Recommending Knowledge Concepts on MOOC Platforms with Meta-path-based Representation Learning

Guangyuan Piao
Department of Computer Science, Hamilton Institute
Maynooth University
Maynooth, Co Kildare, Ireland
guangyuan.piao@mu.ie

ABSTRACT

Massive Open Online Courses (MOOCs) which enable large-scale open online learning for massive users have been playing an important role in modern education for both students as well as professionals. To keep users' interest in MOOCs, recommender systems have been studied and deployed to recommend courses or videos that a user might be interested in. However, recommending courses and videos which usually cover a wide range of knowledge concepts does not consider user interests or learning needs regarding some specific concepts. This paper focuses on the task of recommending knowledge concepts of interest to users, which is challenging due to the sparsity of user-concept interactions given a large number of concepts. In this paper, we propose an approach by modeling information on MOOC platforms (e.g., teacher, video, course, and school) as a Heterogeneous Information Network (HIN) to learn user and concept representations using Graph Convolutional Networks based on user-user and concept-concept relationships via meta-paths in the HIN. We incorporate those learned user and concept representations into an extended matrix factorization framework to predict the preference of concepts for each user. Our experiments on a real-world MOOC dataset show that the proposed approach outperforms several baselines and state-of-the-art methods for predicting and recommending concepts of interest to users.

Keywords

User Modeling, MOOC, Learning Analytics, Knowledge Concept, Recommender Systems

1. INTRODUCTION

MOOCs (Massive Open Online Courses), which are free online courses available to anyone to enroll around the world, have gained a lot of popularity in the past decade. By the end of 2018, popular MOOC platforms such as edX[1], and Coursera[2] have provided 11,400 courses with 101 million users/learners on those platforms[3]. Previous studies have shown that MOOCs do have a real impact [24, 8]. For example, Chen et al. [8] showed that 72% of survey respondents reported career benefits and 61% reported educational benefits. Despite of the popularity, one main challenge of MOOCs is the overall completion rate of those courses is normally lower than 10% [19, 30]. Therefore, understanding and predicting user behaviors and learning needs are important to keep users learning on MOOC platforms.

To this end, previous studies have focused on understanding dropout or procrastination behavior [28, 12, 38, 14] and recommending content such as courses and learning paths that a user might be interested in [16, 26, 4]. A MOOC can be seen as a sequence of videos where each video is associated with some knowledge concepts. For example, a video in a computer science MOOC can cover several concepts such as "software" and "hardware". More recently, Gong et al. [13] argued that course or video recommendations overlook user interests regarding specific knowledge concepts. For example, data mining courses taught by different teachers can be quite different in a microscopic view, and a user who is interested in some specific concepts such as "association rules" might be interested in various video clips or learning materials from different teachers covering those concepts from different perspectives. Therefore, understanding a user's learning needs from a microscopic view and predicting knowledge concepts that the user might be interested in are important.

In this work, we focus on predicting and recommending knowledge concepts that might be interesting to users on MOOC platforms. Based on the interaction history between users and concepts (i.e., a user has interacted with a concept if the user has learned that concept), traditional recommendation approaches such as collaborative filtering (CF) — which recommends similar items (concepts) based on a user's interaction history or interesting items from similar users — can be applied. However, the sparsity of user-item (user–concept) relationships can limit the performance of CF-based methods. In addition to users and concepts, MOOC platform data normally contain other entities such as courses, videos, and teachers as well as the relationships among those entities.

To cope with the sparsity problem, we model those enti-

[1] https://www.edx.org/

[2] https://www.coursera.org/

[3] https://bit.ly/3tScITp

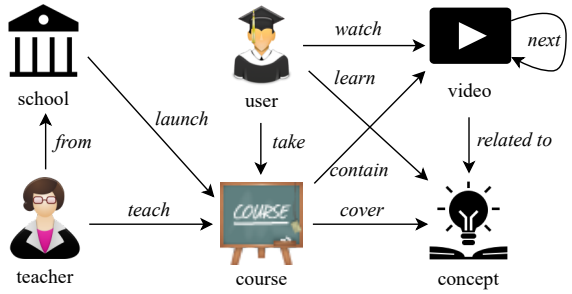

Figure 1: Different types of entities and relationships between two entities in MOOCs. A user u is interested in a concept if u has learned or is going to learn it in the future.

ties and their relationships as a heterogeneous information network (HIN) [33] consisting of the entities and relationships inspired by [13], which can be used for learning user (concept) representations/embeddings by exploring indirect user-user (concept-concept) relationships with Graph Convolutional Networks (GCNs). Figure 1 illustrates such a HIN which we discuss in detail in Section 3. For example, one can derive a homogeneous user graph based on an indirect path in the HIN, e.g., a graph with users and edges between two users if they have taken the same course. Given such a homogeneous graph, traditional GCNs can be applied to the graph to learn the representations/embeddings of users and concepts with respect to the chosen path.

