# Deep Correlation based Concept Recommendation for MOOCs

1st Shengyu Mao
*Zhejiang University of Technology*
Hangzhou, China
201806062515@zjut.edu.cn

2nd Pengyi Hao
*Zhejiang University of Technology*
Hangzhou, China
haopy@zjut.edu.cn

3th Cong Bai
*Zhejiang University of Technology*
Hangzhou, China
congbai@zjut.edu.cn

***Abstract*—The current course recommendation in massive open online courses (MOOCs) usually ignores students' interests in some certain type of knowledge concepts, resulting in low completion of most courses. Therefore, it requires a concept recommendation to help students accurately choose courses in MOOCs. In this paper, we propose Deep Correlation based Concept Recommendation (DCCR) for MOOCs. It gathers the interactive information obtained by different entities through meta-paths in MOOCs and extracts the semantic information of concepts. To deeply capture the correlation information among users, a multi-relation graph is built to generate the correlation features which aggregates the abundant information under different meta-paths. Then through the graph convolutional neural networks, entity embeddings of users and knowledge concepts are generated. Additionally, a concatenation-based fusion function is designed to get the final joint representations reasonably. By verifying on two public datasets, experiments show that DCCR outperforms the state-of-the-art methods.**

***Index Terms*—concept recommendation, correlation feature, concatenation based fusion, moocs**

## I. Introduction

MOOCs have been developing rapidly in recent years, providing users with a convenient way of education [12]. The emergence of MOOCs' platforms has completely changed the entire education field. However, according to statistics, there is a very low completion rate of online courses [10]. So there are lots of works concentrating on course recommendation in MOOCs. But barely focus on courses would lead to some problems. 1) The normal course recommendation could probably cause students to take courses that they are not attracted to. For example, some computer vision courses only cover knowledge about geometric, and others may cover deep learning, which could mislead the result in recommendation. 2) The content and focus of similar courses are different. For instance, in advanced mathematics, some courses are based on geometry, and some courses are based on calculus, which is difficult for students to select [16]. 3) Students with different prerequisite background may require totally different [5], but if recommending in the traditional way, students may not realize that they have not learnt some necessary prerequisite knowledge until taking certain courses [8]. Therefore, the MOOCs need to accurately locate the learning needs of students, which makes it necessary to do concept recommendation, a more fine-grained recommendation task.

The existing course or concept recommendations are mainly divided into two categories. One is the methods based on collaborative filtering (CF), which considers the historical interaction behavior of users with other online resources (e.g., videos, courses), and explores the potential of the same preference [9]. For example, He et al. proposed a neural network model to evaluate the similarity between items by using an attention network [7]. Elbadrawy et al. used neighborhood-based user collaborative filtering to design a course ranking model [6]. These methods have achieved great success in recommendation courses, but suffering from the problem of data sparsity and cold start, so the performance is limited. The other one constructs the MOOCs information into heterogeneous information networks, and utilize meta-paths to guide the dissemination of student preferences. It always captures the corresponding fruitful semantic relationships between different types of entities, learns the embedding, and finally generates a recommendation list through matrix factorization [16]. Vashishth et al. [14] proposed a three-way neural interaction model to use the rich meta-path-based information for recommendation. Gong et al. [16] proposed a method to capture the representation of different types of entities in heterogeneous information networks, fused the content features of entities to do recommendation. However, due to the relative independence of different meta-path relationships, this kind of methods can not completely capture the interactive information from the heterogeneous networks in MOOCs, and may lose some inner information among different meta-paths, resulting in an unsatisfactory performance.

In this paper, we excavate the deep correlation of users. Motivated by some efficient multi-relational graph methods like [15], we propose deep correlation based concept recommendation (DCCR) for MOOCs, where a multi-relational graph is constructed to deeply capture the inner correlation of users among different meta-paths. With the deep correlation feature extracting from the multi-relational graph , representations for users in MOOCs can be better learnt from graph convolutional network to accurately reflect users' preference. Meanwhile, we deeply extract the information contained in concepts themselves including names, definitions, etc. as auxiliary features, which makes the recommendation system

DOI reference number: 10.18293/SEKE2022-107

more effective. We also propose a concatenation-based fusion function to better combine the entity representations under different meta-paths, and finally get the rating matrix from users towards knowledge concepts. The key contributions of this paper can be summarized as: (i) deep correlation features of users are generated by constructing a multi-relational graph among different meta-paths, which leads to better embedded representations of user entities. (ii) a feature fusion function is proposed by concentrating on the different entity embeddings under different meta-paths, which leads to more reasonable representation for users and concepts. (iii) the DCCR is evaluated on two publicly available real datasets MOOCCube [17] and XuetangX [16], not only the performance in each dataset is compared, but also the difference between the two datasets are analysised.

