electronics

MDPI

Article

Automatic Generation of Meta-Path Graph for Concept Recommendation in MOOCs

Jibing Gong [1,2,3,*,†], **Cheng Wang** [1,2,3,†], **Zhiyong Zhao** [1,2,3] **and Xinghao Zhang** [1,2,3]

[1] School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China; wangcheng_ysu@163.com (C.W.); ys_zhaozhiyong@163.com (Z.Z.); 13293151871@163.com (X.Z.)
[2] The Key Lab for Computer Virtual Technology and System Integration of Hebei Province, Yanshan University, Qinhuangdao 066004, China
[3] Key Laboratory for Software Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004, China
[*] Correspondence: gongjibing@ysu.edu.cn
[†] These authors contributed equally to this work.

Abstract: In MOOCs, generally speaking, curriculum designing, course selection, and knowledge concept recommendation are the three major steps that systematically instruct users to learn. This paper focuses on the knowledge concept recommendation in MOOCs, which recommends related topics to users to facilitate their online study. The existing approaches only consider the historical behaviors of users, but ignore various kinds of auxiliary information, which are also critical for user embedding. In addition, traditional recommendation models only consider the immediate user response to the recommended items, and do not explicitly consider the long-term interests of users. To deal with the above issues, this paper proposes AGMKRec, a novel reinforced concept recommendation model with a heterogeneous information network. We first clarify the concept recommendation in MOOCs as a reinforcement learning problem to offer a personalized and dynamic knowledge concept label list to users. To consider more auxiliary information of users, we construct a heterogeneous information network among users, courses, and concepts, and use a meta-path-based method which can automatically identify useful meta-paths and multi-hop connections to learn a new graph structure for learning effective node representations on a graph. Comprehensive experiments and analyses on a real-world dataset collected from XuetangX show that our proposed model outperforms some state-of-the-art methods.

Keywords: concept recommendation; MOOCs; heterogeneous information network; reinforcement learning

check for updates

Citation: Gong, J.; Wang, C.; Zhao, Z.; Zhang, X. Automatic Generation of Meta-Path Graph for Concept Recommendation in MOOCs. *Electronics* **2021**, *10*, 1671. https://doi.org/10.3390/electronics10141671

Academic Editors: Feliz Gouveia and Amir H. Gandomi

Received: 8 May 2021
Accepted: 8 July 2021
Published: 13 July 2021

1. Introduction

Massive Open Online Courses (MOOCs), aimed at unlimited participation and open access via the web, are rapidly becoming an established online and distant education method. For example, Coursera, edX, and Udacity, the three pioneering MOOC platforms, offer millions of users access to numerous courses from internationally renowned universities. In China, XuetangX MOOCs has offered more than 1000 courses and attracted over 6,000,000 users worldwide; it is one of the largest MOOCs platforms in China. In MOOCs, We use course concepts to refer to the knowledge concepts taught in course videos and help users better to understand the related topics of course videos. The goal of concept recommendation is to recommend related topics to users to facilitate their online study.

Many existing efforts have been made towards user behavior understanding and concept extraction in MOOCs, such as prerequisite relation mining among knowledge concepts [1], course concept extraction [2], learning behavior prediction [3], and course recommendation [4,5]. Pan et al. propose to learn latent representation via an embedding-based method for course concept extraction in MOOCs [2]. Both user interests and profiles are employed to feed a proposed content-aware deep learning framework for course recommendation in MOOCs [4,5].

However, these approaches still suffer from two major limitations: (a) They ignore rich heterogeneous information across MOOCs. These approaches [5] fully consider the semantic information of user-profiles and leverage diverse historical courses to make personalized course recommendations. Nevertheless, it is not enough to mine potential semantics because a larger amount of semantic information hidden in relations among different entities in MOOCs is not exploited. (b) They cannot consider current reward and future reward simultaneously from an online recommendation in the dynamic learning environment. For example, although Pan et al. leverage the demographics and course prerequisite relation to better reveal users' potential choices, they overlook the MOOCs system as a dynamic learning environment, unable to model the current reward and future reward of users' choices [2]. This results in that its approach cannot provide personalized candidate concepts.

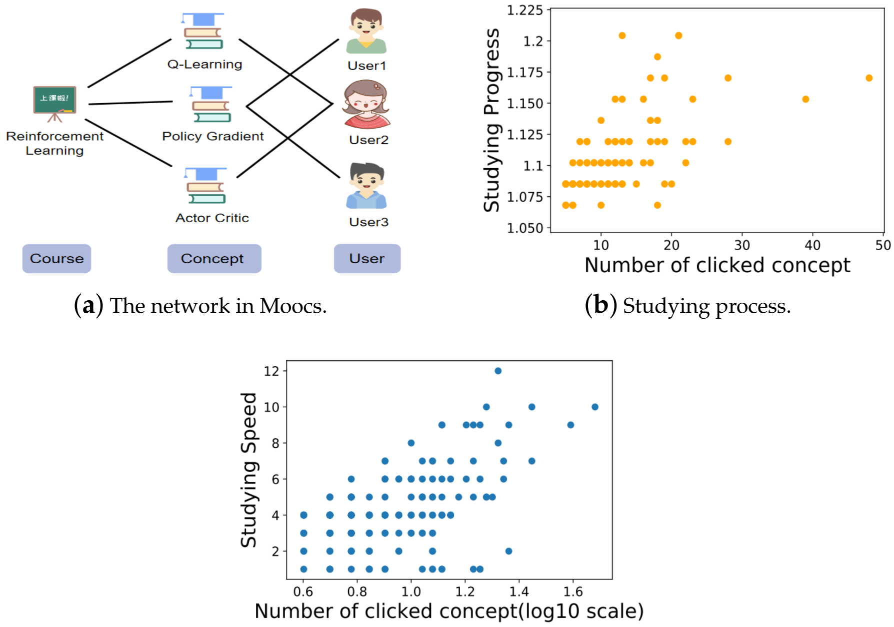
These limitations mentioned above motivate us to design a model that learns more comprehensive representations of users and offers personalized concept recommendation. We model XuetangX MOOCs as a Heterogeneous Information Network (HIN) and propose a concept recommendation framework with reinforcement learning (RL) to offer a personalized and dynamic knowledge concept label list to users for getting a course certificate. For the limitation (a), it is common sense that users with diverse backgrounds or levels have different domains of expertise, and the reinforcement learning course is well accepted by certain users but might be hard for others. Meanwhile, existing concept labels listed in a course video were previously provided by the teacher, but only in a fixed and single way. It neither dynamically considered users' historical learning behaviors, nor utilized other users' global progress on the MOOCs. For the limitation (b), the reasons why we employ reinforcement learning in this study include: (1) The concept clicking rate in MOOCs is relatively sparse [6], (2) in real online learning scenarios of MOOCs, the recommender usually interacts with the users for multiple rounds [7], and (3) interactions between users and the recommender agent should be sequential [8].

Figure 1 illustrates our motivation of this work. Figure 1a gives an example to show the relationships among users, courses, and concepts in MOOCs. Specifically, a course reinforcement learning may contain many related concept labels, e.g., Q-learning, policy gradient, and actor-critic. Different users can click many concept labels to instruct his/her learning of the reinforcement learning course. Figure 1b tells us that the more concept labels are clicked by a user when learning a course, the bigger the studying process will be that is achieved by him/her. Figure 1c states that the more concept labels are clicked by a user when learning a course, the faster the studying speed will be obtained by him/her. Both of them deliver the significant importance of concepts on users' learning in MOOCs. The former studying progress denotes the extent to which the concept label clicking can influence the effectiveness of online course learning of a user, but the latter studying speed states to what extent the course can be learned by a user considering the efficiency. Based on these discoveries, the paper studies the problem of concept recommendation in a fine-grained view when compared with the course recommendation.

In summary, the main contributions of the paper are as follows.

- We propose a novel model which can automatically identify effective meta-paths and multi-hop connections to better represent users with sparse data in the heterogeneous information network of MOOCs. Furthermore, we utilize the reinforcement learning framework to capture users' long-term interests and generate personalized dynamic recommendation lists.
- Unlike the previous studies, we investigate concept recommendation, more fine-grained than course recommendation, in XuetangX MOOCs from the perspective of reinforcement learning.
- We validate the effectiveness of our proposed model on a real-world dataset collected from XuetangX MOOCs. Comprehensive experiments and analyses show that our proposed model is superior to some state-of-the-art methods.