Based on different indirect paths chosen, we can derive various user (concept) representations, and those representations of users (concepts) regarding different paths can be aggregated, e.g., using the mean of those representations. Instead of the straightforward mean aggregation, we propose and investigate different attention mechanisms to derive aggregated user (concept) representations based on different paths. The intuition behind using an attention mechanism is that different paths might have different importance for each user. Afterwards, those learned user and concept representations can be used for predicting the preference scores of concepts for recommendations. Our contributions in this work are as follows: (1) We propose an end-to-end framework[4] for predicting and recommending knowledge concepts of a user's interest in Section 4; (2) We investigate two attention mechanisms for aggregating information from different meta-paths (the definition can be found in Section 3) to derive user and concept representations. We then incorporate those representations into our extended matrix factorization framework for predicting the preference score of a concept with respect to a user; (3) Finally, we evaluate our approach with several baselines and state-of-the-art approaches in terms of well-established evaluation metrics, and show the effectiveness of our proposed approach in Section 6.

2. RELATED WORK

Recommender Systems and User Modeling on MOOC Platforms. There has been growing interest in recommender systems on MOOC platforms since 2013 with respect to different aspects such as course, video, and learning paths [16, 26, 3, 43, 9, 23]. For instance, the authors in [3] proposed YouEDU, which is a pipeline for classifying MOOC forum posts and recommending instructional video clips that might be helpful for resolving confusion detected in those posts. In [21], the authors showed that peer recommendations can improve users' engagement significantly in the context of a Project Management MOOC. Dai et al. [9] proposed analyzing course content for recommending personalized learning paths on MOOC platforms. Khalid et al. [18] provides a comprehensive survey on recent advances regarding different recommender systems in the context of MOOCs. More recently, researchers have started modeling user interests in the context of MOOCs while user modeling has been widely studied in other domains such as social media [42]. For example, Li et al. [22] investigated the impact of acquiring user interests via surveys or questionnaires on course recommendations. In [2], the authors proposed LeCoRe which exploits user interest modeling for recommending courses as well as similar users for promoting peer learning in enterprise environment. Gong et al. [13] argued that course recommendations overlook user interests regarding specific knowledge concepts, and studying users' online learning interests from a microscopic view and recommending knowledge concepts can capture user interests better and provide the flexibility of choosing learning resources of their interest. In this work, we also focus on the microscopic view for knowledge concept recommendations.

Recommendation Approaches with HIN. The basic idea of early recommendation approaches with HIN is to leverage path-based semantic relatedness between users and items over HINs, e.g., leveraging meta-path-based similarities for recommendation [40, 32, 41]. For example, Shi et al. [32] proposed predicting item ratings based on those from similar users measured via different meta-paths. With the advances of graph representation learning, the authors in [31] proposed using pre-trained user and item embeddings based on meta-path information with random walk, and incorporated those pre-trained embeddings as features into an extended matrix factorization framework. The most similar work to ours is Gong et al. [13], which is one of the first works for recommending knowledge concepts on MOOC platforms in a heterogeneous view. The authors showed that their proposed approach outperforms other CF-based baselines as well as metapath2vec [11], which uses learned node representations of a given HIN for knowledge concept recommendations by measuring the similarities between two nodes. Our work differs from [13] in several aspects. First, we formulate interacted concepts for each user as implicit feedback while [13] treated the number of clicks as ratings and formulated the problem as rating prediction for recommending top–k unknown concepts with higher ratings. Secondly, we investigate different attention mechanisms including the one incorporating the latent features of users (items) from matrix factorization. Thirdly, the prediction layer (Eq. 6) for estimating the preference score of a concept is different from [13] which uses the user (item) representations as features for the final prediction.

3. PRELIMINARIES

In this work, we consider the task of predicting and recommending concepts that a user might be interested in based

[4]GitHub repo:`https://github.com/parklize/kgc-rec`

on their learning history, which includes a set of learned concepts and their contextual information such as courses, videos, etc. With n users $\mathcal{U} = \{u_1, \cdots, u_n\}$, and m concepts $\mathcal{C} = \{c_1, \cdots, c_m\}$, we define an implicit feedback matrix $\mathbf{R} \in \mathbb{R}^{n \times m}$ with each entry $r_{u,c} = 1$ if u has learned c and $r_{u,c} = 0$ otherwise. The task can be framed in the context of HIN which is denoted as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ consisting of a object set $\mathcal{V}$ and a link set $\mathcal{E}$. A HIN is also associated with an object type mapping function $\phi : \mathcal{V} \rightarrow \mathcal{O}$ and a link type mapping function $\psi : \mathcal{E} \rightarrow \mathcal{R}$. $\mathcal{O}$ and $\mathcal{R}$ denote the sets of predefined object and link types, where $|\mathcal{O}|+|\mathcal{R}| > 2$ [33]. The MOOC data in our study can be represented as a HIN. The HIN consists of six types of entities such as *user*, *concept*, *video*, *course*, *school*, and *teacher*. In addition, there is a set of links describing the relationships among those entities. On top of the definition of HIN, the concept of network schema is used to describe the meta structure of a network [31].

