of rating predictions. Tang and Wang (2018) proposed a convolutional sequence embedding method to consider the recent items as an 'image', as well as learning features using convolution filters.

2.2. Graph-based recommendation

Although all deep learning models show promising improvements over traditional algorithms, these methods have difficulty dealing with graph datasets and cannot extract the rich semantic information present in graph structures. A considerable amount of research effort on graph neural network (GNNs) (Fan, Zhu, et al., 2019; Li & Chen, 2013; Song et al., 2019; Zhang, Song, Huang, Swami, & Chawla, 2019) models have emerged to addressing the problem. The key idea is how to recursively propagates the node's embedding representations based on the local neighbors. Meanwhile, node and side information can be aggregated to augment the graph-structured data. There has been a body of work based on spectral graph theory (Monti, Bronstein, & Bresson, 2017; Zheng, Lu, Jiang, Zhang, & Yu, 2018). Although these methods are applied as a powerful tool in node classification, they generate the node representation for the whole graph, which is computationally expensive for large graphs and require large memory to keep the value of eigenvectors of the Laplacian in the graph.

sRMGCNN (Monti et al., 2017) applied GNNs to extract the pairwise relationships from the interactions in the graph structure data, then perform a diffusion process based on recurrent neural network. Graph Convolutional Matrix Completion (GCMC) (Berg, Kipf, & Welling, 2017) employs a graph auto-encoder framework to predict the link on a bipartite user-item graph through a form of differentiable message passing. PinSage (Ying et al., 2018) applies a graph neural network with random walk to obtain the vector representations of items on the homogeneous graph. Notably, PinSage has shown significant performance in web scale graphs. GNNs have also been employed on generating node feature representations in various domains, for example, social recommendations (Fan, Ma, et al., 2019; Wang, Tan, & Zhang, 2010) and news recommendations (Wang, Zhang, Xie, & Guo, 2018). In addition, Fan, et al. (2019) have proposed an approach to learn latent factors of items and users by leveraging both user-item interactions and opinions.

2.3. Personalized learning systems

Personalized learning systems aim to provide learning materials to cater various learning interests, needs and strengths from individuals (Xie, Chu, Hwang, & Wang, 2019). Several personalized learning recommendation systems were developed. Chen, Hsu, Li and Peng (2006) have developed a mobile learning system, which suggests reading materials by taking individual reading ability into account. Hsu (2008) has developed an English learning system, which recommends courses according to reading patterns. Nguyen and Pham (2012) have proposed a personalized context-aware mobile learning system by accessing the contextual information including location, time and progress. Chookaew, Wanichsan, Hwang, and Panjaburee (2015) have proposed a personalized learning support system, which is to improve learning motivation based on learning performance and attitude. Wongwatkit, Srisawasdi, Hwang, and Panjaburee (2017) have further developed the personalized learning systems based on the individual performance from the formative assessment. More recently, the learner profiling techniques have been integrated with vocabulary learning theories for establishing some personalized word learning systems (Zou & Xie, 2018; Zou, Wang, Xie, Cheng, Wang, & Lee, 2020).

Recommendation algorithm is the core issue for establishing a successful personalized learning system. From the previous studies, we can find that they have focused on predicting a user's preference, whereas little attention has been given to the issue of shift of preferences. Furthermore, we can find that recommendation methods based on graph neural networks have achieved comparable success. In this study, we therefore propose a novel graph neural network to address the issue of preference shift for sequence recommendations in MOOCs.

3. Proposed framework

In this section, we first formulate our research question, then explain how to generate the user and course vector representations from the graph, and finally describe how to optimize the model parameters to recommend the Top-N.

3.1. Problem statement

Graph-based recommendations usually construct the user-item interaction graphs from session sequences (Qu, Bai, Zhang, Nie, & Tang, 2019). The set of user-item interactions can be readily modeled as a heterogeneous graph (as shown in Fig. 2(a)) $G = (\{U, C, T\}, S)$, where each $u \in U$ represents a user, each $c \in C$ represents a course and each $t \in T$ represents a course category. Each sequence in $S = \{S_0, \cdots, S_u\}$ represents a sequence interaction between a user u and a course c. We embed every user u and course c into a unified $\mathbb{R}^d$ embedding space, d is the dimension and the node vectors $e_u^* \in \mathbb{R}^d$ and $e_c^* \in \mathbb{R}^d$ indicates the latent representation of u and c, as learned via our method, (as demonstrated in Fig. 1).

3.2. Generating user embeddings

Generating user embeddings is to obtain user latent factors, denoted as $e_u^* \in \mathbb{R}^d$ for user u. The challenge is how to inherently detect the general and dynamic preferences from the user space. The user space contains two components, one is the user co-learning relationships from the user-item interaction bipartite graph G_{uc} (as shown in Fig. 2 (b)) and the second part is the user-item sequences relationship which can be regarded as G_{uc}^s (as shown in Fig. 2 (c)). To tackle this issue, two aggregate functions are employed for two graphs to capture user factors (Fan et al., 2019). The first aggregation aims to obtain user general preferences, where user latent factor $e_{u \leftarrow c} \in \mathbb{R}^d$ is learned from the G_{uc}. The second aggregation, denoted as the current sequence aggregation, is used to learn $e_{u \leftarrow s} \in \mathbb{R}^d$ as user latent factor from the G_{us}. These two factors are then integrated to obtain the final users' vector representations with an attention layer (Fan et al., 2019). The high-order connectivity information is propagated in the G_{uc} which can encode co-learning relationship to user embeddings.