The remainder of the paper is organized as follows. We highlight some related works with comparisons in Section 2. In Section 3, we introduce some preliminaries about heterogeneous information networks and reinforcement learning. Section 4 presents the details of our proposed approach. Section 5 shows the experimental results and analyses. Finally, we conclude the paper and propose some future research directions in Section 6.



(**a**) The network in Moocs.

(**b**) Studying process.

(**c**) Studying speed.

Figure 1. An illustration of concept recommendation. (**a**) The network in MOOCs. (**b**,**c**) The correlation between the number of clicked concepts and the progress as well as speed of studying, respectively.

2. Related Work

In this part, we review the related studies from three perspectives, namely existing works on mining in MOOCs, recommender systems, and reinforcement learning for recommendation.

2.1. Mining in MOOCs

As a newly emerging HIN, MOOCs are more typical and contain richer semantics in objects and links, and thus they form a new development for data mining. Among previous works on MOOCs, such as course concept extraction [2], learning behavior predicting [3], and course recommendation [4,5], the course recommendation is the work most similar to us, and employs a hierarchical reinforcement learning algorithm to revise the user profiles and tune the course recommendation model on the revised profiles. However, our work is different from it in many aspects. Our work considers MOOCs as a large HIN with abundant semantic information to recommend more fine-grained knowledge concepts to users. Furthermore, we integrate meta-path-based embedding of the HIN with an extended deep reinforcement learning framework to recommend knowledge concepts.

2.2. Recommender Systems

Early works mainly adopt collaborative filtering (CF) or content-based methods to complete recommendation tasks. CF utilizes historical interactions for recommendation, either explicit or implicit feedback. The content-based recommendation is based on comparisons between items and users' auxiliary information. However, the two methods usually suffer from serious cold start problems and data sparsity issues [9,10]. Currently, many recommender systems focus on enriching the semantic representation of users and items based on deep learning. For example, Ma et al. [11] integrate social relations into matrix factorization in recommendation. As a newly emerging direction, HIN can naturally model complex objects and their rich relations in recommender systems [12,13].

The comprehensive information integration and rich semantic information of the HIN make it the focus of everyone's research [14,15]. Sun et al. [16] firstly propose to explore the meta-structure of the information network, i.e., put forward the concept of the meta-path to systematically capture numerous semantic relationships across multiple types of objects defined as a path over the graph of network schema. Meta-path can guide searching and mining the network (e.g., recommendation), and help to analyze the semantic meaning of the objects and relations in the network [17]. Recently, Graph Transformer Networks [18] has shown its effectiveness in HIN embedding. Meta-path-based methods have shown performance superior to other existing methods due to their excellent modeling of heterogeneous information.

2.3. Reinforcement Learning for Recommendation

Reinforcement learning has been widely used in the recommendation field [8,19–22]. There are several RL works on movie recommendation [19], news recommendation [20], and music recommendation [21]. However, these methods have two major issues. First, they fail to capture semantic relations among objects and usually employ the rating scores between users and items in the recommender system. Thus, they recommend similar objects, which may cause users to get bored. Different from the existing RL-based recommendation work [23,24], our model identifies effective meta-paths and leverages rich meta-path-based contexts to learn interaction-specific representation for users, courses and concepts. <user, meta-path, concept> has been explicitly modeled in an RL-based interaction model for the task of multi-round concepts recommendation in MOOCs.

3. Preliminaries

Before introducing our proposed method, in this section, we first introduce some background about HIN and the RL-based frameworks. Furthermore, we will give some related preliminaries about our proposed method.

3.1. Heterogeneous Information Network

A HIN is a special kind of information network containing multiple types of objects or multiple types of links.

Definition 1. *Heterogeneous information network [25]. A HIN is denoted as $G = \{V, E\}$ consisting of an object set V and a link set E. A HIN is also associated with an object type mapping function $\Phi : V \to A$ and a link type mapping function $\varphi : E \to R$. A and R denote the sets of the predefined object and link types, where $|A| + |R| > 2$. As shown in Figure 1a, we build a simple heterogeneous information network with three entities: User, course, and concept.*

Definition 2. *Network schema [25]. The network schema is denoted as $S = (A, R)$. It is a meta template for an information network $G = \{V, E\}$ with the object type mapping $\phi : V \to A$ and the link type mapping $\varphi : E \to R$, which is a directed graph defined over object types A, with edges as relations from R. We define the network schema of XuetangX in Figure 2 and our proposed model generates meta-paths based on this network schema.*

Definition 3. *Meta-path [16]. A meta-path ρ is defined on a network schema $S = (A, R)$ and is denoted as a path in the form of $A_1 \overset{R_1}{\to} A_2 \overset{R_2}{\to} \cdots \overset{R_l}{\to} A_{l+1}$ (abbreviated as $A_1 A_2 \ldots A_{l+1}$), which describes a composite relation $R = R_1 \circ R_2 \circ \ldots \circ R_l$ between object A_1 and A_{l+1}, where $\circ$ denotes the composition operator on relations. In our scenario, we pre-define two meta-paths to compare with the meta-paths our model generates.*

- **U$_1$-K-U$_2$**: Denotes users 1 and 2 click the same knowledge concept.

- **U$_1$-K$_1$-C-K$_2$-U$_3$**: Associates two users who click different concepts in the same course.

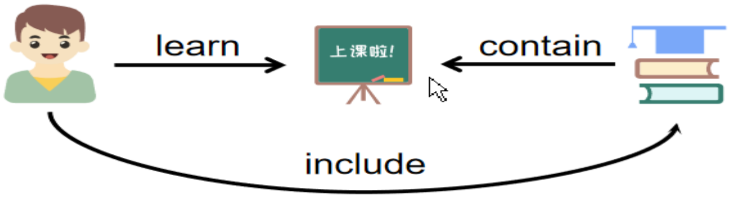


Figure 2. Network schema of HIN in XuetangX.

A MOOCs Network is a typical heterogeneous network, containing objects from six types of entities/objects: Courses ($\mathcal{C}$), teachers ($\mathcal{T}$), users (i.e., users) ($\mathcal{U}$), concepts (i.e., reinforcement learning agents/environments) ($\mathcal{K}$), videos ($\mathcal{V}$) and schools ($\mathcal{S}$). For each course $c \in \mathcal{C}$, it has links to a teacher, a set of users, a set of concepts, a video, and a school. For each teacher $\in \mathcal{T}$, it has links to a set of schools and a set of courses. For each concept $\in \mathcal{K}$, it has links to a set of courses and a set of videos. Each video $\in \mathcal{V}$ has links to a course, a set of users, and concepts. For each user $\in \mathcal{U}$, it has links to a set of courses and videos (the latter is decided by the former). Each school $\in \mathcal{S}$ has links to teachers and a set of courses (the latter is decided by the former). Note that these relations define the link types.

Definition 4. *Graph Convolutional Network (GCN) [26]. In our work, GCN is used to aggregate the neighbors of users to learn user embedding. Assuming that H^l is the user feature of* l*th, the feature of* (l+1)*th users is expressed as*

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

where $\tilde{A} = A + I \in \mathbf{R}^{N\times N}$ is the adjacency matrix A *of the graph* G *with added self-connections, $\tilde{D}$ is the degree matrix of $\tilde{A}$ and $W^{(l)} \in \mathbf{R}^{d\times d}$ is a trainable weight matrix.*

3.2. Recommender as a MDP

We consider the concept recommendation problem under the standard RL framework. At each time step t, the agent (recommender) observes a state s_t about the environment and then takes action (concept) according to its policy π, which is usually a mapping from states to action probabilities. One time step later, as a result of its action, the agent receives a numerical reward (rating) r_{t+1} and a new state s_{t+1} from the environment. The goal of the agent is to maximize the cumulative reward it receives over T time steps. According to [27], such an RL task can be mathematically described by an MDP, a tuple $(\mathcal{S};\mathcal{A};\mathcal{P};\mathcal{R})$ defined as follows.

$\mathcal{S}$ is the state space. The state S_t represents the observed preference of user u at time step t. The representation of state is the n-dimensional user representation in the HIN.