The **network schema** [35] is denoted as $\mathcal{S} = (\mathcal{O}, \mathcal{R})$. It is a meta template for an information network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with the object type mapping $\phi : \mathcal{V} \rightarrow \mathcal{O}$ and the link type mapping $\psi : \mathcal{E} \rightarrow \mathcal{R}$, which is a directed graph defined over object types $\mathcal{O}$, with edges as links from $\mathcal{R}$. Fig. 1 shows the network schema of our MOOC dataset with the six different entity types and the semantic links between them. Given the network schema, we can extract semantic meta-paths between a pair of entities. A meta-path can be formally defined as follows:

A **meta-path** [34] MP is defined on a network schema $\mathcal{S} = (\mathcal{O}, \mathcal{R})$ and is denoted as a path in the form of $O_1 \xrightarrow{R_1} O_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} O_{l+1}$, which describes a composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_l$ between object O_1 and O_{l+1}, where $\circ$ denotes the composition operator on relations.

4. PROPOSED APPROACH

In this section, we introduce our proposed approach `MOOCIR` (MOOC Interest Recommender) based on meta-paths in the MOOC HIN. In high level, our approach extends the matrix factorization (MF) $\hat{y}_{u,c} = \mathbf{x}_u^T \mathbf{z}_c$, where $\hat{y}_{u,c}$ denotes the predicted preference score of concept c with respect to user u, and $\mathbf{x}_u$ and $\mathbf{z}_c$ refer to *latent features* of u and c, respectively. We extend the MF with user (concept) representations/embeddings that are learned by applying GCNs to meta-path-based graphs. Fig. 2 shows an overview of our approach, which consists of four main components. In the following, we describe each component in detail.

Table 1: Meta-paths selected for extracting user-user and concept-concept relationships.

Type	Meta-path
User	$user \rightarrow concept \xrightarrow{-1} user$
	$user \rightarrow course \xrightarrow{-1} user$
	$user \rightarrow video \xrightarrow{-1} user$
	$user \rightarrow course \rightarrow teacher \xrightarrow{-1} course \xrightarrow{-1} user$
Concept	$concept \rightarrow user \xrightarrow{-1} concept$
	$concept \rightarrow course \xrightarrow{-1} concept$

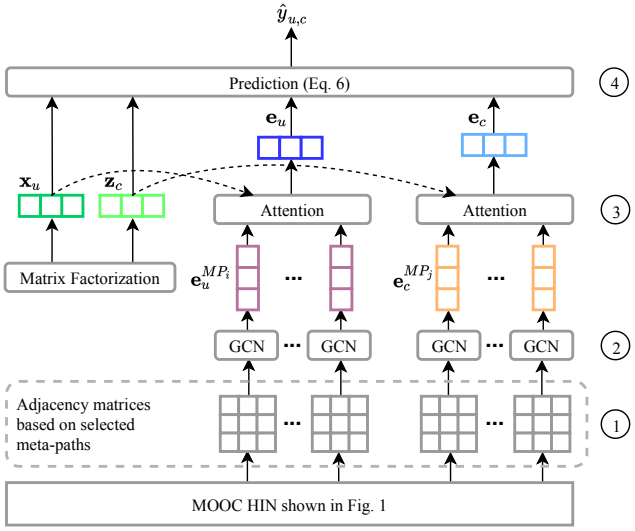

Figure 2: Overview of our proposed approach `MOOCIR`.

Meta-path selection. As discussed in Section 3, meta-paths provide the capability to derive entity-entity relationships through those paths. Similar to previous studies [13, 31], we consider user-user and concept-concept relationships via different meta-paths. To fairly compare with [31] in our experiments, we use the same set of meta-paths used in [31] for our study. Table 1 summarizes six meta-paths used for our work where four paths for users and two for concepts. For each meta-path, a homogeneous graph with respect to users (concepts) can be extracted, which is depicted as its corresponding adjacency matrix in Fig. 2. As one might expect, each entry in the adjacency matrix $\mathbf{A}$ regarding a meta-path is equal to one if two users (concepts) can be connected via that meta-path, and zero otherwise. Afterwards, we can learn user (concepts) representations for each meta-path using GCNs.

Graph Convolution Networks (GCNs). GCNs learn node representations of a graph by inspecting neighboring nodes. In this work, we adopt the following layer-wise propagation rule to learn user (concept) representations/embeddings with respect to a meta-path.

$$\mathbf{h}^{(l+1)} = g(\mathbf{P}\mathbf{h}^l\mathbf{W}^l) \tag{1}$$

where $g(\cdot)$ is an activation function which we use $ReLu$ [25] here. $\mathbf{P} = \mathbf{D}^{-1}\widetilde{\mathbf{A}}$ where $\mathbf{D}$ is the diagonal node degree matrix of $\mathbf{A}$ to normalize the matrix $\widetilde{\mathbf{A}}$, and $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is an adjacency matrix with self-loops in a graph based on a specific meta-path. $\mathbf{W}^l$ refers to a trainable weight matrix at layer l for all nodes. $\mathbf{h}^0$ can be fed with features of each node if there is a set of features for each node or can be initialized and learned afterwards as well. The output representation of the last layer can be used as user (concept) representations. For example, when $l = 2$, the representation of a user u for a meta-path MP_i will be $\mathbf{e}_u^{MP_i} = \mathbf{h}^3_{u^{MP_i}}$ where $\mathbf{h}^3_{u^{MP_i}}$ is the output of the last layer of GCNs for the meta-path MP_i with respect to u. In our study, we use a single layer GCN where $\mathbf{h}^0$ is initialized randomly and learned during the training process, but one can easily extend it with more

layers or using existing features for $\mathbf{h}^0$.