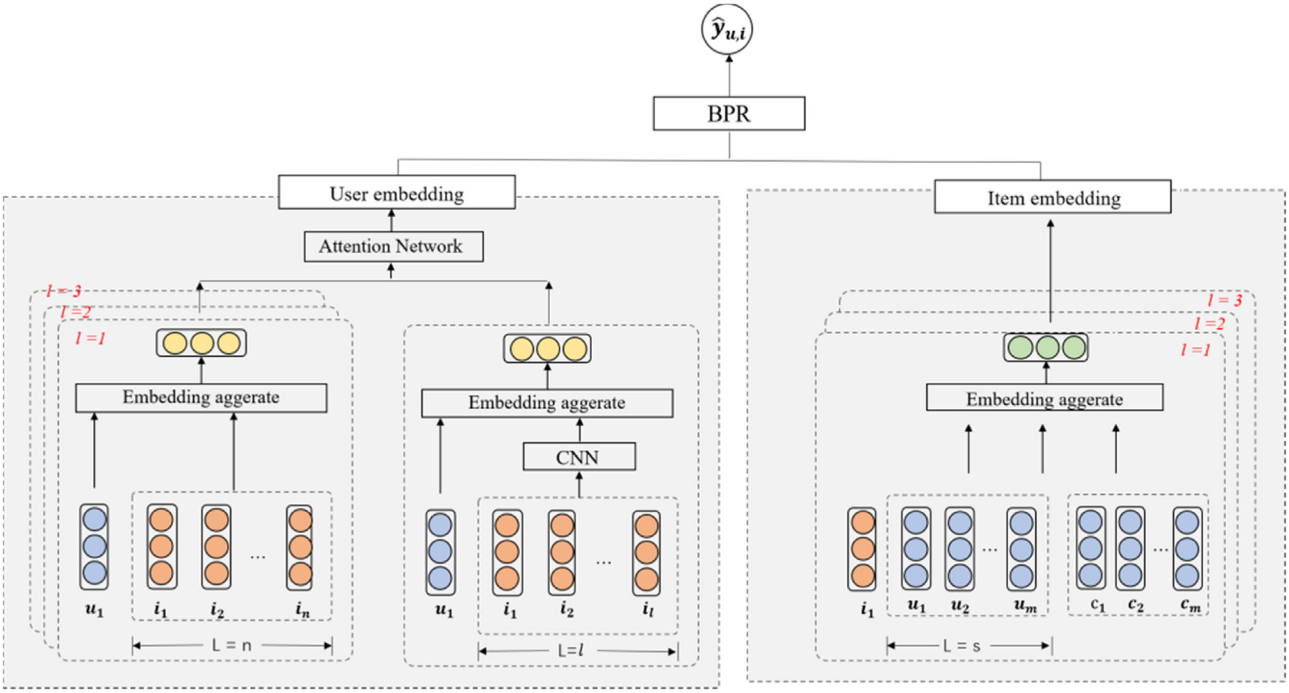


Fig. 1. The workflow of the proposed TP-GNN method. It contains three major components: user embedding, item embedding, and rating prediction.

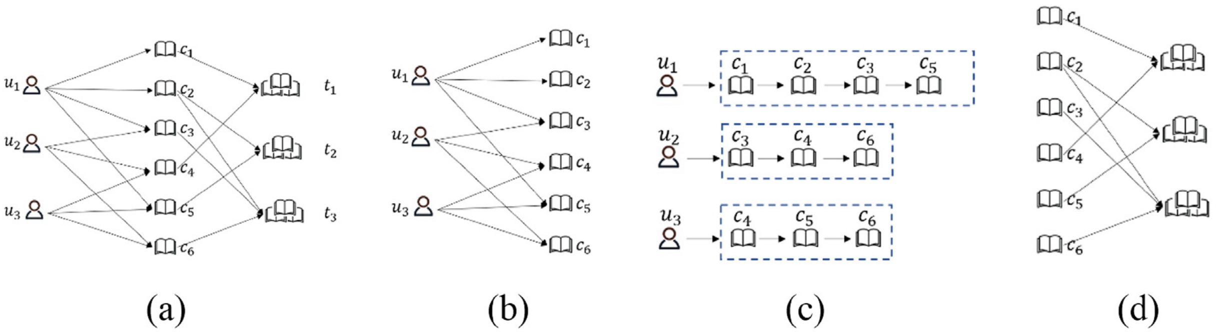


Fig. 2. The left figure (a) is the original user-item interactions graph, which can be divided into (b), (c), and (d). Fig. 2(b) is the user-item bipartite graph and Fig. 2 (c) is the user sequence relation graph. Fig. 2 (d) is the course-category graph.

- Co-learning Relationship Aggregation

Users' online study behavior may be subject to influence or can be similar to each other. The direct usage of his/her directly connected social friends or explicit co-learning relationships may lead to an inferior performance. This is because high-order connectivity helps to find the implicit co-learners who are not socially connected to the user but have similar tastes. The high-order connectivity is important to measure the relevance between users and courses. Therefore, we explicitly integrate the high-order information into users' embeddings. The most intuitive method to capture user's features or preferences is to aggregate the directly interacted items. These interacted items can be regarded as users' neighbors in G_{uc}. More specifically, the current layer of co-learning relationship space user latent factor $\mathrm{e}^{(l)}_{\mathrm{u}\leftarrow\mathrm{c}}$, is to aggregate u_i's neighbors $N(i)$ as the user's features, as follows:

$$e^{l}_{u\leftarrow c} = Aggre_{neighbors}\left(\left\{e^{(l)}_{u_i}, \forall i \in N_u\right\}\right) \tag{1}$$

$$e^{(l+1)}_{u\leftarrow c} = \sigma\left(W \cdot e^{(l)}_{u\leftarrow c}\right) + b) \tag{2}$$

where $Aggre_{neighbors}$ represents the aggregation function. $e^{l}_{u\leftarrow c}$ is the current layer l representation of user u. Recursively propagates the user's embedding representations (as Formula 2 shows) to obtain the high-order connectivity information. In our method, we adopt the mean operator as the aggregation function. The detailed updating function is as follows:

$$e^{(l+1)}_{u\leftarrow c} = \sigma\left(W \cdot \left\{\sum_{i\in N_u} \frac{1}{\sqrt{|N_u|}} e^{(l)}_{u\leftarrow c}\right\} + b\right) \tag{3}$$

The symmetric normalization term $\frac{1}{\sqrt{|N_u|}}$ follows the design of standard GCN, which can avoid the scale of embeddings increasing with graph convolution operations.

- Sequences Item Aggregation

Sequence features represent the shift of a user's preferences. Each user u has a sequence of courses (from course set C) as Fig. 2 (c) shows, $S_u = (c_0, \ldots, c_t)$, where $c_i \in C$. The index t for c_t demonstrates the order in the sequence S_u. These item sequences can be regarded as a user's current neighbors. Similar to the user co-learning relationship space, $e_{u\leftarrow s}$ is obtained as follows:

$$e_{u\leftarrow s} = \sigma\left(W \cdot Aggre_{neighbors}\left(\left\{e^{s}_{c_i}, \forall i \in N_u\right\}\right) + b\right) \tag{4}$$

where $Aggre_{neighbors}$ means apply the aggregate function on a user's neighbors. While conventional aggregate function like mean or max, in graph convolutional network ignore the sequential order of items. To address this drawback, the convolution filters in the Convolutional Neural Networks (CNN) are adopted as the aggregation function to capture local significant features as sequential features. The key step to incorporating a CNN to regular neural networks is the selection of unified neighbors, on image a CNN utilize a slide window and pixels on the grid shared weight. Due to online learning, the number of enrolled courses varies greatly. As such, the length of sequences can be very different: some sessions may only have two events, while other sessions may have more than hundreds of events. Therefore, it is quite difficult to pre-define a constant for the number of neighbors. In our model, we generate a unified number by sliding the long sequence to sub-sequences. Sub-sequence not only solve the problem of unified length, but more importantly each short sequence can be regarded as uses' short-term preference, which helps to capture the shift of user preferences.