$\mathcal{A}$ is the action space. We define $\mathcal{A}$ as the set of all concepts, i.e., $\mathcal{A} = \mathcal{K}$. In each state s_t, an action a_t can be taken from the set of available actions $\mathcal{A}(s_t)$, which is defined recursively: $\mathcal{A}(s_t) = \mathcal{A}(s_{t-1})\backslash\{a_{t-1}\}$ for $t \neq 0$. In other words, the agent is not allowed to choose the concepts that have been recommended at previous time steps.

$\mathcal{P}$ is the transition function. $\mathcal{P}^a_{ss'} = Pr[s_{t+1} = s^{'}|s_t = s, a_t = a]$ denotes the probability that the environment transits to state $s^{'}$ after receiving action a in state s. In the recommendation setting, the exact transition probabilities are unknown in advance. The agent can observe specific state transitions by interacting with the environment step by step.

$\mathcal{R}$ is the reward function. $\mathcal{R}^a_{ss'} = \mathbb{E}[r_{t+1}|s_t = s, a_t = a, s_{t+1} = s^{'}]$ denotes the expected immediate reward the environment generates after the transition from state s to $s^{'}$ due to action a. In the recommendation setting, the immediate reward of executing an action a only depends on the rating given by user u. Therefore, we define $\mathcal{R}^a_{ss'} = \mathcal{R}_{ua}$.

3.3. Notations and Explanations

The notations used throughout the paper are summarized in Table 1.

Table 1. Notations and Explanations.

Notation	Explanation
G	the heterogeneous information network
V	the set of vertexs
E	the set of edges
S	the network schema
U	the set of users
C	the set of courses
K	the set of concepts
s	environmental status
a	action
r	reward
γ	discount rate
τ	the trajectory of completing an episode
$\mathrm{P}(\mathrm{s_i})$	probability of being in an environmental state
$\mathrm{P}_\theta(\mathrm{T})$	the probability that the trajectory is τ
$\mathrm{R}(\mathrm{T})$	total reward
$\overline{R_\theta}$	total reward expectations
L_{RL}	value function
$\pi_\theta(a_i \mid s_i)$	policy function
$\mathrm{P}(\mathrm{s_{i+1}} \mid \mathrm{s_i}, \mathrm{a_i})$	state transition probability

4. Materials and Methods

In this section, we introduce our proposed methodology in detail. We first give an overview of the whole framework. Then we present our heterogeneous information network for user embedding. Consequently, we introduce our reinforcement learning approach to recommend concepts to users. We also provide some training techniques to train the whole network.

4.1. An Overview of AGMKRec

Figure 3 shows an overview of the network architecture of our proposed model AGMKRec. This framework mainly consists of two components. (a) Meta-path-based user embedding (Section 4.2). This part has two components: One is the meta-path generation (MG) layer shown in Figure 4, the other is node representation. We use the MG layer to generate a new meta-path graph and employ GCN to aggregate node features in an end-to-end fashion. (b) Reinforcement learning for concept recommendation (Section 4.3). In this part, we integrate the embedding network and feed user embedding into the reinforcement learning model to complete the concept recommendation task. We will elaborate on them in the following subsections.

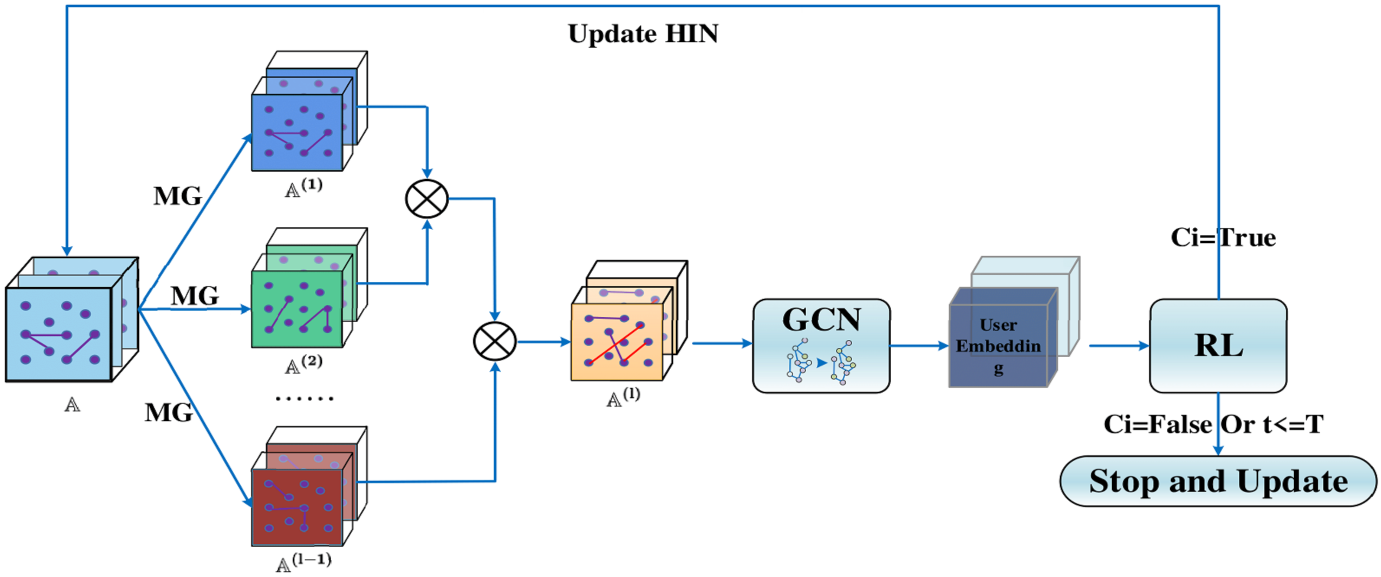


Figure 3. An overview of our proposed model. It has two section: one is the Meta-path-based User Embedding, the other is the Reinforcement Learning for Concept Recommendation.

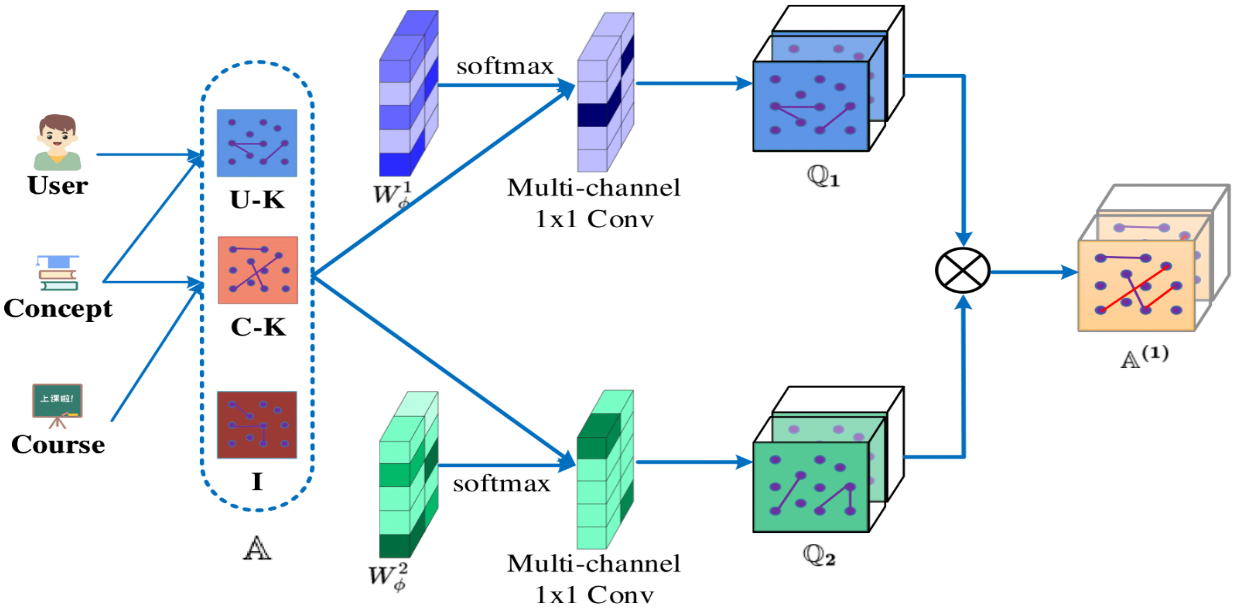


Figure 4. Meta-path generation (MG) layer.