## II. Proposed Method

### A. Probelm Statement

In the recommendation task, MOOCs' data usually includes five specific entities (user(**u**), knowledge concept(**k**), video(**v**), course(**c**), teacher(**t**) [13], [16]). Additionally, there are abundant text information along with knowledge concepts, including their definitions, descriptions, and classifications. The purpose is to generate concept recommendation list for each user. The framework of DCCR is shown in Fig. 1, and the explanations of notations are given in Table I.

TABLE I: Notations and explanations.

| Notation | Explanation |
|---|---|
| $S_k$ | semantic feature of knowledge concepts |
| $C_u$ | correlation feature of users |
| $d_u, d_k$ | the dimension of correlation feature and semantic feature |
| $N_u, N_k$ | the number of users and knowledge concepts |
| $MP^u, MP^k$ | the meta-path sets of users and knowledge concepts |
| $A_{mp^u}, A_{mp^k}$ | the adjacency matrix sets of users and knowledge concepts |
| $G$ | the correlation triad set |
| $z$ | the correlation triad threshold |
| $f_{mp^u}, f_{mp^k}$ | the representation sets of users and knowledge concepts |
| $p$ | the scale parameter |
| $F_k, F_u$ | the final representations of users and knowledge concepts |
| $d_F$ | the dimension of the final representations of users and knowledge concepts |
| $r_{u,k}$ | the true rating of user $u$ to knowledge concept $k$ |
| $\hat{r_{u,k}}$ | the predicted rating of user $u$ to knowledge concept $k$ |
| $x_u$ | the latent factors of user $u$ |
| $y_k$ | the latent factors of knowledge concept $k$ |
| $t_u, t_k$ | the parameters to integrate the $F_u$ and $F_k$ in the same space |
| $\beta_u, \beta_k$ | the tuning parameters |
| $\delta$ | the regularization parameters |

### B. Semantic Feature Extraction

Ordinarily, the name of a concept, almost the generalization of itself, contains rich semantic information. Moreover, just like the name, there is also a wealth of information in concept's subject classifications, e.g., the subject classifications of 'Binary tree' are 'theoretical computer science' and 'data structure', so at least two specific subject classifications can be collected for this knowledge concept.

After separating the text informations into words, a parameter $c$ is set as the number of classification we chose, and then the word vectors $S_{name}, S_{class_1}, S_{class_2}, \cdots, S_{class_c} \in \mathbb{R}^{N_k \times d_v}$ are generated by *Word2Vec* [18], in which $N_k$ indicates the total number of knowledge concepts, and $d_v$ indicates the dimension of word vectors. We stitch the word vectors together to get the semantic feature of concepts $S_k \in \mathbb{R}^{N_k \times d_k}$, $S_k = [S_{name}, S_{class_1}, S_{class_2}, \cdots, S_{class_c}]$, here $d_k = d_v \times (c+1)$ indicates the dimension of semantic feature.

### C. Meta-Path Adjacency Matrices

Given the interactive information between different kinds of entities in MOOCs (e.g., user *u3125* have learned courses *c254* and *c617*), we build the interactive matrices between entities, including user-click-knowledge concept matrix, user-watch-video matrix, user-learn-course matrix, user-learn-course-taught by-teacher matrix, knowledge concept-included by-video matrix and knowledge concept-involved-course matrix. Each element in each matrix belongs to $\{0, 1\}$, which represents the interaction between two specific entities.