Attention. Attention mechanism [36] is motivated by how we pay visual attention to different regions of an image or relevant words in one sentence, and has been used widely to advance various fields such as natural language processing and recommender systems [37, 13]. In our context, different meta-paths can have different importance with respect to each user, and incorporating the importance of each meta-path differently for each user can be beneficial when aggregating user representations from different meta-paths, i.e., $\{\mathbf{e}_u^{MP_1} \dots \mathbf{e}_u^{MP_{|MP|}}\} \rightarrow \mathbf{e}_u$. In our work, we apply the attention mechanism from [6] for our context as follows:

$$\alpha_u^{MP_i} = \frac{exp(\mathbf{V}_u^T \sigma(\mathbf{W}_u \mathbf{e}_u^{MP_i}))}{\sum_{j \in |MP|} exp(\mathbf{V}_u^T \sigma(\mathbf{W}_u \mathbf{e}_u^{MP_j}))} \tag{2}$$

where the output $\alpha_u^{MP_i}$ indicates the weight (or importance) for $\mathbf{e}_u^{MP_i}$, and $\mathbf{V}_u^T$ and $\mathbf{W}_u$ are trainable matrices for users. The attention mechanism can be formulated in the same manner for concepts. Next, the user representations coming from different meta-paths can be aggregated as follows:

$$\mathbf{e}_u = \sum_{j \in |MP|} \alpha_u^{MP_j} \mathbf{e}_u^{MP_j} \tag{3}$$

The above-mentioned attention mechanism takes into account different meta-paths but does not consider any context in the extended MF, which can be the latent features of users and concepts for MF. Therefore, we also investigate the following attention mechanism which considers the latent features of user $\mathbf{x}_u$, which has not been explored in previous studies. In this case, a meta-path based embedding $\mathbf{e}_u^{MP_i}$ and $\mathbf{x}_u$ are concatenated together when calculating the attention scores as follows.

$$\beta_u^{MP_i} = \frac{exp(\mathbf{V}_u^T \sigma(\mathbf{W}_u [\mathbf{e}_u^{MP_i}; f(\mathbf{x_u})]))}{\sum_{j \in |MP|} exp(\mathbf{V}_u^T \sigma(\mathbf{W}_u [\mathbf{e}_u^{MP_j}; f(\mathbf{x_u})]))} \tag{4}$$

where $f(\mathbf{x_u})$ applies non-linearity with a single layer feed-forward neural networks to $\mathbf{x_u}$ instead of using it directly, which is inspired by [31] where the authors showed that non-linear fusion is required when combining latent features from matrix factorization and entity embeddings from GCNs. Afterwards, the final user representation can be obtained in the same manner as Eq. 3.

$$\mathbf{e}_u = \sum_{j \in |MP|} \beta_u^{MP_j} \mathbf{e}_u^{MP_j} \tag{5}$$

The attention mechanism can be formulated in the same manner for concepts.

Prediction. Given those learned user and concept representations/embeddings $\mathbf{e}_u$ and $\mathbf{e}_c$. The preference score of a concept c for a user u can be calculated as follows by extending the matrix factorization framework:

$$\hat{y}_{u,c} = \mathbf{x}_u^T \mathbf{z}_c + \gamma \cdot \mathbf{e}_u^T \mathbf{M} \mathbf{e}_c + b_c \tag{6}$$

where $\hat{y}_{u,c}$ is the preference score, $\mathbf{x}_u$ and $\mathbf{z}_c$ are the latent features for the matrix factorization, and b_c is a bias term. In addition, $\mathbf{M}$ is a trainable matrix to let $\mathbf{e}_u$ in the same space with $\mathbf{e}_c$, and γ is a trainable parameter for the trade-off between the prediction scores from matrix factorization and the user and concept embeddings.

4.1 Training Details

Loss function. We use the Bayesian Personalized Ranking (BPR) [29] which has been widely used for recommender systems with implicit feedback [7, 5, 27]. The intuition behind BPR is that a learned concept for a user should be ranked higher (with a higher score) compared to a random one in the list of concepts with which the user has not interacted, which can be formulated as follows:

$$\mathcal{L} = \sum_{(u,i,j) \in D_s} -ln(\sigma(\hat{y}_{uij})) + \lambda \left\|\Theta\right\|^2 \tag{7}$$

where (u, i, j) refers to a triplet including a user u, an interacted concept i and an unknown concept j for the user. $\hat{y}_{uij} = \hat{y}_{ui} - \hat{y}_{uj}$ measures the preference difference between the interacted concept and the unknown one, σ denotes the *sigmoid* function: $s(x) = \frac{1}{1+e^{-x}}$, λ is the regularization parameter for the $\mathcal{L}_2$ norm, and Θ denotes the set of parameters to be learned. The training set D_s can be constructed by paring an unknown concept randomly with an interacted concept in the training set of a user.

To learn the parameters of our proposed approach for minimizing the loss in Eq. 7, we use a mini-batch gradient decent with 1,024 as the batch size, and use the Adam update rule [20] to train the model using the training set. In addition, the learning rate is set as 0.01, the regularization parameter λ is set as $1e-8$, and the dimension of latent features for MF and that of user (concept) embeddings are set as 30 and 100 respectively as in [13].