Fig. 3 shows the process of how to generate a unified number of neighbors. Borrowing from the idea of using a Markov chain, if a users' sequence length larger than L, here L is the sub-sequence length, a sliding window is used for elements of sentences and the windowed fragments are used in the training batches. The item next to c_L is the target. This method enables aims to make sequence features to be searchable through

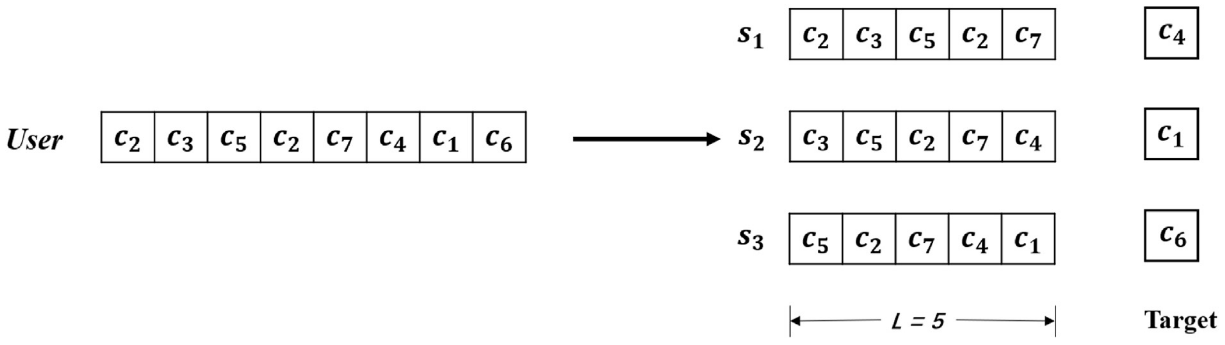

User
c_2 c_3 c_5 c_2 c_7 c_4 c_1 c_6
s_1
c_2 c_3 c_5 c_2 c_7
c_4
s_2
c_3 c_5 c_2 c_7 c_4
c_1
s_3
c_5 c_2 c_7 c_4 c_1
c_6
L = 5
Target

Fig. 3. The proposed method for generating a unified number of neighbors.

the convolution filters. Our proposed architecture learns the embedding layer by $1-D$ convolutional filters.

To obtain the significant features in the sequences and avoid the influence of noise items, we choose n filters $F^k \in \mathbb{R}^{h\times d}$, $1 \le k \le n$, filter's height is h, $h \in \{1, \cdots, L\}$. For example, if $L = 5$, the height of the filters in $\{1, 2, 3, 4, 5\}$. Filters F^k interact the sequences from top to bottom, as follows:

$$c_i^k = \sigma\left(f\left(S_{i:i+h},\ F^k\right)\right) \tag{5}$$

where $S_{i:i+h}$ is the items in the receptive field, and f denotes the convolution operator, and σ denotes the activation function. F^k, as the final output of convolution layer, is in the form of vectors:

$$c^k = \left[c_1^k c_2^k \cdots c_{L-h+1}^k\right] \tag{6}$$

Each filter captures the specific features from the sequence, including the noise feature. To avoid the noise influence and enhance the vector representation, a max pooling operator is used after the convolution to identify the salient feature in the sequence. The final vector is $e_{u\leftarrow s}$.

$$e_{u\leftarrow s} = \left\{\max\left(c^1\right), \max\left(c^2\right), \cdots, max(c^n)\right\} \tag{7}$$

- Learning User Latent Factors

Since the co-learning relationship graph and the sequence both provide inferences about users, we should consider combining the co-learning latent factors and sequence latent factors. Therefore, we propose to concatenate the two d-dimensional vectors to generate the final user latent factor e_u^* as follows:

$$e_u = \begin{bmatrix} e_{u\leftarrow c} \\ e_{u\leftarrow s} \end{bmatrix} \tag{8}$$

To discriminate between the different levels of importance and filter the uninformative features of these two factors, we integrate attention mechanisms (Li et al., 2017; Liu et al., 2018; Seo et al., 2017) assigning an individualized weight to each element using the SoftMax function. The new embedding can be given as the following:

$$\alpha_{u,i} = \frac{exp(\alpha_{u,i})}{\sum_{i\in\mathcal{M}} exp(\alpha_{u,i})} \tag{9}$$

$$e_u^* = \sum_{i\in\mathcal{M}} \alpha_{u,i} e_u \tag{10}$$

3.3. Generating course embeddings

Generating course embeddings aim to learn course latent factors, denoted as $\mathrm{e}_c^* \in \mathbb{R}^{\mathrm{d}}$ for course c_i. Course latent factor contains the common category semantic information, as well as the associated learners in the interaction graph. Therefore, these two factors should be combined to generate the final representation.

- Course Category Relationship Aggregation

High-order relation contains semantic information like courses share the common category. Course category relationship aims to aggregate the common category set of courses. To denote course latent features from the perspective of this category, neighboring courses are aggregated from the course-category graph G_{ct}. More specifically, the current layer course embedding of course-category graph $e_{c\leftarrow c}^{(l)}$ is as follows:

$$e_{c\leftarrow c}^{(l)} = Aggre_{neighbors}\left(\left\{e_{t_i}^{(l)}, \forall i \in N_u\right\}\right) \tag{11}$$

$$e_{c\leftarrow c}^{(l+1)} = \sigma\left(W \cdot e_{c\leftarrow c}^{(l)}\right) + b\Big) \tag{12}$$

where $Aggre_{neighbors}$ denotes the aggregation function on a course's neighbors. $Aggre_{neighbors}$ is the mean operator as the same used in user co-learning relationship space.

- Course-user Interaction Aggregation

Course-user aggregation is similar to the user co-leaning relationship aggregation. However, directly using the same method as the course aggregation is not suitable. This is because node's degree in the user-course learning graph varied significantly. In fact, most learners acted as visitors in MOOCs, which only enroll 3–5 courses; it is seldom for learners to choose more than 10 courses. In fact, 20% of courses have been enrolled by more than 1000 users and 25% of courses have less than 100 users. When a course has more than 1000 learners, using the method of mean integration will lead to course feature over-smoothing. To solve this problem, we followed the method of deep GCN (Chen, Ma, & Xiao, 2018), because in GCN when the depth becomes deeper in graph neural network, there are too many neighbor nodes related to the current node, which leads to over-smoothing. The random sampling method is used to enhance the robustness and generalization ability of the model. In this work, we restrict the neighborhood size by randomly sampling a fixed number set of neighbors. Then we use the similar method as user co-learning relationship aggregation to generate the course embeddings.