4.2. Meta-Path-Based User Embedding

4.2.1. Meta-Path Generation (MG) Layer

Existing studies [28–31] on heterogeneous graph representation learning are based on the predefined meta-paths, which convert the heterogeneous graph into a homogeneous graph according to the meta-paths to learn the representation of nodes on the heterogeneous graph. While this method has achieved good results, building effective meta-paths manually needs to have specific domain knowledge or experience.

However, our proposed model can automatically identify valid meta-paths to learn the representation of nodes and improve the accuracy of the downstream concept recommendation task. As shown in Figure 4, through the MG layer, we select two of the adjacency matrices Q_1 and Q_2 from the set of adjacency matrices A and get a new meta-path graph, namely meta-path adjacency matrix. This computes the convex combination of adjacency matrices as $\sum_{t_l \in \mathcal{T}^e} \alpha_{t_l}^{(l)} A_{t_l}$ by 1×1 convolution as in Figure 4 with the weights from the softmax function as

$$Q = F(\mathbb{A}; W_\phi) = \phi(\mathbb{A}; \text{softmax}(W_\phi)) \tag{2}$$

where ϕ is the convolution layer and $W_\phi \in \mathbf{R}^{1\times1\times K}$ is the parameter of ϕ. Finally, the meta-path adjacency matrix is computed by matrix multiplication, $A^{(1)} = Q_1 Q_2$. For numerical stability, we use its degree matrix $A^{(l)} = D^{-1} Q_1 Q_2$ to normalize it. In addition, the adjacency matrix of arbitrary length l meta-paths can be calculated by

$$A_p = \prod_{i=1}^{l} \sum_{t_i \in T^e} \alpha_{t_i}^{(i)} A_{t_i} \tag{3}$$

where A_P denotes the adjacency matrix of meta-paths, T^e denotes a set of edge types and $\alpha_{t_l}^{(l)}$ is the weight for edge type t_l at the lth MG layer. When α is not one-hot vector, A_P can be considered as the weighted sum of all length-l meta-path adjacency matrices. However, one problem is that as the MG layer increases, so does the length of the meta-path. It cannot satisfy the application scenarios where long paths are as important as short paths. So we add the identity matrix I to A, i.e., A_4= I. This trick allows us to learn any length of meta-paths up to l + 1 when l MG layers are stacked.

4.2.2. Node Representation

To consider multiple types of meta-paths simultaneously, the output channels of 1×1 convolution in Figure 4 are set to C. Then, the MG layer yields a set of meta-paths and the intermediate adjacency matrices Q_1 and $Q_2 \in \mathbf{R}^{N\times N\times C}$. It is beneficial to learn different node representations via multiple different graph structures. As shown in Figure 3, after the stack of l MG layers, a GCN is applied to each channel of meta-path tensor $A^{(l)} \in \mathbf{R}^{N\times N\times C}$ and multiple node representations are concatenated as

$$Z = \|_{i=1}^{C} \sigma\left(\tilde{D}_i^{-1} \tilde{A}_i^{(l)} XW\right) \tag{4}$$

where | | is the concatenation operator, C denotes the number of channels, $\tilde{A}_i^{(l)} = A_i^{(l)}$ is the adjacency matrix from the ith channel of $\mathbb{A}(l)$, $\tilde{D}_i$ is the degree matrix of $\tilde{A}_i^{(l)}$, $W \in \mathbf{R}^{d\times d}$ is a trainable weight matrix shared across channels and $X \in \mathbf{R}^{N\times d}$ is a feature matrix. Z contains the node representations from C different meta-path graphs with variable, at most l + 1, lengths. We concatenate the user embedding obtained under different meta-paths as input to the reinforcement learning framework. This architecture can be viewed as an ensemble of GCNs on multiple meta-path graphs learned by the MG layers.

4.3. Reinforcement Learning for Concept Recommendation

The user interests develop with time, and the behaviors of the recommender systems may have a significant impact on the development of the user interests. In a sense, it guides user interests by displaying specific items and hiding the rest. To achieve dynamic personalized recommendation, the system is required to change with the environment.

Many algorithms related to time series have been proposed to model user preferences in different ways. For example, Hidasi et al. [32] propose to input a temporal sequence of the historical items into the gated recurrent unit (GRU) model and output the last embedding vector as the user preference. However, the model is limited by the assumption that all the historical items play the same role in estimating the similarity between the user profile and the target item. To distinguish the effects of different items, attention-based models such as neural attentive item similarity (NAIS) [33] and neural attentive session-based recommendation (NASR) [34] can be used to estimate an attention coefficient for each historical item as its importance in recommending the target item. While the existing attention-based model improves the recommendation performance, it still poses an unresolved challenge. When a user selects diverse items, the effects of the historical items that indeed reflect the user's interests in the target item will be diluted by many irrelevant items.

Reinforcement learning is a mechanism to learn how to map state to action to get the maximum reward [35,36]. Therefore, the recommendation strategy based on reinforcement learning is more conducive to taking the long-term interests of users and the impact of dynamic embedding into consideration [37].

In Deep Q-Learning Network (DQN) [38,39], the loss function comes from the difference between target Q and prediction $\widehat{Q}$. The target Q is used as the correct label guide in the algorithm [40,41]. In our scenario, if the recommendation concept is wrong, the set of adjacency matrices A will not be changed, and the user embedding will not be changed. Then, the predicted Q_{t+1} will be identical to the target Q_t. However, the Q-learning network needs the next step transition s_{t+1} of the current step s_t to update. The algorithm based on Q-learning cannot meet our needs. In this paper, the policy gradient is used for reinforcement learning, realizing the maximization of value function faster [42]. Figure 5 shows a policy gradient model. Since there is no correct label guidance in the policy gradient, we use the expected return to optimize the policy function.

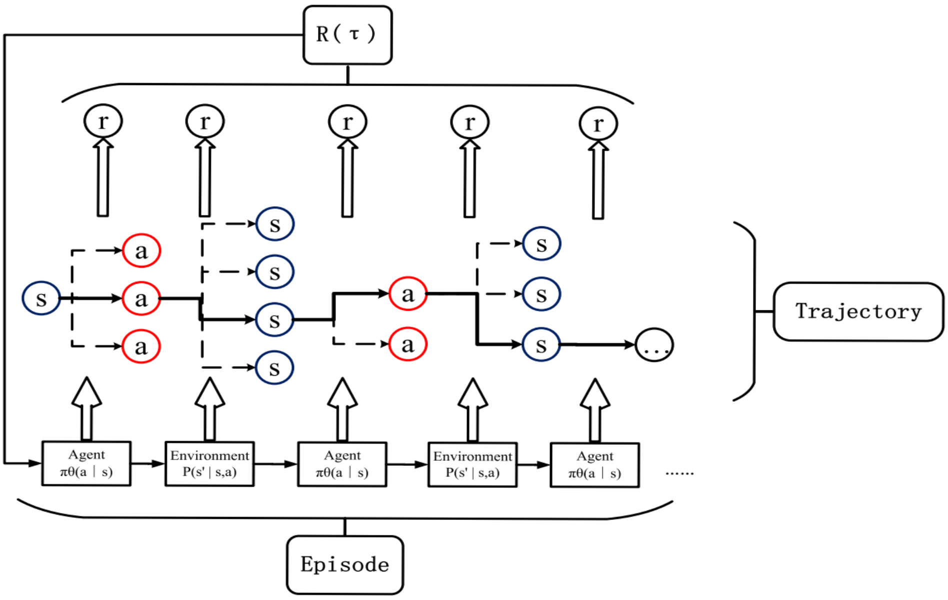


Figure 5. Policy Gradient Model.