To overcome the overfitting problem, we further construct a validation set by using the last interacted concept for each user, and randomly pair each known concept with 99 unknown concepts. We run 500 epochs where the convergence is observed, and monitor the performance of evaluation metrics (see Section 5) on the validation set. At the end, we choose the best-performing model on the validation set in terms of *MRR* (Mean Reciprocal Rank), which is one of the evaluation metrics measuring how well a ground truth concept is ranked in the corresponding set of 100 concepts. Any other evaluation metric can be used for choosing the best-performing model as well based on the preference for a specific metric.

5. EXPERIMENTAL SETUP

MOOC Dataset. We use the MOOCCube dataset [39] from the XuetangX platform for our experiments. The MOOCCube dataset is one of the largest and comprehensive MOOC datasets, and provides rich information about MOOCs and user activities on the platform from 2017 to 2019 [39]. Each course or video has a set of covered knowledge concepts in the dataset. In this work, we use user activities from 2017-01-01 to 2019-10-31 for training and those from 2019-11-01 to 2019-12-31 for testing. We limit users who have learned concepts in both training and testing periods and have at least one new concept (which did not appear in the training

Table 2: Statistics of the MOOCCube dataset for experiments.

Entities	Statistics	Relations	Statistics
users	2,005	user-concept	930,553
concepts	21,037	user-course	13,696
courses	600	course-video	42,117
videos	22,403	teacher-course	1,875
schools	137	video-concept	295,475
teachers	138	course-concept	150,811

period) in the testing period. Overall, the dataset consists of 2,005 users 21,037 concepts, 600 courses, 22,403 videos, 137 schools, 138 teachers, and the relationships among those entities. In total there are 930,553 interactions between users and concepts with 858,072 interactions in the training set and the rest (72,481) in the test set. The overall statistics of the dataset are presented in Table 2.

Evaluation Metrics. We evaluate the top–k predictions of concepts for users with the following widely used evaluation metrics where k is set to 5, 10, and 20. We calculate all metrics for each set of 100 concepts (with one interacted and 99 unknown) in the test set. For each interacted concept with respect to a user u, we generate the corresponding recommendation list $R_u = \{r_u^1, r_u^2, \ldots, r_u^k\}$ where r_u^i indicates concept ranked at the i–th position in R_u based on the predicted scores of those concepts.

Hit Ratio of top–k concepts ($HR@k$) measures the fraction of relevant concepts in the test set that are in the top–k concepts of the recommendations: $HR@k = \frac{1}{N}\sum_u I(|R_u \cap T_u|)$ where N is the total number of sets for testing, $I(x)$ is an indicator function which equals one if $x > 0$ and equals zero otherwise. *Normalized Discounted Cumulative Gain* ($nDCG@k$) takes into account rank positions of the relevant concepts, and can be computed as follows:$nDCG@k = \frac{1}{Z}DCG@k = \frac{1}{Z}\sum_{j=1}^{k}\frac{2^{I(|r_u^j \cap T_u|)}-1}{\log_2(j+1)}$ where Z denotes the score obtained by an ideal top–k ranking which serves as a normalization factor. *Mean Reciprocal Rank* (MRR) is the average of the reciprocal ranks of positive concepts: $MRR = \frac{1}{N}\sum_1^N \frac{1}{rank_i}$ where $rank_i$ refers to the rank position of the one interacted concept in the corresponding set of 100 concepts with the rest of unknown ones.

We use the paired *t-test* for testing the significance where the significance level of α is set to 0.05 unless otherwise noted.

5.1 Compared Methods

To better understand and investigate the contribution of each component and the performance with the two attention mechanisms introduced in Section 4, we first compare several variants of our approach. $\texttt{MOOCIR}_{\text{a1}}$ denotes our approach with the attention mechanism only considering different meta-paths using Eq. 3. $\texttt{MOOCIR}_{\text{a2}}$ refers to our approach with the attention mechanism incorporating the latent features of users (concepts) using Eq. 4. $\texttt{MOOCIR}_{\text{a-}}$ is a variant of our approach without any attention, i.e., different meta-paths are treated *equally* and the representations learned from those paths are averaged. $\texttt{MOOCIR}_{\text{mf-}}$ refers to a variant without the matrix factorization part for prediction in Eq. 6, which only uses meta-path-based user and concept representations for predicting the preference score of a concept.