- Learning Course Latent Factor

We propose to concatenate the two d-dimensional vectors and use a linear function to reduce the dimension to d. Formally, the course latent factor e_c^* is defined as,

$$e_c^* = W' \begin{bmatrix} e_{c\leftarrow c} \\ e_{c\leftarrow u} \end{bmatrix} + b' \tag{13}$$

where $W' \in \mathbb{R}^{|I|\times d}$ denotes the weight matrix, and $b' \in \mathbb{R}^{|I|}$ is the bias term.

3.4. Model prediction

Finally, with the user and item embedding, the degree of user preference for the target item is predicted as:

$$\widehat{r}_{u,i} = e_u^{*T} e_c^* \tag{14}$$

To optimize the parameters of our model, we adopted the pairwise BPR loss function (Rendle, Freudenthaler, Gantner, & Schmidt-Thieme, 2012), which is commonly used in recommender systems. BPR states that items in the user-item interaction reflects users' preferences. Specifically, each observed item and unobserved one composes a pair item. BPR aims to optimize the ordering of these pairs. The objective function is as follows,

$$Loss = \sum_{(u,i,j)\in o} -ln\sigma\left(\widehat{r}_{u,i} - \widehat{r}_{u,j}\right) + \lambda\left|\left|\Theta\right|\right|_2^2 \tag{15}$$

where $o = \{(u,i,j) | \, i \in C^+, j \in C^-\}$ demonstrates the pairwise used for training, C^+ is the set of positive items, and C^- is the the set of negative items; $\sigma(\cdot)$denotes the activation function; Θ is the parameter set, and λ controls the L_2 regularization strength to prevent overfitting. To optimize the objective function, we adopt Adam as the optimizer to train our model. In our method, we will randomly sample u, all the items in the sequences will be regarded as positive, and the next enrolled in the sequences is the negative one to obtain the batch triples (u,i,j). We first generate all the item representations and then update the parameters of the model using the objective function.

4. Experiments

In this section, we discuss the effectiveness of the proposed model for top-N recommendations using the real-world dataset.

4.1. Experimental setup

4.1.1. Evaluation datasets

We adopted a real-world MOOC dataset from XuetangX, a well-known MOOCs platform in China, to evaluate our model. In our experiment, courses offered in different years will be regarded as the same. To ensure the sequential recommendation make sense, we removed users who enrolled on less than 3 courses. There remain 82,535 learners, 1,302 courses and 458,454 interactions in the dataset. The average actions of learners are 5.55.

4.1.2. Setting

Similar to the study (Tan et al., 2016), the training set, validation set, and test set are divided into 8:1:1 for the input sequence. The sequence length L is set as 7, the max number of node neighbors is set as 30, the learning rate is set as 0.0005. Xavier initializer is used for setting the initial parameters. To model the effective of high order connectivity, the depth of propagation l is set as two. The latent dimensions d is 64. Each instance in the training or the test set is a sequence of historical enrolled courses paired with a target course. During the training process, for each sequence in the training data, we hold out the last course as the target course, and the rest are treated as historical courses.

4.2. Evaluation metrics

Rec@k and Normalized Discounted Cumulative Gain (NDCG@k) (Rodriguez, Posse, & Zhang, 2012) are employed for evaluating the quality of recommendations, where Rec@k denotes the percentage of preferred items in the top-k recommended list, and NDCG@k is a precision-based metrics that accounts for the predicted position of the ground truth instance. We set K in {5, 10, 15, 20}.

4.3. Baseline methods

We compared our method with three classes of methods: (A) traditional recommender systems methods, (B) deep neural network recommender systems methods and (C) graph-structured data-based recommender systems methods. Details of the comparison methods are as follows:

POP: In this method, the most popular items in all users' sequences will be recommended to users regardless of user's online behaviors. Although it is simple, this method works as a strong baseline in some scenarios. As suggested by He et al. (2017), it is a commonly used baseline in RS.

S-POP: This method is also based on the popularity of items. The difference between POP and S-POP is that S-POP suggests the most popular items within the current session. As suggested by Hidasi et al. (2015), it is a commonly used baseline in RS.

BPR (Rendle et al., 2012)**:** Bayesian Personalized Ranking is a widely used pairwise ranking approach for RS. However, it is normally used for recommending non-sequential items.

GRU4REC (Hidasi et al., 2015): GRU4REC uses gate mechanism based on RNN to identify the item transition from session sequences.

PinSage (Berg et al., 2017)**:** This is a graph-based recommendation model, which generates the feature presentations for users/items.

4.4. Experiment 1

4.4.1. Comparison with the baseline methods

Table 1 is the performance comparison with the baseline methods on the same dataset and provides the following conclusions.

Among baselines, we can observe that PinSage and Session-GCN performs better than GRU4REC. It demonstrated that the GCNs are good at learning representations from graphs, as it naturally explicitly integrates the implicit co-learning relationship information into the item embedding from the interactions between neighbors. Although session-GCN is also based on GCN, it fails in modeling user's co-learning relationship features due to the anonymity in session.

Our proposed model TP-GNN outperforms all the compared baseline methods in general. Compared to Pinsage, our model improves 2.1% on Rec@20 and 4.6% on NDCG@10, suggesting the effectiveness of integrating the sequential relations and the users' general preferences. Compared to Session-GCN and GRU4REC, our method reduces the influence of user's enrolling out of curiosity or by accident in the sequence instead of interests, which is a common case in MOOCs, due to the high dropout ratio in MOOCs.

In traditional algorithms, BPR achieves better performance than others, which indicates the importance of item contextual information in sequences. Such simple models like POP and S-POP, make recommendations based on item frequencies, which has relatively poor performance in session-based recommendation scenarios.

In the next studies, we further analyze the impact of the hyperparameters sequence length L and the sample size s by varying the dataset and remaining other hyperparameters at their optimal settings.