Meta-path [4] means the semantic path that connects different entities and illustrates the relational information in the dataset. Here, meta-paths are defined like $\mathbf{u} \stackrel{watch}{\longrightarrow} \mathbf{v} \stackrel{watch^{-1}}{\longrightarrow} \mathbf{u}$, which indicates that two different users are connected because they have watched the same video; $\mathbf{k} \stackrel{clicked\ by}{\longrightarrow} \mathbf{u} \stackrel{clicked\ by^{-1}}{\longrightarrow} \mathbf{k}$ means that two different knowledge concepts have been clicked by the same user. In this way, the meta-path sets $MP^u = \{mp_1^u, mp_2^u, \cdots, mp_m^u\}$ and $MP^k = \{mp_1^k, mp_2^k, \cdots, mp_n^k\}$ are set for users and knowledge concepts respectively, where $m, n$ indicate the number of meta-paths of users and knowledge concepts.

With the interactive information matrices and the meta-path sets, the corresponding adjacency matrix sets $A_{mp^u} = \{A_{mp_1^u}, A_{mp_2^u}, \cdots, A_{mp_m^u}\}$ and $A_{mp^k} = \{A_{mp_1^k}, A_{mp_2^k}, \cdots, A_{mp_n^k}\}$ are generated for users and concepts. Taking users as an example, $A_{mp^u}$ is generated and normalized by the following formula,

$$A_{mp_i^u} = Norm(L_u^e \cdot {L_u^e}^T), \tag{1}$$

here $L_u^e$ is the corresponding interactive matrix of $mp_i^u$(e.g., the user-watch-video matrix is the corresponding interactive matrix of meta-path $\mathbf{u} \stackrel{watch}{\longrightarrow} \mathbf{v} \stackrel{watch^{-1}}{\longrightarrow} \mathbf{u}$), $A_{mp_i^u}$ is the normalized adjacency matrix. After iterating through all the meta-paths in $MP^u$, the meta-path adjacency matrix $A_{mp^u}$ of users is obtained. In the same way, the meta-path adjacency matrix of knowledge concepts $A_{mp^k} = \{A_{mp_1^k}, A_{mp_2^k}, \cdots, A_{mp_n^k}\}$ is also generated.

### D. Correlation Feature Extraction

The meta-path adjacency matrix sets generated above have captured many significant information from MOOCs. However, the relative independence of those adjacency matrices loses lots of correlation information among meta-paths. Fig. 2 shows the contrast between with and without correlation among meta-paths. Since there are various connections between users and other entities, the correlation information among different users' preferences can be deeply gathered.