Next, we compare `MOOCIR` with the following baselines and state-of-the-art methods to evaluate the performance of recommending knowledge concepts for users. `TopPop` is a straightforward baseline method which ranks concepts based on their popularity. Here, the popularity of a concept can be measured based on the number of users that have learned the concept. `MFBPR` [29] is a matrix factorization approach which optimizes a pairwise ranking loss for the recommendation task as our approach but without meta-path-based representation learning. That is, the second component in Eq. 6 based on user (concept) representations is removed. `FISM` [17] is an item-to-item collaborative filtering approach which provides recommendations based on the average embeddings of all interacted concepts and the embeddings of the target concept. `NAIS` [15] is also an item-to-item collaborative filtering approach, but with an attention mechanism, which is capable of distinguishing which historical items in a user profile are more important for a prediction. We use the author's implementation for both `NAIS` and `FISM`[5]. `metapath2vec` [11]. `metapath2vec` is a meta-path-based representation learning model which leverages meta-path-based random walks to construct the heterogeneous neighborhood of a node and then leverages a heterogeneous skip-gram model to learn node embeddings. We use the StellarGraph [10] implementation of `metapath2vec` for our experiment in which the parameters of `metapath2vec` are set the same as in [11] except the number of random walks is set as 500 instead of 1000[6]. `ACKRec` [13] also models the MOOC dataset as a HIN and extracts user (concept) representations from the same set of meta-paths in Table 1. However, `ACKRec` treats the problem as rating prediction task where the rating of a concept for a user is the number of interactions between the user and the concept. Also, it exploits user and concept representations as features while extending the matrix factorization framework. We use the author's implementation[7] for our experiments. `MFBPR` and those `MOOCIR` variants are implemented using Tensorflow [1]. All experiments are run on an Intel(R) Core(TM) i5-8365U processor laptop with 16GB RAM, and `MOOCIR` variants take less than two days for training.

6. RESULTS

Table 3 summarizes the results using the variants of `MOOCIR`. As we can see from the table, $\texttt{MOOCIR}_{\text{mf-}}$ — which uses user and concept representations learned based on meta-paths with the HIN but without the matrix factorization component — provides worse performance compared to the other variants. The results indicate that extending the matrix factorization is necessary for `MOOCIR`.

Next, we compare $\texttt{MOOCIR}_{\text{a-}}$ and the variants with attention mechanisms (i.e., $\texttt{MOOCIR}_{\text{a1}}$ and $\texttt{MOOCIR}_{\text{a2}}$). We observe that both $\texttt{MOOCIR}_{\text{a1}}$ and $\texttt{MOOCIR}_{\text{a2}}$ outperform $\texttt{MOOCIR}_{\text{a-}}$ in terms of all evaluation metrics, which shows that using at-

[5] `https://github.com/AaronHeee/Neural-Attentive-Item-Similarity-Model`

[6] We noticed that using 1000 random walks took more than 10 days for training and did not improve the performance compared to using 500.

[7] `https://github.com/JockWang/ACKRec`

Table 3: Performance of several variants of our proposed approach in term of different evaluation metrics with the best-performing scores in bold.

	HR			*nDCG*			*MRR*
	k=5	10	20	k=5	10	20	
MOOCIR$_{mf^-}$	0.676	0.812	0.906	0.499	0.543	0.567	0.468
MOOCIR$_{a^-}$	0.701	0.832	0.920	0.513	0.556	0.578	0.477
MOOCIR$_{a1}$	**0.704**	0.836	**0.922**	**0.520**	**0.562**	**0.584**	**0.484**
MOOCIR$_{a2}$	0.703	**0.838**	**0.922**	0.517	0.561	0.583	0.482

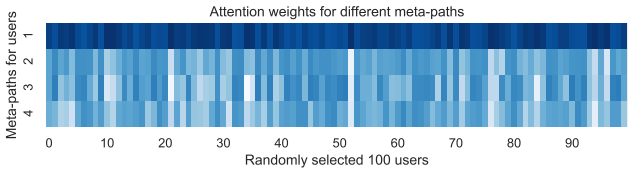

Figure 3: Attention weights of different meta-paths for 100 randomly chosen users learned using MOOCIR$_{a1}$ where we observe different weights of meta-paths for each user. In this heatmap, a darker cell indicates a higher attention weight.

tention can indeed improve the performance and different meta-paths have different importance for deriving user (concept) representations. This can be verified further by investigating learned attention weights for different meta-paths in MOOCIR as well. For example, Fig. 3 shows a heatmap regarding the learned attention weights for 100 randomly selected users using MOOCIR. In the figure, x-axis refers to the 100 users and y-axis indicates the attention weights for the four different meta-paths for users described in Table 1 in Section 4. From the figure, we can notice that the first meta-path (i.e., $user \rightarrow concept \xrightarrow{-1} user$) overall has a higher weight compared to others. In addition, we observe that the attention weights vary across users, which indicates the importance of each meta-path varies for different users.

Finally, by comparing the two different attention mechanisms, we observe that the one incorporating the latent features of users and concepts (Eq. 4) does not improve the performance compared to the simpler one (Eq. 3), which is different from our assumption. Instead, we observe that MOOCIR$_{a2}$ performs significantly worse than MOOCIR$_{a1}$ in terms of *HR@10* and *HR@20* for the users who have interacted with a limited number of concepts. Table 4 shows the performance for three groups of users with less than 150, 350, and 550 concepts, respectively. As we can see form the figure, MOOCIR$_{a1}$ outperforms MOOCIR$_{a2}$ significantly for the first group of 353 users. The results suggest that fusing information from the latent features of users (concepts) into the attention mechanism is a non-trivial task, and other approaches should be investigated in the future.

Table 4: Results of *HR@10* and *HR@20* for MOOCIR$_{a1}$ and MOOCIR$_{a2}$ for three groups of users (G150, G350, G550) with less than 150, 350, 550 concepts in the training set.