4.4.2. Influence of the layer length

To further understand the work of the multiple layer co-leaning propagation, we vary the layer depth in {1,2,3}. Table 2 denotes the performance influenced by layers, TP-GNN$_1$ indicates apply a single embedding propagation layer in the model, and others with similar notations. The followings are our findings:

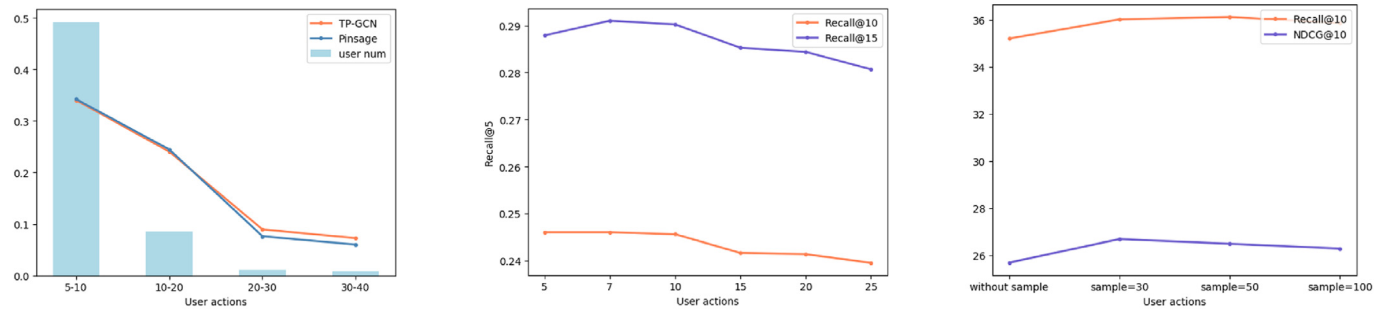
- Different propagation layer can adopt different information into the item representations. In TP-GNN$_1$, there is no high order relations, only taking the direct connection neighbors into consideration.TP-GNN$_2$ and TP-GNN$_3$ outperform the TP-GNN$_1$, which demonstrates the importance of the high-order information.
- When increasing the depth of propagation layers from 2 to 3, TP-GNN degrades the performance. This is because a higher depth of the layer means high order neighbors will be considered to exact the feature as

Table 1
Recommendation performances.

	Rec@5	Rec@10	Rec@15	Rec@20	NDCG@5	NDCG@10	NDCG@15	NDCG@20
POP	7.28%	12.43%	19.34%	25.48%	5.80%	7.79%	9.86%	11.67%
BPR	21.67%	34.54%	39.05%	42.56%	13.97%	18.04%	20.33%	19.42%
S-POP	10.21%	14.33%	20.04%	26.48%	10.21%	12.19%	15.11%	17.01%
GRU4REC	24.87%	34.04%	38.05%	43.56%	14.30%	18.04%	21.01%	20.33%
Session-GCN	25.43%	36.68%	38.91%	46.57%	14.50%	18.00%	20.31%	21.63%
PinSage	24.99%	37.83%	44.96%	49.79%	14.70%	19.90%	22.19%	23.58%
TP-GNN	**26.35%**	**38.82%**	**46.11%**	**50.85%**	**15.51%**	**20.83%**	**22.47%**	**23.91%**

Table 2
Effect of graph convolution layers.

	Rec@5	Rec@10	Rec@15	Rec@20	NDCG@5	NDCG@10	NDCG@15	NDCG@20
TP-GNN$_1$	25.76%	37.09%	44.02%	49.89%	14.63%	19.87%	19.04%	23.66%
TP-GNN$_2$	26.35%	38.82%	46.11%	50.85%	15.51%	20.83%	22.47%	23.91%
TP-GNN$_3$	26.53%	37.73%	45.59%	50.27%	23.12%	20.79%	22.00%	23.59%



(a) (b) (c)

Fig. 4. (a) and (b) is the performance comparison over the sequence length (sequence sparsity). (c) is the performance comparison over different number of samples.

the center node representation. More neighbors make the center node easier 'over-smooth', in that case high order neighbors carried noise to the representation learning instead of the reliable social information. In our dataset, two layers of the propagation is the best.

4.4.3. Analysis on session sequence length (sequence sparsity)

Sequence sparsity means the intensity of sub-sequences collected from the use interactions. A small sequence length L decides a high sequence sparsity, for example, for an input sequences $S_u = (c_0, c_1, c_2, \ldots, c_t)$, we generate a series of sequences and target item $([c_0, c_1, \ldots, c_L], c_{L+1})$, $([c_1, c_2, \ldots, c_{L+1}], c_{L+2})$, . . . , $([c_n, c_{n+1}, \ldots, c_{n+L}], c_{L+n+1})$, where $[c_n, c_{n+1}, \ldots, c_{n+L}]$ is the generated sequence and c_{l+n+1} denotes the next-clicked item, in our method is the positive item. We vary L in the same dataset and the sequence sparsity from different datasets to analyze the effectiveness of Pinsage and our method while other optimal hyper-parameters are unchanged. In Fig. 4(a), we divided the whole dataset into four sub datasets depending on the number of user actions: the interaction numbers per user are between in 5–10, 10–20, 20–30, and more than 30 respectively. The bar in Fig. 4(a) is the total number of users of the sub dataset. We specify L equals ten, the results show that:

- When L is ten and the max of user sequences length is also ten, the result is not significantly different about the Pinsage and our method, because when L equals the sequences length, all the sequences will be as the CNN input to extract the significant features in the user's general preferences. Current items cannot be effectively captured. From Fig. 4, we can observe that increasing the value of L, the benefits of the current items become obvious, which demonstrates the effectiveness of the current items drift.
- To better explore the value of L, we conducted the experiment using the same dataset as different value L. As Fig. 4(b) shows, we select the user actions number between 10 and 20 as the dataset, L varied in $\{5, 7, 10, 15, 20, 25\}$. Fig. 4 shows when L is larger than 20, the result obtains a worse performance.

That said, when L is lower than 20, the result has no significant different in our experiment, since it does not equate users enrolled in more than 10 courses. In general, a smaller L will improve the performance due to a dense sequence sparsity. Nevertheless, if the value of L is too small, not only adds noise to the model, but also increases the complexity of model. Therefore, a proper value of sequence L need to consider balance sequence sparsity and complexity.

4.4.4. Analysis on sample number

Fig. 4(c) shows the performance comparison with different sample size in the original dataset. In general, sample method will improve the performance. When increasing the sample size from 30 to 50 there is no significant different. However, continually increase the sample size to 100, the performance decreases. It demonstrates that using a sample mechanism can avoid the overfitting and our method is not sensitive to the sample size.