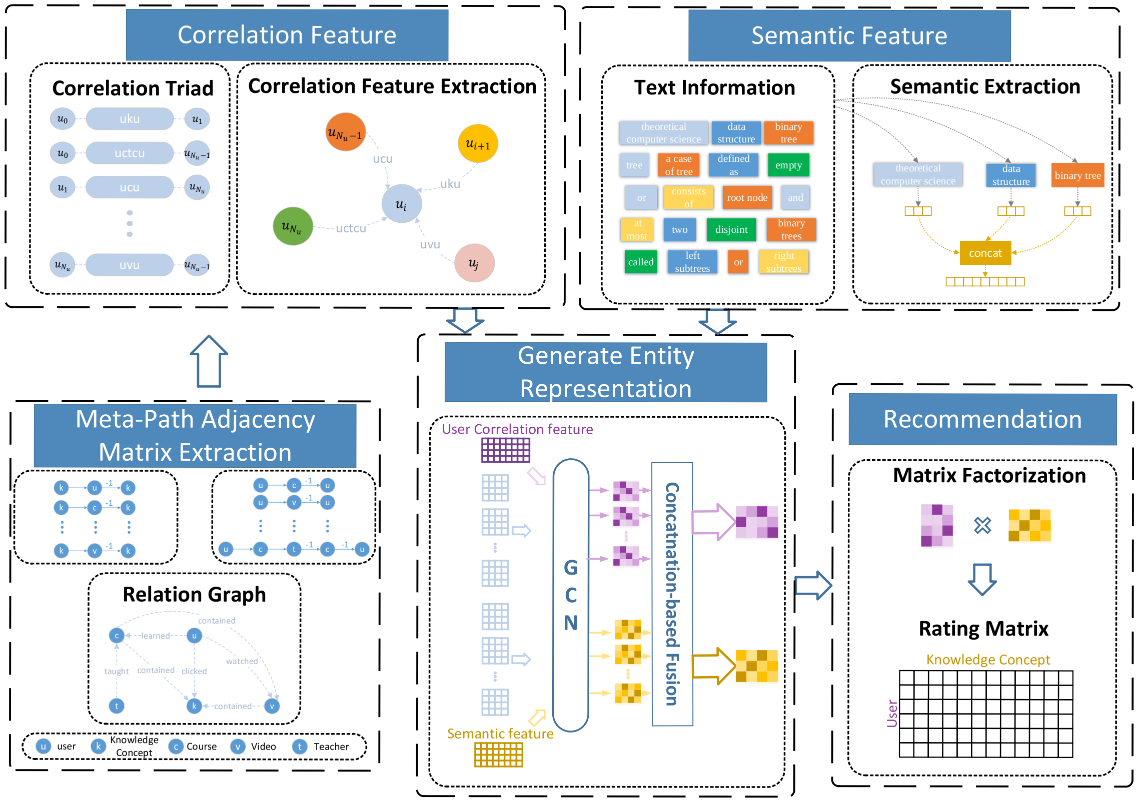

Fig. 1: The framework of DCCR for MOOCs.

First, a threshold $z$ is used to select the users that are strongly correlated. Then, a user correlation triad set $G$ is created. For specific user entities $u_i$ and $u_j$, if their corresponding value in $A_{mp_q^u}(q \in [1, m])$ is lager than $z$, which means that these two users are correlated in $mp_q^u$, then the correlation information $(u_i, r, u_j)$ will be written into $G$, where $r$ is the meta-path $mp_q^u$.

Based on the triad set $G$, we build a multi-relational graph by $Graph = (\mathcal{V}, \mathcal{R}, \mathcal{E}, \mathcal{X}, \mathcal{Z})$, where $\mathcal{V}$ is the node set of entities, $\mathcal{R}$ is the correlation set of entities, $\mathcal{E}$ is the enlarged correlation triad set, $\mathcal{X}$ and $\mathcal{Z}$ are the initial feature of nodes and relations, $\mathcal{V}, \mathcal{R}, \mathcal{E}$ are built by $G$. For every triad $(u_i, r, u_j) \in G$, $u_i, u_j$ are involved in $\mathcal{V}$; $\mathcal{R} = \mathcal{R}' \cup \mathcal{R}'_{inv} \cup \{Se\}$, where $\mathcal{R}' = \{r | (u_i, r, u_j) \in G\}$, $\mathcal{R}'_{inv} = \{r^{-1} | (u_i, r, u_j) \in G\}$, $Se$ means the self-loop correlation; $\mathcal{E} = \{(u_i, r, u_j) | (u_i, r, u_j) \in G\} \cup \{(u_j, r^{-1}, u_i) | (u_i, r, u_j) \in G\} \cup \{(u, Se, u) | u \in \mathcal{V}\}$. After that, we get the embeddings with following rules,

$$h_{u_j}^1 = tanh\Big(\sum_{(u_i, r) \in N(u_j)} W_{\lambda(r)}^1 \phi(x_{u_i}, z_r)\Big), \tag{2}$$

$$h_{u_j}^2 = tanh\Big(\sum_{(u_i, r) \in N(u_j)} W_{\lambda(r)}^2 \phi(h_{u_i}^1, h_r^1)\Big), \tag{3}$$

where $N(u_j)$ is a set of immediate neighbours of $u_j$ for its outgoing edges, $\phi : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a composition operator, $\lambda(r)$ indicates the relation type of $r$, $W_{\lambda(r)}^k$ is a relation-type shared parameter at $k$-th layer, $x_{u_i}$ and $z_r$ are the initial features of node $u_i$ and relation $r$, $h_{u_j}^1$ is the feature of node $u_j$ generated at the first layer, $h_r^1$ is the representation of relation generate at first layer which follows the rule $h_r^{k+1} = W_{rel}^k h_r^k$, here $W_{rel}^k$ are the shared parameters for each relation. $h_{u_j}^2 \in \mathbb{R}^{d_u}$ is the output of the second layer as well as the correlation feature of $u_j$. All the parameters are randomly initialized. Finally the correlation feature matrix $C_u \in \mathbb{R}^{N_u \times d_u}$ is generated as,

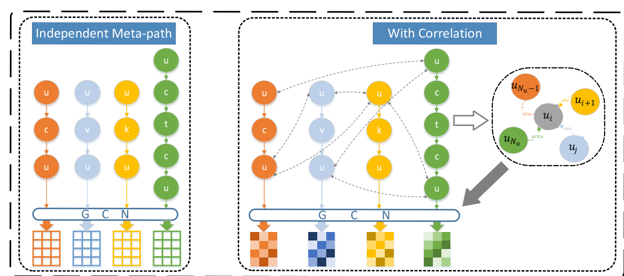

Fig. 2: Contrast between with and without correlation

$$C_u = [h_{u_1}^2, h_{u_2}^2, \cdots, h_{u_{N_u}}^2]^T, \tag{4}$$

here, $N_u$ is the number of users and $d_u$ is the dimension of correlation feature.