	HR@10			*HR@20*		
	G150	G350	G550	G150	G350	G550
MOOCIR$_{a1}$	0.806	0.830	0.851	0.894	0.911	0.927
MOOCIR$_{a2}$	0.801	0.829	0.852	0.886	0.908	0.925

Table 5: Performance of MOOCIR$_{a1}$ and compared methods in term of different evaluation metrics with the best-performing scores in bold.

	HR			*nDCG*			*MRR*
	k=5	10	20	k=5	10	20	
TopPop	0.486	0.629	0.767	0.343	0.390	0.425	0.332
MFBPR	0.668	0.811	0.907	0.481	0.527	0.552	0.448
FISM	0.584	0.701	0.800	0.438	0.476	0.501	0.418
NAIS	0.568	0.691	0.811	0.420	0.461	0.491	0.403
metapath2vec	0.642	0.773	0.873	0.468	0.511	0.537	0.440
ACKRec	0.659	0.764	0.842	0.503	0.538	0.557	0.475
MOOCIR$_{a1}$	**0.704**	**0.836**	**0.922**	**0.520**	**0.562**	**0.584**	**0.484**

Overall, MOOCIR$_{a1}$ provides the best performance among all variants. In the following, we discuss the performance of MOOCIR$_{a1}$ compared with other baselines and state-of-the-art methods.

Table 5 shows the performance of MOOCIR$_{a1}$ and compared methods. We first observe that all the other methods outperform TopPop which is a baseline method recommending popular concepts. For example, MOOCIR$_{a1}$ and ACKRec improves *MRR* over TopPop 45.8% and 43.1%, respectively. Among all the compared methods in Table 5, MOOCIR$_{a1}$ provides the best performance followed by ACKRec, MFBPR, and metapath2vec. ACKRec performs best in terms of *nDCG* and *MRR*, and MFBPR performs best in terms or *HR* among compared methods. In detail, a significant improvement of MOOCIR$_{a1}$ over ACKRec in *MRR* (+1.9%), *nDCG@5* (+3.1%), *nDCG@10* (+4.5%), *+nDCG@20* (4.8%) can be noticed ($\alpha < 0.01$). Compared to MFBPR, MOOCIR$_{a1}$ improves the *HR* scores 6.7%, 9.2%, and 9.4% when k =5, 10, 20, respectively ($\alpha < 0.01$). The two item-item CF methods (FISM and NAIS) do not perform well compared to MFBPR and ACKRec. One possible explanation might be due to the sparsity of the dataset, which makes that deriving item-item similarities based on interacted users for each item is challenging and limits the performance.

Those results indicate that the proposed approach MOOCIR$_{a1}$ can achieve competitive performance in terms of those evaluation metrics for top–k concept recommendations compared to the baselines and state-of-the-art methods.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we presented MOOCIR for predicting and recommending concepts that might be of users' interest on MOOC platforms. The comparison of MOOCIR variants in Section 6 shows that extending the matrix factorization with user and concept representations learned from different meta-paths and using attention for deriving those representations play crucial roles in achieving better performance. In addition, the results compared to other baselines and state-of-the-art methods indicate that MOOCIR$_{a1}$ can improve the performance of predicting and recommending concepts significantly. The comparison between the two introduced attention mechanisms (Eq. 3 and 4) suggests that a more comprehensive approach is required while fusing the latent features of users and concepts into the attention mechanism, which will be investigated in the near future.

8. REFERENCES

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, pages 265–283, 2016.

[2] K. Abhinav, V. Subramanian, A. Dubey, P. Bhat, and A. D. Venkat. Lecore: A framework for modeling learner's preference. In *EDM*, 2018.

[3] A. Agrawal, J. Venkatraman, S. Leonard, and A. Paepcke. Youedu: addressing confusion in mooc discussion forums by recommending instructional video clips. 2015.

[4] F. ALSaad and A. Alawini. Unsupervised approach for modeling content structures of moocs. In *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, pages 18–28, 2020.

[5] V. W. Anelli, T. Di Noia, E. Di Sciascio, A. Ragone, and J. Trotta. How to make latent factors interpretable by feeding factorization machines with knowledge graphs. In *International Semantic Web Conference*, pages 38–56. Springer, 2019.

[6] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[7] Y. Cao, X. Wang, X. He, Z. Hu, and T.-S. Chua. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The world wide web conference*, pages 151–161, 2019.

[8] Z. Chen, A. Brandon, C. Gayle, E. Nicholas, K. Daphne, and J. E. Ezekiel. Who's benefiting from moocs, and why. *Harvard Business Review*.

[9] Y. Dai, Y. Asano, and M. Yoshikawa. Course content analysis: An initiative step toward learning object recommendation systems for mooc learners. *International Educational Data Mining Society*, 2016.

[10] C. Data61. Stellargraph machine learning library. `https://github.com/stellargraph/stellargraph`, 2018.

[11] Y. Dong, N. V. Chawla, and A. Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 135–144, 2017.

[12] J. Gardner, Y. Yang, R. S. Baker, and C. Brooks. Modeling and experimental design for mooc dropout prediction: A replication perspective. *International Educational Data Mining Society*, 2019.