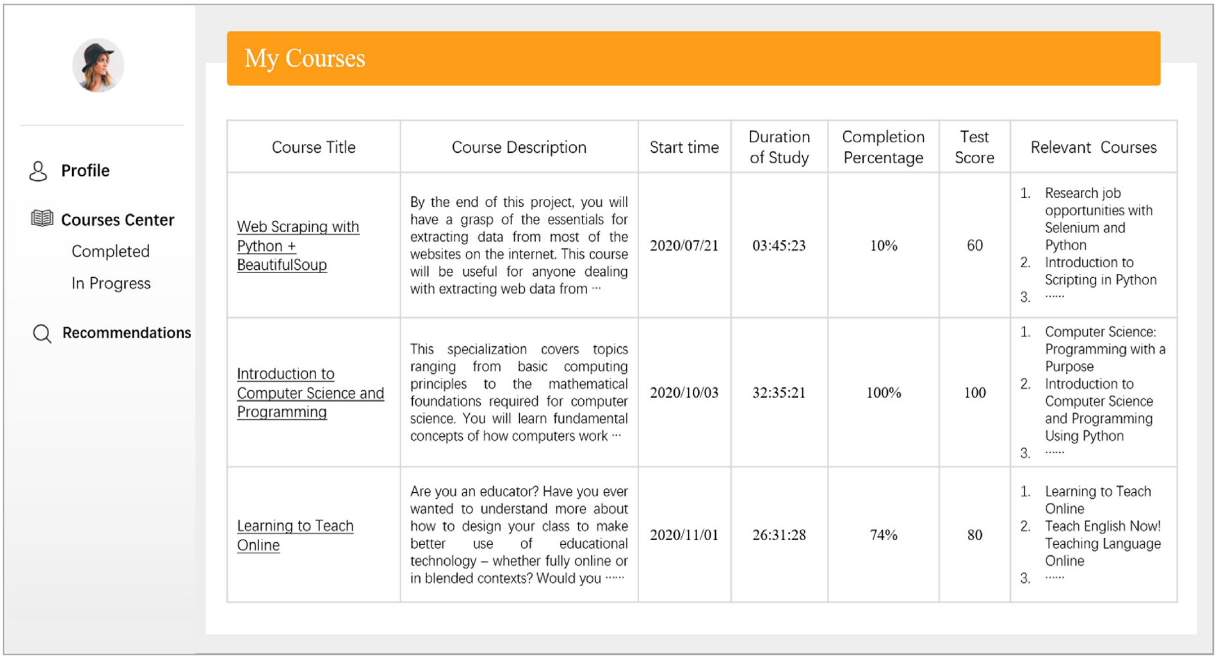
4.5. Experiment 2

Experiment 1 demonstrates the effectiveness of our method TP-GNN on various common evaluation metrics. It is equally important to investigate whether the system could benefit learners within an e-learning environment. As a result, another user evaluation is conducted to understand the learners' actual perception on the effectiveness of the system.

The optimal hyperparameters obtained in Experiment 1 are adopted in the following experiment. The proposed recommendation system contains course data about computer science, education technology and English learning. The effectiveness of our method is evaluated by benchmarking with GRU4REC. Among the baseline methods, GNN-based recommenders significantly outperform RNN-based recommenders. GRU4REC is a RNN-based recommender and achieve results comparable to GNN-based recommenders in various studies. Because our system is designed based on GNN, GRU4REC is selected for a fair comparison.

The system recommended the top 10 courses based on two models (i.e., GRU4REC and TP-GNN). Three participants from our research team who have academic background from education technology and computer science participated in this user study. During the user study, each participant was provided with ten learner profiles from database, and then identified three most similar learner profiles based on their perceptions. Specifically, each participant simulated these three users, and then evaluated the relevance for top 10 recommended courses by two different models. They scored each recommended course ranging from 1 to 5 (1 = very irrelevant, 2 = irrelevant, 3 = fair, 4 = relevant, 5 = very relevant). In addition, the system prototype interfaces are shown in Figs. 5 and 6.

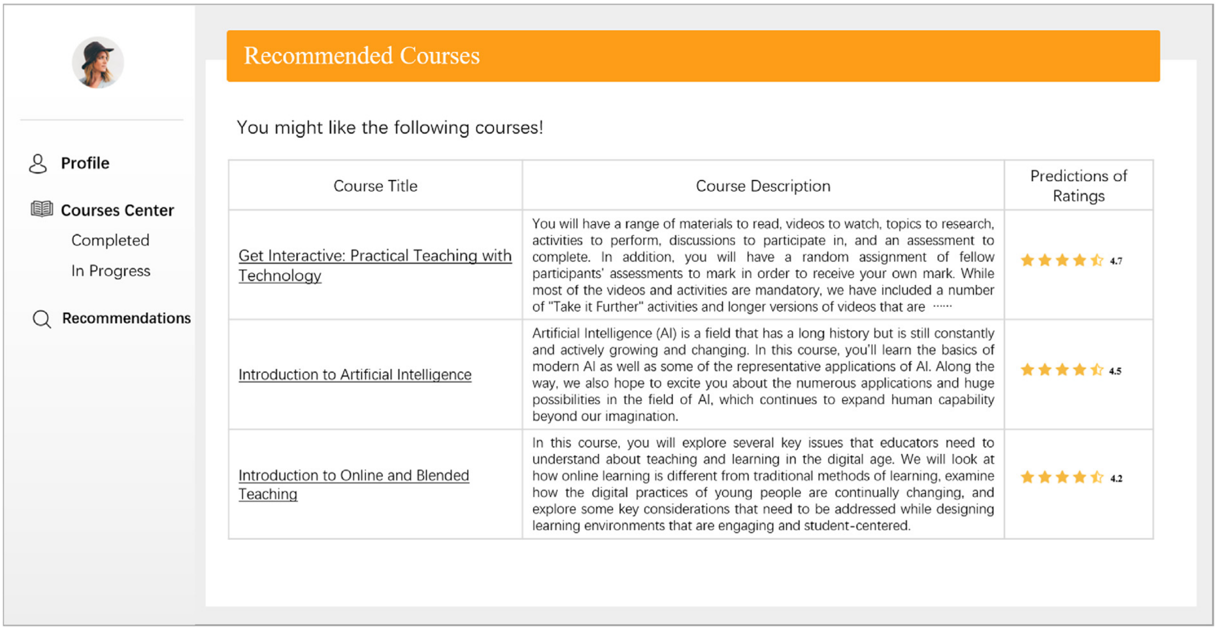
The simulated relevance perception of participants with the recommended results of two models are presented in Table 3. Both TP-GNN and GRU4REC have achieved excellent performance. For TP-GNN, 30.0% recommended courses are marked as "Very Relevant"; 30.0%