### E. Concatenation-based Representations Learning

Unlike the specific values in meta-path adjacency matrices $A_{mp}$, the triads in the multi-relation graph can not tell the degree that two users are correlated. So it's not efficient to take the correlation feature $C_u$ as the final representations of users. We need to further generate the representations of users and concepts under every meta-path. Given the semantic feature $S_k$, correlation feature $C_u$, and the meta-path adjacency matrices $A_{mp^k}$ and $A_{mp^u}$ as inputs, graph convolutional network (GCN) [19] is adopted with a layer-wise propagation rule for both users and concepts. Taking user as an example, the propagation layer is defined as

$$h_{mp_i^u}^{(l+1)} = ReLU(P_{mp_i^u} h_{mp_i^u}^l W^l). \tag{5}$$

Here $h_{mp_i^u}^{(l+1)}$ indicates the new representation of users under $mp_i^u$ at layer $l+1$. $P_{mp_i^u} = \widetilde{D}^{-\frac{1}{2}} \cdot (A_{mp_i^u} + I) \cdot \widetilde{D}^{-\frac{1}{2}}, \widetilde{D} = diag((A_{mp_i^u} + I) \cdot \mathbf{1})$, $I$ is the identity matrix, $\mathbf{1}$ is the all-ones vector, $W^l$ is the shared trainable weight matrix at layer $l$ for every meta-path. Particularly, we take $C_u$ as $h_{mp_i^u}^0$ for users at the first layers, and take the output at the third layer as the representation under $mp_i^u$,

$$f_{mp_i^u} = h_{mp_i^u}^3. \tag{6}$$

After iterating through all the matrices in $A_{mp_i^u}$, we ultimately get the representation sets $f_{mp^u} = \{f_{mp_1^u}, f_{mp_2^u}, \cdots, f_{mp_m^u}\}$ for users. In the same way, we take $S_k$ as $h_{mp_i^k}^0$ for knowledge concept at the first layer to adopt GCN and generate the representations as

$$f_{mp_i^k} = h_{mp_i^k}^3. \tag{7}$$

Finally, $f_{mp^k} = \{f_{mp_1^k}, f_{mp_2^k}, \cdots, f_{mp_m^k}\}$ is obtained for concepts.

In order to evenly consider meta-paths, joint representation by fusing representations under different meta-paths should be considered. However, in MOOCs, users' interactivity under different meta-paths are quite dissimilar. To solve this problem, we design a concatenation-based fusion function, which generates fusion weights by concentrating on representations under both the current meta-path and the others at the same time. In this way, we obtain a reasonable joint representation, which can reflects the association between meta-paths effectively. Taking users as an example, fusion weight $\alpha_{mp_i^u}$ of $mp_i^u$ is calculated as

$$\alpha_{mp_i^u} = softmax(v(tanh(w_1 f_{mp_i^u} p + w_2 \mathtt{f}_{mp_i^u}(1-p) + b))), \tag{8}$$

where $\mathtt{f}_{mp_i^u} = \frac{1}{m-1} \cdot \sum_{j \neq i}^{m} f_{mp_j^u}$ is the second concerned object of $mp_i^u$, $v, w_1, w_2, b$ are trainable parameters, $p \in (0, 1)$ is a scale hyper-parameter. Then the final joint representation of user $F_u \in \mathbb{R}^{N_u \times d_F}$ is calculated as

$$F_u = \sum_{i=1}^{m} \alpha_{mp_i^u} f_{mp_i^u}. \tag{9}$$

Similarly, the final representation of knowledge concept $F_k \in \mathbb{R}^{N_k \times d_F}$ is calculated as

$$F_k = \sum_{i=1}^{m} \alpha_{mp_i^k} f_{mp_i^k}. \tag{10}$$

Algorithm 1 shows how to generate $F_u$ and $F_k$.