The probability that the trajectory of completing an episode is τ:

$$\mathrm{P}_{\theta}(\tau) = \prod_{\mathrm{t}=1}^{\mathrm{T}} [\mathrm{P}(\mathrm{s}_{\mathrm{i}+1} \mid \mathrm{s}_{\mathrm{i}}, \mathrm{a}_{\mathrm{i}}) \pi_{\theta}(\mathrm{a}_{\mathrm{i}} \mid \mathrm{s}_{\mathrm{i}})] \tag{5}$$

The formula of total rewards for completing the trajectory is as follows:

$$\mathrm{R}(\varnothing) = \sum_{\mathrm{t}=1}^{\mathrm{T}} \gamma^{\mathrm{t}-1} \mathrm{r}_{\mathrm{t}} \tag{6}$$

Given the particular user if the predicted concept $\hat{s}$ is confirmed, our model will update HIN by adding a link between the user and the correct recommended concept. Furthermore, the reward will be set as 1. Otherwise, it will be -1. The average reward for all episodes is:

$$\overline{\mathrm{R}_{\theta}} = \sum_{\tau} \mathrm{R}(\tau) \mathrm{P}_{\theta}(\tau) \approx \frac{1}{\mathrm{N}} \sum_{\mathrm{n}=1}^{\mathrm{N}} \mathrm{R}(\varnothing) \tag{7}$$

Because the goal of the policy gradient is to maximize the value function, the gradient rising method is used to increase the value function. The gradient strategy is divided into Monte Carlo MC and time-series differential TD. In the Monte Carlo algorithm, an episode is learned once, while in the time series differential algorithm, each step is updated once. In this paper, the policy gradient is accustomed to learning by reinforcement of the algorithm after each episode. According to the policy gradient theorem [43,44], the gradient of expected cumulative rewards can be calculated by the formula:

$$\begin{aligned}\nabla L_{RL}(\theta) &= \sum_{\tau} R(\o)\nabla P_{\theta}(\tau)\\ &= \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T} R(\tau)\nabla \log[\pi_{\theta}(a_t \mid s_t)]\end{aligned} \tag{8}$$

In this paper, entropy regularization is used to alleviate the special problem of trade off between exploration and utilization in reinforcement learning.

$$H[\pi_{\theta}(a_t \mid s_t)] = \sum_{t=1}^{T}\sum_{c_t \in C} \pi_{\theta}(a_t \mid s_t)\log[\pi_{\theta}(a_t \mid s_t)] \tag{9}$$

The final objective is:

$$\mathbb{E}_{c\sim\pi_{\theta}(a_t|s_t)} L_{RL}(\theta) + \lambda H[\pi_{\theta}(a_t \mid s_t)] \tag{10}$$

where λ is the weight of the regularization method. The algorithm process of AGMKRec is summarized in Algorithm 1.

5. Experiments

In this part, We present the details and results of all the experiments. Our experiments are designed to answer the following questions:

RQ1: Does our proposed model outperform state-of-the-art baselines?

RQ2: Are the new meta-path graphs generated by the MG layers effective for learning node representations?

5.1. Experimental Dataset

We conduct a comprehensive statistical analysis on a real-world dataset from XuetangX (Available online: http://www.xuetangx.com, accessed on 5 July 2020). This dataset records the enrolled behaviors from 1 January 2015 to 31 December 2019. It consists of 2527 concepts, 3,708,461 users, 7327 courses, 96,950 videos and 140,446,950 relations among them. The detailed statistics of the XuetangX MOOCs datasets are shown in Table 2.

We select the enrolled behaviors from 1 October 2016 to 30 December 2017 as the training set, and those from 1 January 2018 to 31 March 2018, as the test set. Each instance in the training or the test set is a sequence representing a user's history of click behaviors. During the training process, for each sequence in the training data, we hold out the last clicked concept as the target. Furthermore, the rest is treated as historical clicked concepts. For each positive instance, we randomly sample X concepts that a user has never interacted with before as negative instances. In our experiments, we set X as 4, an empirical number that has shown good performance in experiments [33]. During the test process, we treat each concept in the test set as the target concept and the corresponding concepts of the same user in the training set as the historical clicked concepts. Moreover, a user often continuously clicks the same concept label and then generates multiple clicked concept records. These records are treated as a single record. Each positive instance in the test set is paired with 99 randomly sampled negative instances [33].

For our evaluation, we adopt four evaluation metrics that are widely used in the recommendation system, including Hit Ratio of top-K items (HR@K), Normalized Discounted

Cumulative Gain of top-K items (NDCG@K) and Mean Reciprocal Rank (MRR), area under the ROC curve (AUC). In the experiments, we set K to 5, 10 and 20.

Algorithm 1 The overall learning algorithm of AGMKRec.

```
Input:
    Training set $\mathcal{U}_{\text{train}}$ ,
    Feature matrix $X$,
    The set of adjacency matrices $A$,
    Number of episodes $K$,
    Number of time steps $T$,
    Discount rate $\gamma$,
    $\epsilon$-greedy parameter $pi_\theta$.
Output:
    The learned recommender policy $pi_\theta$.
Initialize characteristic matrix and adjacency matrix
Input the characteristic matrix and adjacency matrix into AGMKRec
Multi-channel $1 \times 1$ conv is used to generate $l$ adjacency matrix $Q$
The characteristic matrix is generated by multiplying $l$ adjacency matrices $Q$
The characteristic matrix is aggregated by GCN
Initialize recommender policy $pi_\theta$ with random weights
for $episode = 1 \to K$ do
    Uniformly pick a user $u_0 \in \mathcal{U}_{\text{train}}$ as the environment
    Learning the user embedding u
    Set t=0
    while $c_t$ and t$\leq$T do
        Select action $a_t$ using $\epsilon$-greedy policy w.r.t $\pi_\theta$
        Take $a_t$ observe reward $r_{t+1}$
        Update adjacency matrix and compute user embedding $u_{t+1}$
        Set t=t+1
    end while
    Update weights of $\pi_\theta$
end for
```

Table 2. An overview of XuetangX MOOCs dataset.

Nodes	Count	Links	Count
concept	2527	concept-course	21,507
		concept-video	11,732
user	3,708,461	user-course	15,045,219
course	7327	course-concept	69,012
		course-user	16,724,852
Total	3,718,315	Total	31,872,322

5.2. Evaluation Metrics

HR@K is a commonly used metric to measure the recall rate. The HR@K is defined as

$$HR@K = \frac{\text{\# Hits @ K}}{|GT|} \tag{11}$$

where $|GT|$ denotes the size of the test set. NDCG@K is a position-aware metric that assigns larger weights on higher positions. The NDCG@K is defined as

$$NDCG@K = \frac{1}{|Q|} \sum_{q=1}^{|Q|} Z_{kq} \sum_{j=1}^{k} \frac{2^{r(j)} - 1}{\log(1 + j)} \tag{12}$$

where $r(j)$ denotes the correlation score for each item and Z_{kq} is a normalization factor. MRR is used to evaluate the performance of the retrieval system by the correct ranking of the retrieval results in the retrieval results. It is defined as

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{13}$$

where $|Q|$ is the size of the query set.

5.3. Baselines

To verify the effectiveness of our proposed model, we compare AGMKRec with the method based on HIN and some other baselines. The comparison methods include:

- Matrix Factorization (MF): MF has the characteristics of collaborative filtering, hidden semantic analysis, and supervised learning, coupled with easy implementation and high expansibility. It has become a very classical algorithm in the field of recommendation.
- Bayesian Personalized Ranking (BPR): BPR is a personalized ranking algorithm based on matrix factorization. It does not optimize the global score but optimizes the ranking according to each user's item preferences.
- Mutiple Layer Perception (MLP): MLP is a forward-structured artificial neural network that maps a set of input vectors to a set of output vectors.
- Factor Item Similarity Models (FISM): FISM is essentially an item-based collaborative filtering algorithm. To solve the problem of sparse datasets, FISM uses the mapping of item vectors to represent user vectors, which greatly improves the use of information.
- Neural Attentive Item Similarity (NAIS): NAIS adds attention network to traditional item-based collaborative filtering.
- Neural Attentive Session-based Recommendation (NASR): NASR is a session-based recommendation algorithm that takes into account the sequential behaviors and main intentions of users in the current session.
- Heterogeneous Information Network Embedding for Recommendation (HERec): HERec is a traditional heterogeneous model that learns node representations by applying DeepWalk to predefined meta-paths.
- AGMKRec-SL: AGMKRec-SL represents that we only learn user embedding and use supervised learning to complete recommendation tasks without reinforcement learning.

5.4. Implementation Details

For the proposed AGMKRec, we train our model on an Nvidia GeForce GTX2080Ti GPU card with 11GB RAM and implement AGMKRec with Tensorflow. We use 80% of the dataset as the training set, 10% as the validation set, and the remaining 10% as the test set. The number of MG layers is set to 3, and the parameters of 1×1 convolution are initialized with a constant value. Considering the memory overhead, we only use two channels. We set the final user embedding dimension to 64. To improve the effectiveness of the model, we use supervised learning to pre-train our model for 12,000 episodes and then use reinforcement learning to train it. Finally, we optimize the model with Adam.