My Courses

Course Title	Course Description	Start time	Duration of Study	Completion Percentage	Test Score	Relevant Courses
Web Scraping with Python + BeautifulSoup	By the end of this project, you will have a grasp of the essentials for extracting data from most of the websites on the internet. This course will be useful for anyone dealing with extracting web data from ···	2020/07/21	03:45:23	10%	60	1. Research job opportunities with Selenium and Python 2. Introduction to Scripting in Python 3. ······
Introduction to Computer Science and Programming	This specialization covers topics ranging from basic computing principles to the mathematical foundations required for computer science. You will learn fundamental concepts of how computers work ···	2020/10/03	32:35:21	100%	100	1. Computer Science: Programming with a Purpose 2. Introduction to Computer Science and Programming Using Python 3. ······
Learning to Teach Online	Are you an educator? Have you ever wanted to understand more about how to design your class to make better use of educational technology – whether fully online or in blended contexts? Would you ······	2020/11/01	26:31:28	74%	80	1. Learning to Teach Online 2. Teach English Now! Teaching Language Online 3. ······

Fig. 5. An illustrative example of the learner interface.



Recommended Courses

You might like the following courses!

Course Title	Course Description	Predictions of Ratings
Get Interactive: Practical Teaching with Technology	You will have a range of materials to read, videos to watch, topics to research, activities to perform, discussions to participate in, and an assessment to complete. In addition, you will have a random assignment of fellow participants' assessments to mark in order to receive your own mark. While most of the videos and activities are mandatory, we have included a number of "Take it Further" activities and longer versions of videos that are ······	★★★★☆ 4.7
Introduction to Artificial Intelligence	Artificial Intelligence (AI) is a field that has a long history but is still constantly and actively growing and changing. In this course, you'll learn the basics of modern AI as well as some of the representative applications of AI. Along the way, we also hope to excite you about the numerous applications and huge possibilities in the field of AI, which continues to expand human capability beyond our imagination.	★★★★☆ 4.5
Introduction to Online and Blended Teaching	In this course, you will explore several key issues that educators need to understand about teaching and learning in the digital age. We will look at how online learning is different from traditional methods of learning, examine how the digital practices of young people are continually changing, and explore some key considerations that need to be addressed while designing learning environments that are engaging and student-centered.	★★★★☆ 4.2

Fig. 6. An illustrative example of the learner interface for recommendation results.

Table 3
The simulated relevance scores by participants with the two methods

	Very Relevant (5)	Relevant (4)	Fair (3)	Irrelevant (2)	Very Irrelevant (1)
GRU4REC	33.3%	28.8%	31.1%	6.6%	0%
TP-GNN	30.0%	30.0%	34.4%	5.5%	0%

recommended courses are marked as “Relevant”; 34.4% recommended courses are marked as “Fair”; 5.5% recommended courses are marked as “Irrelevant”; and no recommended courses are marked as “Very Irrelevant”. For GRU4REC, 33.3% recommended courses are marked as “Very Relevant”; 28.8% recommended courses are marked as “Relevant”; 31.1% recommended courses are marked as “Fair”; 6.6% recommended courses are marked as “Irrelevant”; and no recommended courses are marked as “Very Irrelevant”. From the results, we can find that TP-GNN has more stable result than GRU4REC for the recommended courses above and equal to “Fair” score, whereas GRU4REC slightly outperforms TP-GNN in terms of the ratio of courses with “Very Relevant” score.

5. Conclusion

In this work, we propose a Top-N personalized course recommendation method based on graph convolution network. Our method explicitly incorporated the co-learning relationship and co-category relationship into the item embedding as users' general preferences. To capture users' current preferences, we modified the sequences as the CNN input, which avoids the limitation in RNN that all adjacent items are dependent on each other. With the general preference level, TP-GNN learns how to embed items with high-order connectivity. Additionally, TP-GNN uses CNN to extract significant features of the sequential patterns with the current of items. The experimental results from using a real-world dataset demonstrate the effectiveness of our model. In the next step, more learning behaviors such as learning progress, opinions will be considered to explore learner characteristics.

Statements on open data and ethics

The participants were protected by hiding their personal information in this study. They were voluntary and they knew that they could withdraw from the experiment at any time. The data can be provided upon requests by sending e-mails to the corresponding author.

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.

Acknowledgements

The work described in this paper was supported by the Katie Shu Sui Pui Charitable Trust — Academic Publication Fellowship (Project Reference No. KSPF2019-03) of the Open University of Hong Kong, Hong Kong, HKIBS Research Seed Fund 2019/20 (190-009) and the Research Seed Fund (102367) of Lingnan University, Hong Kong.

References

Berg, R.v. d., Kipf, T. N., & Welling, M. (2017). *Graph convolutional matrix completion.* arXiv preprint arXiv:1706.02263.

Chen, C.-M., Hsu, S.-H., Li, Y.-L., & Peng, C.-J. (2006). Personalized intelligent m-learning system for supporting effective English learning. In *Paper presented at the 2006 IEEE international conference on systems, man and cybernetics*.

Chen, J., Ma, T., & Xiao, C. (2018). *Fastgcn: Fast learning with graph convolutional networks via importance sampling.* arXiv preprint arXiv:1801.10247.

Chookaew, S., Wanichsan, D., Hwang, G.-J., & Panjaburee, P. (2015). Effects of a personalised ubiquitous learning support system on university students' learning performance and attitudes in computer-programming courses. *International Journal of Mobile Learning and Organisation, 9*(3), 240–257.

Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., et al. (2019). Graph neural networks for social recommendation. In *Paper presented at the the world wide web conference* (pp. 417–426).

Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., & Yin, D. (2019). Graph neural networks for social recommendation. In *The World Wide Web Conference* (pp. 417–426).

Fan, S., Zhu, J., Han, X., Shi, C., Hu, L., Ma, B., et al. (2019). Metapath-guided heterogeneous graph neural network for intent recommendation. In *Paper presented at the proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*.

Figueiredo, F., Ribeiro, B., Almeida, J. M., & Faloutsos, C. (2016). TribeFlow: Mining & predicting user trajectories. In *Paper presented at the Proceedings of the 25th international conference on world wide web*.

Gupta, A., Jain, S., Goyal, P., & Lalit, A. (2020). *Engagement analysis in MOOCs* (pp. 56–64). CSI Adhyayan.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. In *Paper presented at the Proceedings of the 26th international conference on world wide web*.

Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). *Session-based recommendations with recurrent neural networks.* arXiv preprint arXiv:1511.06939.

Hsu, M.-H. (2008). A personalized English learning recommender system for ESL students. *Expert Systems with Applications, 34*(1), 683–688.