**Algorithm 1:** Generate the representations of users and concepts

**Input:** The interactive information between different entities in MOOCs, the meta-path sets of users and concepts $MP^u, MP^k$, the text information of concepts

**Output:** The representations $F_u, F_k$ of users and concepts

Initialize $A_{mp^u}, A_{mp^k}, f_{mp^u}, f_{mp^k}$ as empty lists
Extract semantic feature $S_k$ from text information
**for** *each* $mp_i^k \in MP^k$ **do**
  Calculate $A_{mp_i^k}$ by Eq (1)
  Add $A_{mp_i^k}$ to $A_{mp^k}$
**for** *each* $mp_i^u \in MP^u$ **do**
  Calculate $A_{mp_i^u}$ by Eq (1)
  Add $A_{mp_i^u}$ to $A_{mp^u}$
Represent a multi-relational graph by $A_{mp^u}$
Extract correlation feature $C_u$ by Eq (2,3,4)
**for** *each* $A_{mp_i^u} \in A_{mp^u}$ **do**
  $h_{mp_i^u}^0 \leftarrow C_u$
  **for** $l = 0$ *to* 2 **do**
    Calculate $h_{mp_i^u}^{l+1}$ by Eq (5)
  $f_{mp_i^u} = h_{mp_i^u}^3$ according to Eq (6)
  Add $f_{mp_i^u}$ to $f_{mp^u}$
**for** *each* $A_{mp_i^k} \in A_{mp^k}$ **do**
  $h_{mp_i^k}^0 \leftarrow S_k$
  **for** $l = 0$ *to* 2 **do**
    Calculate $h_{mp_i^u}^{l+1}$ by Eq (5)
  $f_{mp_i^k} = h_{mp_i^k}^3$ by Eq (7)
  Add $f_{mp_i^k}$ to $f_{mp^k}$
**for** *each* $f_{mp_i^u} \in f_{mp^u}$ **do**
  Calculate $\alpha_{mp_i^u}$ by Eq (8)
Generate $F_u$ by Eq (9)
**for** *each* $f_{mp_i^k} \in f_{mp^k}$ **do**
  Calculate $\alpha_{mp_i^k}$ by Eq (8)
Generate $F_k$ by Eq (10)
**return** $F_u, F_k$

### F. Concept Recommendation for User

Lastly, an extend matrix factorization is utilized to complete the recommendation task. The predicted rating matrix $\hat{r_{u,k}}$ is got as follows,

$$\hat{r_{u,k}} = x_u^T y_k + \beta_u F_u^T t_k + \beta_k t_u^T F_k, \tag{11}$$

where $x_u$ and $y_k$ are randomly initialized latent factors of user and knowledge concept, $t_u$ and $t_k$ are parameters that make sure $F_u$ and $F_k$ to be in the same space, $\beta_u$ and $\beta_k$ are tuning parameters. We define the following loss function for reaching an appropriate rating prediction,

$$\begin{aligned} Loss = & \frac{1}{N_u \times N_k} \sum_{u=1}^{N_u} \sum_{k=1}^{N_k} (r_{u,k} - \hat{r_{u,k}})^2 + \\ & \delta(||x_u||_2 + ||y_k||_2 + ||t_u||_2 + ||t_k||_2), \end{aligned} \tag{12}$$

where $r_{u,k}$ is the target rating matrix of user on knowledge concept, $\delta$ is the regularization parameter. Finally, with the rating matrix of user on knowledge concept, the concepts are recommended with the highest rating for each user.

## III. Experiment

### A. Datasets

To evaluate the effectiveness of the proposed method, we adopt two datasets, MOOCCube [17] and the real data from XuetangX [16]. MOOCCube [17] is a large-scale data repository of over 700 MOOC courses, 100k concepts, 8 million student behaviors with an external resource. The abundant data of MOOCCube was mostly obtained from Baidubaike, Wikipedia, and Termonline. We divide the interactive behaviors of users to concepts into training set and test set with a ratio of 8:2. XuetangX [16] includes a training set occurring between October 1st, 2016 and December 30th, 2016 and a test set with the data occurring between January 1st, 2018 and March 31st, 2018. It contains 7,020 MOOC courses 43,405 videos, 1,029 course concepts, and 9,986 real MOOC users. For both datasets, we paired 99 randomly sampled negative instances with 1 positive instance for each users, and output the prediction rating [7].