5.5. Experimental Results

As shown in Table 3, the recommendation performance of all methods on MOOCs (RQ1). HIN-based methods (HERec, AGMKRec-SL, AGMKRec) perform better than traditional recommendation methods, which indicate heterogeneous information is helpful. Compared with HERec, our model gets better results which demonstrate the meta-paths that our model generates better information extraction than predefined meta-paths. Table 4 shows that MG layers can not only generate the same meta-path as the predefined meta-path between target nodes (the first node in the meta-path is of the same type as the last node), but also find other essential meta-paths that are helpful for node representations (RQ2).

What is more, to interpret the meta-path graph learned by the MG layers, we visualize the attention score of the adjacency matrix (edge type) in Figure 6. We find that the MG layer sticks to the shorter meta-path by assigning the higher attention store to the identity matrix.

Table 3. Recommendation performance of all methods (%).

Methods	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20	MRR	AUC
FM	43.29	59.87	76.25	33.92	36.78	36.09	31.22	85.64
BPR	36.58	61.6	78.03	33.12	38.01	41.72	32.13	86.42
MLP	44.48	62.64	76.62	31.35	34.84	36.11	28.37	84.05
FISM	55.61	70.87	75.31	38.8	41.51	43.56	32.75	85.35
NAIS	43.77	67.65	84.17	23.77	32.92	37.63	29.4	87.31
NASR	44.51	65.82	75.29	23.02	31.66	39.42	27.88	83.33
HERec	53.26	70.37	80.1	33.35	39.64	45.11	32.36	87.52
AGMKRec-SL	60.62	73.32	**88.74**	37.26	43.25	47.55	35.21	**88.2**
AGMKRec	**61.57**	**76.85**	87.53	**40.6**	**45.28**	**49.88**	**37.91**	87.76

Table 4. Predefined meta-paths and generated meta-paths by the MG layers.

Predefined Meta-Path	**Meta-Path Learnt by the MG Layers** **Top 3 (between Target Nodes)**	**Top 3 (All)**
UKU	UKU	UKU
UKCKU	UKCKU	UKCKU
	UKUKU	UKC

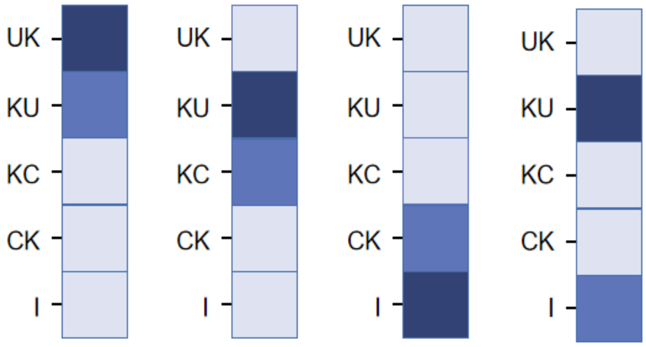


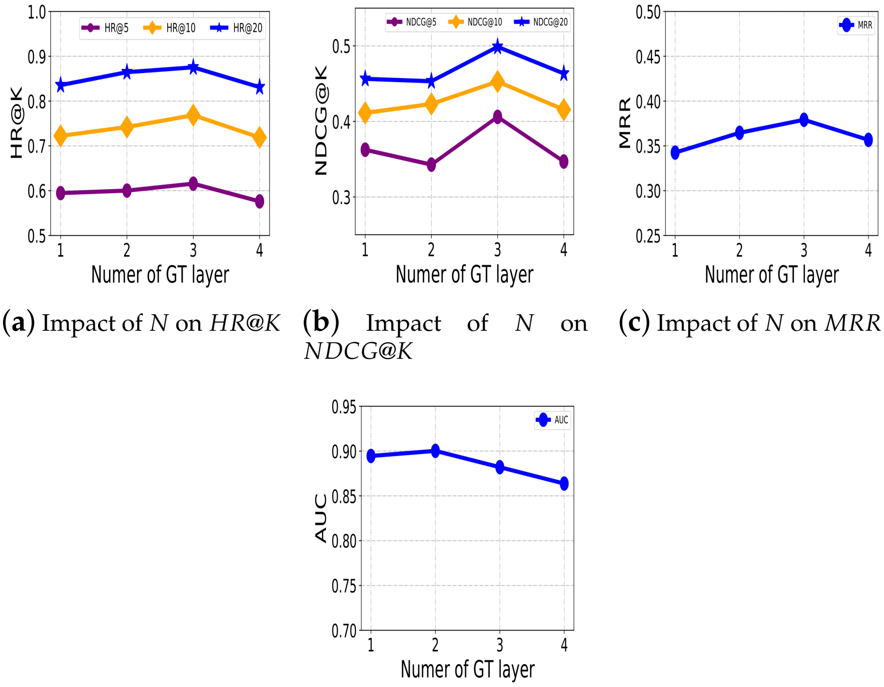
Figure 6. After applying the softmax function on 1×1 Conv fillter W^i_ϕ (i: Index of the MG layer) in Figure 4, we visualize this attention score of adjacency matrix (edge type) in XuetangX. Each edge indicates (User-Concept), (Concept-User), (Concept-Course), (Course-Concept) and identity matrix. KU, KC is the transpose of UK and CK, respectively. The darker the color, the higher the attention score.

5.6. Parameters Analysis

To get the best experimental result, we conduct many experiments by tuning three model parameters, including the number of MG layers, user embedding dimension and regularization rate.

5.6.1. Impact of MG Layer in HIN Embedding

Using the different number of MG layers can learn meta-paths of different lengths. To select an appropriate value, we have carried out four groups of experiments, setting the number of layers as 1, 2, 3, 4, respectively. As shown in Figure 7, we find that HR@K, NDCG@K, and MRR achieve better results when the value is 3. While the number of MG layers and the training time of the model are increased, the stability of the model is improved, and a better effect is achieved.



(**a**) Impact of N on $HR@K$ (**b**) Impact of N on $NDCG@K$ (**c**) Impact of N on MRR

(**d**) Impact of N on AUC

Figure 7. Parameter sensitivity of AGMKRec over the MG layer N.

5.6.2. Impact of Embedding Dimension in HIN Embedding

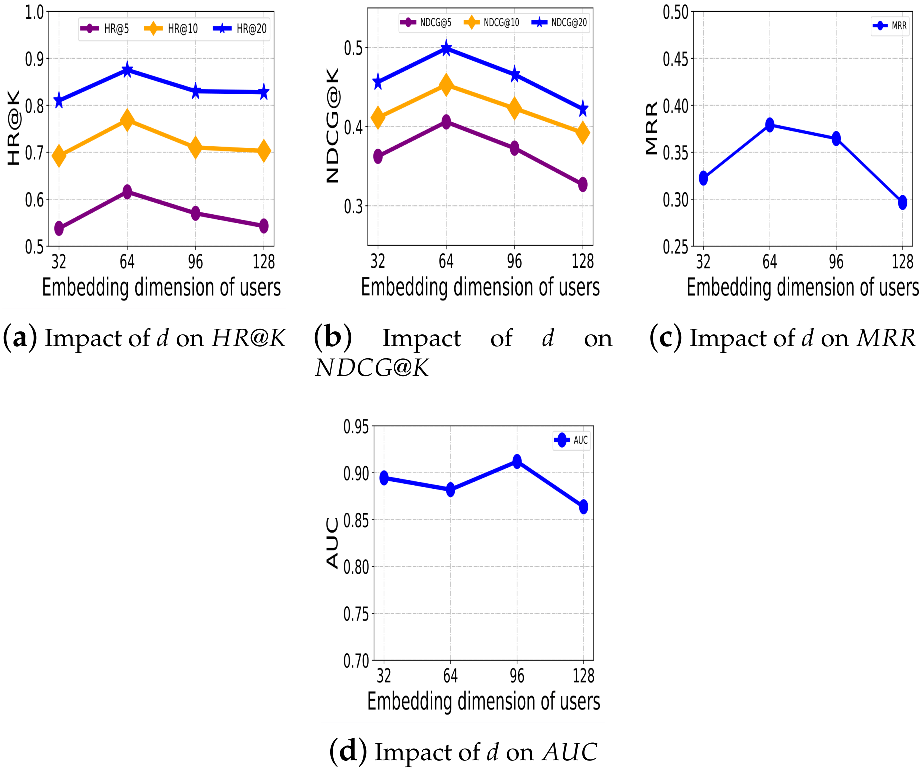
As the input of the RL framework, user embedding directly affects the final recommendation effect of the model. To analyze the effect of different user embedding dimensions, we use different embedding dimensions 32, 64, 96, and 128 to conduct the experiment. In Figure 8, we observe that different user embedding dimensions significantly influence evaluation metrics and 64 is the optimal embedding dimension.