[13] J. Gong, S. Wang, J. Wang, W. Feng, H. Peng, J. Tang, and P. S. Yu. Attentional Graph Convolutional Networks for Knowledge Concept Recommendation in MOOCs in a Heterogeneous View. In *Proceedings of the 43rd SIGIR Conference on Research and Development in Information Retrieval*, pages 79–88, 2020.

[14] H. Hajri, Y. Bourda, and F. Popineau. Personalized recommendation of open educational resources in moocs. In *International Conference on Computer Supported Education*, pages 166–190. Springer, 2018.

[15] X. He, Z. He, J. Song, Z. Liu, Y.-G. Jiang, and T.-S. Chua. Nais: Neural attentive item similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 30(12):2354–2366, 2018.

[16] X. Jing and J. Tang. Guess you like: course recommendation in moocs. In *Proceedings of the International Conference on Web Intelligence*, pages 783–789, 2017.

[17] S. Kabbur, X. Ning, and G. Karypis. Fism: factored item similarity models for top-n recommender systems. In *Proceedings of the 19th ACM SIGKDD*, pages 659–667, 2013.

[18] A. Khalid, K. Lundqvist, and A. Yates. Recommender systems for moocs: A systematic literature survey (january 1, 2012 – july 12, 2019). *The International Review of Research in Open and Distributed Learning*, 21(4):255–291, Jun. 2020.

[19] H. Khalil and M. Ebner. Moocs completion rates and possible methods to improve retention-a literature review. In *EdMedia+ innovate learning*, pages 1305–1313. Association for the Advancement of Computing in Education (AACE), 2014.

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[21] H. Labarthe, F. Bouchet, R. Bachelet, and K. Yacef. Does a peer recommender foster students' engagement in moocs?. *International Educational Data Mining Society*, 2016.

[22] X. Li, T. Wang, H. Wang, and J. Tang. Understanding user interests acquisition in personalized online course recommendation. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, pages 230–242. Springer, 2018.

[23] F. Mi and B. Faltings. Adaptive sequential recommendation for discussion forums on moocs using context trees. In *Proceedings of the 10th international conference on educational data mining*, number CONF, 2017.

[24] C. Milligan and A. Littlejohn. Why study on a mooc? the motives of students and professionals. *International Review of Research in Open and Distributed Learning*, 18(2):92–102, 2017.

[25] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.

[26] Y. Pang, C. Liao, W. Tan, Y. Wu, and C. Zhou. Recommendation for mooc with learner neighbors and learning series. In *International Conference on Web Information Systems Engineering*, pages 379–394. Springer, 2018.

[27] G. Piao and J. G. Breslin. Transfer learning for item recommendations and knowledge graph completion in item related domains via a co-factorization model. In *European Semantic Web Conference*, pages 496–511. Springer, 2018.

[28] B. Prenkaj, P. Velardi, D. Distante, and S. Faralli. A reproducibility study of deep and surface machine learning methods for human-related trajectory prediction. In *Proceedings of the 29th ACM CIKM*, pages 2169–2172, 2020.

[29] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint*

arXiv:1205.2618, 2012.

[30] D. T. Seaton, Y. Bergner, I. Chuang, P. Mitros, and D. E. Pritchard. Who does what in a massive open online course? *Communications of the ACM*, 57(4):58–65, 2014.

[31] C. Shi, B. Hu, W. X. Zhao, and S. Y. Philip. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31(2):357–370, 2018.

[32] C. Shi, Z. Zhang, P. Luo, P. S. Yu, Y. Yue, and B. Wu. Semantic path based personalized recommendation on weighted heterogeneous information networks. In *Proceedings of the 24th ACM CIKM*, pages 453–462, 2015.

[33] Y. Sun and J. Han. Mining heterogeneous information networks: a structural analysis approach. *Acm Sigkdd Explorations Newsletter*, 14(2):20–28, 2013.

[34] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003, 2011.

[35] Y. Sun, Y. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *Proceedings of the 15th ACM SIGKDD*, pages 797–806, 2009.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008, 2017.

[37] C. Wu, F. Wu, M. An, J. Huang, Y. Huang, and X. Xie. Npa: neural news recommendation with personalized attention. In *Proceedings of the 25th ACM SIGKDD*, pages 2576–2584, 2019.

[38] M. Yao, S. Sahebi, and R. F. Behnagh. Analyzing student procrastination in moocs: A multivariate hawkes approach. *International Educational Data Mining Society*, 2020.

[39] J. Yu, G. Luo, T. Xiao, Q. Zhong, Y. Wang, J. Luo, C. Wang, L. Hou, J. Li, and Z. Liu. MOOCCube: A Large-scale Data Repository for NLP Applications in MOOCs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3135–3142, 2020.

[40] X. Yu, X. Ren, Q. Gu, Y. Sun, and J. Han. Collaborative filtering with entity similarity regularization in heterogeneous information networks. *IJCAI HINA*, 27, 2013.

[41] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 283–292, 2014.

[42] F. Zarrinkalam, S. Faralli, G. Piao, E. Bagheri, et al. Extracting, mining and predicting users' interests from social media. *Foundations and Trends® in Information Retrieval*, 14(5):445–617, 2020.

[43] J. Zhao, C. Bhatt, M. Cooper, and D. A. Shamma. Flexible learning with semantic visual exploration and sequence-based recommendation of mooc videos. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2018.