Jishan, S. T., & Wang, Y. (2017). Audience activity recommendation using stacked-LSTM based sequence learning. In *Proceedings of the 9th International Conference on Machine Learning and Computing* (pp. 98–106).

Kim, D., Park, C., Oh, J., Lee, S., & Yu, H. (2016). Convolutional matrix factorization for document context-aware recommendation. In *Paper presented at the Proceedings of the 10th ACM conference on recommender systems*.

Koren, Y. (2008). Factorization meets the neighborhood: A multifaceted collaborative filtering model. In \.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer, 42*(8), 30–37.

Li, X., & Chen, H. (2013). Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. *Decision Support Systems, 54*(2), 880–890.

Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., & Ma, J. (2017). Neural attentive session-based recommendation. In *Paper presented at the proceedings of the 2017 ACM on conference on information and knowledge management*.

Liu, D., Li, J., Du, B., Chang, J., & Gao, R. (2019). Daml: Dual attention mutual learning between ratings and reviews for item recommendation. In *Paper presented at the proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*.

Liu, Q., Zeng, Y., Mokhosi, R., & Zhang, H. (2018). STAMP: Short-term attention/memory priority model for session-based recommendation. In *Paper presented at the proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*.

Lu, Y., Dong, R., & Smyth, B. (2018). Coevolutionary recommendation model: Mutual learning between ratings and reviews. In *Paper presented at the proceedings of the 2018 world wide web conference*.

Luo, L., Xie, H., Rao, Y., & Wang, F. L. (2019). Personalized recommendation by matrix co-factorization with tags and time information. *Expert Systems with Applications, 119*, 311–321.

Monti, F., Bronstein, M., & Bresson, X. (2017). Geometric matrix completion with recurrent multi-graph neural networks. In *Paper presented at the advances in neural information processing systems*.

Nguyen, V. A., & Pham, V. C. (2012). CAMLES: An adaptive mobile learning system to assist student in language learning. In *Paper presented at the 2012 IEEE seventh international conference on wireless, mobile and ubiquitous technology in education*.

Pazzani, M. J., & Billsus, D. (2007). *Content-based recommendation systems The adaptive web*. Springer.

Qiu, R., Li, J., Huang, Z., & Yin, H. (2019). Rethinking the item order in session-based recommendation with graph neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (pp. 579–588).

Qu, Y., Bai, T., Zhang, W., Nie, J., & Tang, J. (2019). An end-to-end neighborhood-based interaction model for knowledge-enhanced recommendation. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data* (pp. 1–9).

Rendle, S. (2010). Factorization machines. In *Paper presented at the 2010 IEEE international conference on data mining*.

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). *BPR: Bayesian personalized ranking from implicit feedback*. arXiv preprint arXiv:1205.2618.

Rodriguez, M., Posse, C., & Zhang, E. (2012). Multiple objective optimization in recommender systems. In *Proceedings of the sixth ACM conference on Recommender systems* (pp. 11–18).

Salakhutdinov, R., Mnih, A., & Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. In *Paper presented at the Proceedings of the 24th international conference on Machine learning*.

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). *Collaborative filtering recommender systems The adaptive web*. Springer.

Seo, S., Huang, J., Yang, H., & Liu, Y. (2017). Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Paper presented at the proceedings of the eleventh ACM conference on recommender systems*.

Song, W., Xiao, Z., Wang, Y., Charlin, L., Zhang, M., & Tang, J. (2019). Session-based social recommendation via dynamic graph attention networks. In *Paper presented at the proceedings of the twelfth ACM international conference on web search and data mining*.

Tang, J., & Wang, K. (2018). Personalized top-n sequential recommendation via convolutional sequence embedding. In *Paper presented at the proceedings of the eleventh ACM international conference on web search and data mining*.

Tan, Y. K., Xu, X., & Liu, Y. (2016). Improved recurrent neural networks for session-based recommendations. In *Paper presented at the proceedings of the 1st workshop on deep learning for recommender systems*.

Tuan, T. X., & Phuong, T. M. (2017). 3D convolutional networks for session-based recommendation with content features. In *Paper presented at the proceedings of the eleventh ACM conference on recommender systems*.

Wang, Z., Tan, Y., & Zhang, M. (2010). Graph-based recommendation on social networks. In *Paper presented at the 2010 12th international asia-pacific web conference*.

Wang, H., Wang, N., & Yeung, D.-Y. (2015). Collaborative deep learning for recommender systems. In *Paper presented at the Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*.

Wang, H., Zhang, F., Xie, X., & Guo, M. (2018). DKN: Deep knowledge-aware network for news recommendation. In *Paper presented at the Proceedings of the 2018 world wide web conference*.

Wang, B., Xiong, S., Huang, Y., & Li, X. (2018). Review rating prediction based on user context and product context. *Applied Sciences, 8*(10), 1849.

Wang, X., He, X., Wang, M., Feng, F., & Chua, T. S. (2019). Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval* (pp. 165–174).

Wongwatkit, C., Srisawasdi, N., Hwang, G.-J., & Panjaburee, P. (2017). Influence of an integrated learning diagnosis and formative assessment-based personalized web learning approach on students learning performances and perceptions. *Interactive Learning Environments, 25*(7), 889–903.

Xie, H., Chu, H.-C., Hwang, G.-J., & Wang, C.-C. (2019). Trends and development in technology-enhanced adaptive/personalized learning: A systematic review of journal publications from 2007 to 2017. *Computers & Education, 140*, 103599.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. In *Paper presented at the proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*.

Zhang, Y., Dai, H., Xu, C., Feng, J., Wang, T., Bian, J., … Liu, T.-Y. (2014). *Sequential click prediction for sponsored search with recurrent neural networks*. arXiv preprint arXiv:1404.5772.

Zhang, C., Song, D., Huang, C., Swami, A., & Chawla, N. V. (2019). Heterogeneous graph neural network. In *Paper presented at the proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*.

Zheng, L., Lu, C.-T., Jiang, F., Zhang, J., & Yu, P. S. (2018). Spectral collaborative filtering. In *Paper presented at the proceedings of the 12th ACM conference on recommender systems*.

Zou, D., Wang, M., Xie, H., Cheng, G., Wang, F. L., & Lee, L. K. (2020). A comparative study on linguistic theories for modeling EFL learners: facilitating personalized vocabulary learning via task recommendations. *Interactive Learning Environments*, 1–13. https://doi.org/10.1080/10494820.2020.1789178

Zou, D., & Xie, H. (2018). Personalized word-learning based on technique feature analysis and learning analytics. *Journal of Educational Technology & Society, 21*(2), 233–244.