5.6.3. Impact of Regularization Rate λ

λ is the parameter of the RL framework. We conduct more elaborate experiments on it. We first set λ from 0.01, 0.1, 1 to 10 and present their experimental results in the four bigger figures of Figure 9. Then, for further analysis, we set λ from 0.01 to 0.1 (i.e., 0.02, 0.04, 0.06, and 0.08) and present the results in the four smaller figures, each of which is contained in the corresponding bigger one, respectively. From Figure 9, we can find that the value of λ from 0.01 to 0.1 can benefit our AGMKRec model to achieve better performance on concept recommendation. Otherwise, the model will gain a lower performance.

5.7. Case Study

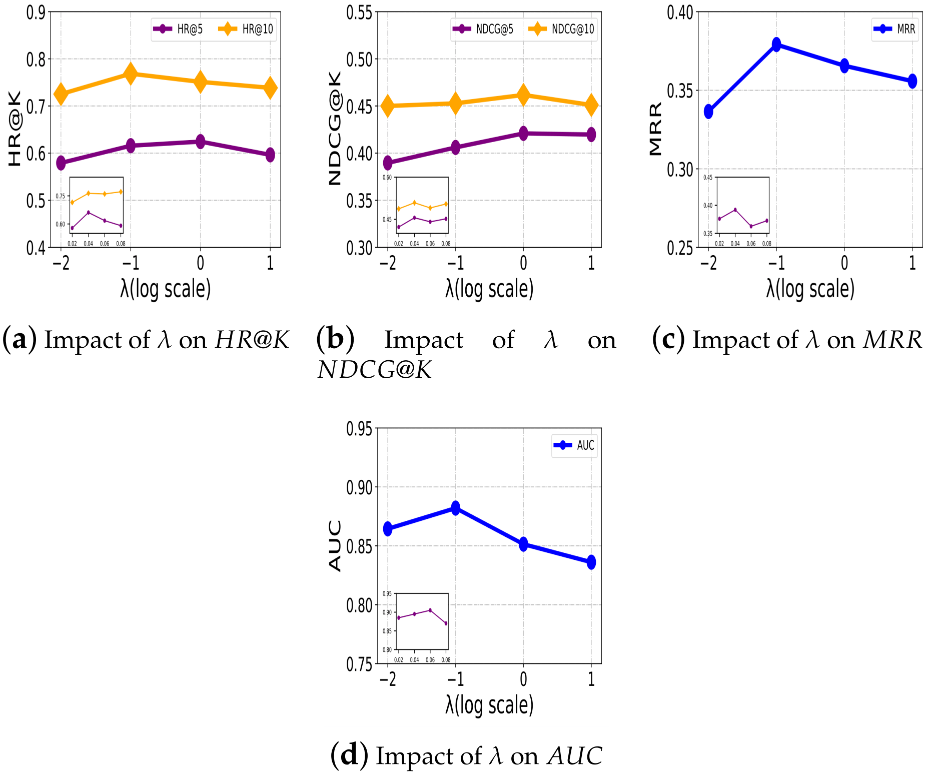
We conduct case studies to demonstrate the effectiveness of our proposed method, especially on the ability of adapting to the dynamic environment by recommending personalized concepts, as shown in Figure 10. We first randomly select two users, i.e., user A with ID 3203335 and user B with ID 7796703, from our test set. Given the clicked history of both users, Figure 10a shows the recommended concept lists to these two users, respectively. From this figure, we can find that user A and user B are from different backgrounds and have different clicked behaviors in the MOOCs. For example, user A is learning a course named “The Fundamentals of Psychology”, while user B is learning a course named “Character Analysis”. These two users have clicked different knowledge concepts related to their courses. While both of them have clicked the “Emotion Analysis”, our model can adaptively consider the instant clicked information, and dynamically recommend the next concept the user may be interested in. To prevent occasionality, we do the same experiment on users C and D and get the same conclusion, as shown in Figure 10b.



(**a**) Impact of *d* on *HR@K* (**b**) Impact of *d* on *NDCG@K* (**c**) Impact of *d* on *MRR*

(**d**) Impact of *d* on *AUC*

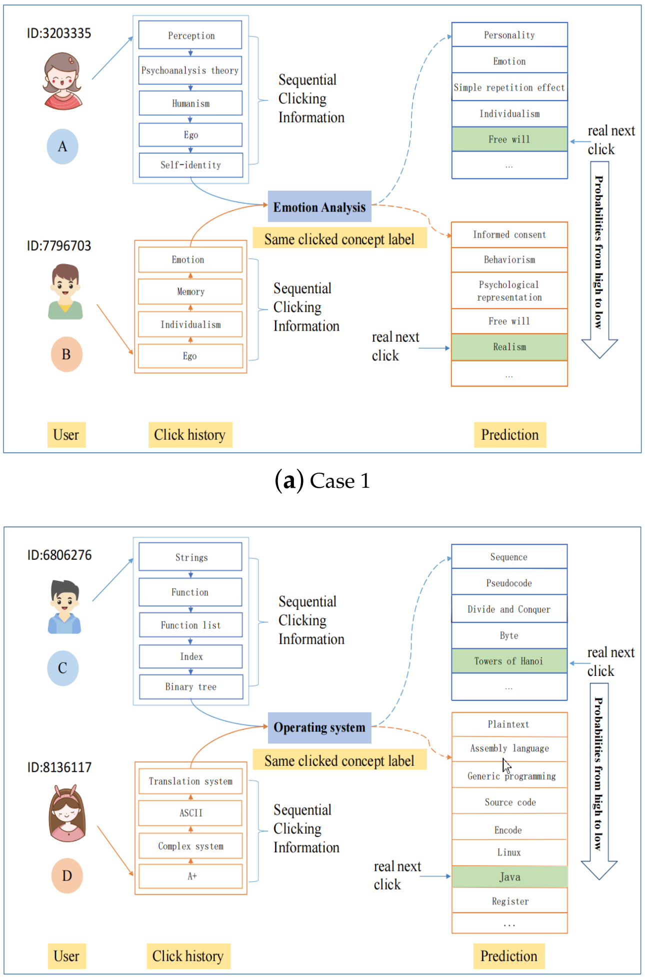
Figure 8. Parameter sensitivity of AGMKRec over embedding dimension *d*.



(**a**) Impact of λ on *HR@K* (**b**) Impact of λ on *NDCG@K* (**c**) Impact of λ on *MRR*

(**d**) Impact of λ on *AUC*

Figure 9. Parameter sensitivity of AGMKRec over regularization rate λ.



(**a**) Case 1

(**b**) Case 2

Figure 10. Diversity recommendation case with sequential clicked concepts information in a dynamic learning environment of XuetangX MOOCs.

6. Conclusions

In this work, we present AGMKRec, a novel reinforced concept recommendation model with a heterogeneous information network. It can automatically identify effective meta-paths and multi-hop connections to represent users in the HIN of MOOCs and incorporate user embedding into the reinforcement learning framework. Our approach converts a heterogeneous graph into multiple new meta-path graphs to learn user representations. To improve the dynamic nature of user preferences, we use the reinforcement method, which can consider current reward and future reward simultaneously to generate the diversified recommendation list. The experiment results on MOOCs show that our approach outperforms some state-of-the-art methods. We will try our proposed model in other domains. For example, people usually want to watch different types of movies and listen to different styles of music.

Author Contributions: Conceptualization, C.W.; methodology, C.W.; software, J.G.; validation, J.G., Z.Z. and X.Z.; formal analysis, C.W.; investigation, Z.Z.; resources, J.G.; data curation, J.G.; writing—original draft preparation, C.W.; writing—review and editing, Z.Z.; visualization, Z.Z.; supervision, J.G.; project administration, J.G.; funding acquisition, J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AGMKRec	Automatic Generation of Meta-path Graph for Concept Recommendation
HIN	Heterogeneous Information Network
RL	Reinforcement Learning
GCN	Graph Convolutional Network
MG	Meta-path Generation
MDP	Markov Decision Process
DQN	Deep Q-Learning Network
CF	Collaborative Filtering

References

1. Pan, L.; Li, C.; Li, J.; Tang, J. Prerequisite relation learning for concepts in moocs. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1447–1456.
2. Pan, L.; Wang, X.; Li, C.; Li, J.; Tang, J. Course concept extraction in moocs via embedding-based graph propagation. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Taipei, Taiwan, 27 November–1 December 2017; pp. 875–884.
3. Qiu, J.; Tang, J.; Liu, T.X.; Gong, J.; Zhang, C.; Zhang, Q.; Xue, Y. Modeling and predicting learning behavior in moocs. In Proceedings of the ninth ACM international conference on web search and data mining, San Francisco, CA, USA, 22–25 February 2016; pp. 93–102.
4. Jing, X.; Tang, J. Guess you like: Course recommendation in moocs. In Proceedings of the International Conference on Web Intelligence, Amantea, Italy, 23–26 August 2017; pp. 783–789.
5. Zhang, J.; Hao, B.; Chen, B.; Li, C.; Chen, H.; Sun, J. Hierarchical reinforcement learning for course recommendation in moocs. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019. [CrossRef]
6. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.-S. Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 173–182.
7. Almaghrabi, M.; Chetty, G. Multilingual sentiment recommendation system based on multilayer convolutional neural networks (mcnn) and collaborative filtering based multistage deep neural network models (cfmdnn). In Proceedings of the 2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA), Antalya, Turkey, 2–5 November 2020.
8. Zhao, X.; Zhang, L.; Xia, L.; Ding, Z.; Yin, D.; Tang, J. Deep reinforcement learning for list-wise recommendations. *arXiv* **2017**, arXiv:1801.00209.
9. Parameswaran, A.; Venetis, P.; Garcia-Molina, H. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Trans. Inf. Syst. (TOIS)* **2011**, *29*, 1–33. [CrossRef]
10. Parameswaran, A.G.; Garcia-Molina, H.; Ullman, J.D. Evaluating, combining and generalizing recommendations with prerequisites. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, Toronto, ON, Canada, 26–30 October 2010; pp. 919–928.
11. Ma, H.; Zhou, D.; Liu, C.; Lyu, M.R.; King, I. Recommender systems with social regularization. In Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, Hong Kong, China, 9–12 February 2011; pp. 287–296.
12. Kouris, P.; Varlamis, I.; Alexandridis, G. A package recommendation framework based on collaborative filtering and preference score maximization. In *International Conference on Engineering Applications of Neural Networks*; Springer: Athens, Greece, 2017; pp. 477–489.
13. Kouris, P.; Varlamis, I.; Alexandridis, G.; Stafylopatis, A. A versatile package recommendation framework aiming at preference score maximization. *Evol. Syst.* **2018**, *11*, 1–19. [CrossRef]
14. Hu, B.; Shi, C.; Zhao, W.X.; Yang, T. Local and global information fusion for top-n recommendation in heterogeneous information network. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 1683–1686.
15. Cen, Y.; Zou, X.; Zhang, J.; Yang, H.; Zhou, J.; Tang, J. Representation learning for attributed multiplex heterogeneous network. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1358–1368.
16. Sun, Y.; Han, J.; Yan, X.; Yu, P.S.; Wu, T. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proc. Vldb Endow.* **2011**, *4*, 992–1003. [CrossRef]
17. Wang, X.; Lu, Y.; Shi, C.; Wang, R.; Mou, S. Dynamic heterogeneous information network embedding with meta-path based proximity. *IEEE Trans. Knowl. Data Eng.* **2020**, *99*, 1. [CrossRef]
18. Yun, S.; Jeong, M.; Kim, R.; Kang, J.; Kim, H.J. Graph transformer networks. *arXiv* **2019**, arXiv:1911.06455.
19. Zhou, Q. A novel movies recommendation algorithm based on reinforcement learning with ddpg policy. *Int. J. Intell. Comput. Cybern.* **2020**, ahead-of-print. [CrossRef]

20. Zheng, G.; Zhang, F.; Zheng, Z.; Xiang, Y.; Yuan, N.J.; Xie, X.; Li, Z. Drn: A deep reinforcement learning framework for news recommendation. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 167–176.
21. Dm, A.; Xl, A.; Qd, A.; Dha, B. Humming-query and reinforcement-learning based modeling approach for personalized music recommendation. *Procedia Comput. Sci.* **2020**, *176*, 2154–2163.
22. Zhao, X.; Xia, L.; Zhang, L.; Ding, Z.; Yin, D.; Tang, J. Deep reinforcement learning for page-wise recommendations. In Proceedings of the 12th ACM Conference on Recommender Systems, Vancouver, BC, Canada, 2–7 October 2018; pp. 95–103.
23. Pan, F.; Cai, Q.; Tang, P.; Zhuang, F.; He, Q. Policy gradients for contextual recommendations. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 1421–1431.
24. Chen, X.; Li, S.; Li, H.; Jiang, S.; Song, L. Neural model-based reinforcement learning for recommendation. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019.
25. Sun, Y.; Han, J. Mining heterogeneous information networks: Principles and methodologies. *Synth. Lect. Data Min. Knowl. Discov.* **2012**, *3*, 1–159. [CrossRef]
26. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
27. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
28. Gong, J.; Wang, S.; Wang, J.; Feng, W.; Yu, P.S. Attentional graph convolutional networks for knowledge concept recommendation in moocs in a heterogeneous view. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Xi'an, China, 25–30 July 2020; pp. 79–88.
29. Shi, C.; Hu, B.; Zhao, W.X.; Philip, S.Y. Heterogeneous information network embedding for recommendation. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 357–370. [CrossRef]
30. Dong, Y.; Chawla, N.V.; Swami, A. metapath2vec: Scalable representation learning for heterogeneous networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 135–144.
31. Fu, T.-y.; Lee, W.-C.; Lei, Z. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6 November 2017; pp. 1797–1806.
32. Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; Tikk, D. Session-based recommendations with recurrent neural networks. *arXiv* **2015**, arXiv:1511.06939.
33. Chua, T.S.; He, X.; He, Z.; Song, J.; Liu, Z.; Jiang, Y.G. Nais: Neural attentive item similarity model for recommendation. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 2354–2366.
34. Zhang, T.; Huang, M.; Zhao, L. Learning structured representation for text classification via reinforcement learning. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
35. Zhu, S.; Ng, I.; Chen, Z. Causal discovery with reinforcement learning. *arXiv* **2019**, arXiv:1906.04477.
36. Yang, Y.; Zhang, G.; Xu, Z.; Katabi, D. Harnessing structures for value-based planning and reinforcement learning. *arXiv* **2019**, arXiv:1909.12255.
37. Zou, L.; Xia, L.; Ding, Z.; Song, J.; Liu, W.; Yin, D. Reinforcement learning to optimize long-term user engagement in recommender systems. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2810–2818.
38. Fakoor, R.; Chaudhari, P.; Soatto, S.; Smola, A.J. Meta-q-learning. *arXiv* **2019**, arXiv:1910.00125.
39. Hester, T.; Vecerik, M.; Pietquin, O.; Lanctot, M.; Schaul, T.; Piot, B.; Horgan, D.; Quan, J.; Sendonaris, A.; Dulac-Arnold, G. Deep q-learning from demonstrations. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2017.
40. Jing, P.; Williams, R.J. Incremental multi-step q-learning. *Mach. Learn.* **1998**, *22*, 226–232.
41. Galkin, B.; Fonseca, E.; Amer, R.; Dasilva, L.A.; Dusparic, I. Reqiba: Regression and deep q-learning for intelligent uav cellular user to base station association. *arXiv* **2020**, arXiv:2010.01126.
42. Sutton, R.S.; Mcallester, D.; Singh, S.; Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. *Submitt. Adv. Neural Inf. Process. Syst.* **1999**, *12*, 1057–1063.
43. Xu, P.; Gao, F.; Gu, Q. Sample efficient policy gradient methods with recursive variance reduction. *arXiv* **2019**, arXiv:1909.08610.
44. Ilyas, A.; Engstrom, L.; Santurkar, S.; Tsipras, D.; Janoos, F.; Rudolph, L.; Madry, A. A closer look at deep policy gradients. *arXiv* **2018**, arXiv:1811.02553.