### B. Evaluation Metrics and Implementation Details

Several metrics are utilized to evaluate the recommendation methods. $HR_{@K}$ is a common recall measure that shows the percentage of top-K recommendations that were successful. $NDCG_{@K}$ [3] is used to evaluate the differences between this ranking list and the user's actual interaction list. $MRR$ [7] is used for evaluating any process that produces a list of possible responses to a sample of queries. Additionally, $AUC$ is also used as a metric.

The methods are run in the environment of python-3.7, tensorflow-1.13.1. When extracting correlation features, We set the initial dimension size for nodes and relations to 100 and the output dimension to 200. When learning representations, we set the dimension to 256, 128 and 64 at the first, second and output layer, respectively. Moreover, we set the dropout rate to be 0.5 and the latent dimension to be 30 in MF. As for the learning rate, we set it to be 0.001, and implement an exponential learning rate decays every 100 steps.

### C. Analysis of the Proposed Method

*1) Evaluation of Meta-path Combinations:* We emphatically analyse the influence of the selection and combination of different meta-paths through the whole recommendation task by referring the combinations in [16], but we further conduct a detailed analysis in two different datasets. Specifically, we consider the following meta-path combinations in both datasets, including $mp_1 : \mathbf{u} \rightarrow \mathbf{k} \overset{-1}{\rightarrow} \mathbf{u}, mp_2 : \mathbf{u} \rightarrow \mathbf{c} \overset{-1}{\rightarrow} \mathbf{u}, mp_3 : \mathbf{u} \rightarrow \mathbf{v} \overset{-1}{\rightarrow} \mathbf{u}$ and $mp_4 : \mathbf{u} \rightarrow \mathbf{c} \rightarrow \mathbf{t} \overset{-1}{\rightarrow} \mathbf{c} \overset{-1}{\rightarrow} \mathbf{u}$.

From Table II, the rank of effectiveness is $mp_1 > mp_3 > mp_4 > mp2$ in MOOCCube, and $mp_3 > mp_1 > mp2 > mp_4$ in XuetangX. Additionally, the combinations of meta-paths perform better than the individual, and the tendency of the effect is the same as the individual. For instance, in MOOCCube the performance of $mp_{1\&3}$ is better than $mp_{1\&2}$, and in XuetangX the performance of $mp_{1\&2\&3}$ is better than $mp_{1\&2\&4}$, which indicates that users have dissimilar interactive behaviors under different meta-paths. In general, it works best when combining all the four meta-paths. In MOOCCube, the $AUC$ of $mp_{1\&2\&3\&4}$ is 4.17%, 6.18%, 4.78%, 5.67% higher than $mp_1$, $mp_2$, $mp_3$, $mp_4$, respectively. In XuetangX, the $AUC$ of $mp_{1\&2\&3\&4}$ is 6.74%, 9.79%, 6.48%, 10.07% higher than $mp_1$, $mp_2$, $mp_3$, $mp_4$, respectively.

From Table II, it exhibits a larger increase in XuetangX than in MOOCCube when combing more meta-paths. For example, the $AUC$ grows 0.75% from $mp_{1\&2}$ to $mp_{1\&2\&3}$ in MOOCCube, while it grows 2.01% in XuetangX. It is because of that XuetangX has more courses, videos and teacher entities besides users and concepts, depending on which we can extract more complete correlation features from XuetangX.

TABLE II: Results of different meta-path combinations

| meta-path | MOOCCube | | | | XuetangX | | | |
|---|---|---|---|---|---|---|---|---|
| | $HR_{@5}$ | $NDCG_{@5}$ | $MRR$ | $AUC$ | $HR_{@5}$ | $NDCG_{@5}$ | $MRR$ | $AUC$ |
| $mp_1$ | 0.6336 | 0.5184 | 0.5247 | 0.9077 | 0.5871 | 0.4166 | 0.3933 | 0.8909 |
| $mp_2$ | 0.6027 | 0.5013 | 0.4735 | 0.8876 | 0.4559 | 0.3184 | 0.3115 | 0.8604 |
| $mp_3$ | 0.6292 | 0.5125 | 0.5179 | 0.9016 | 0.6058 | 0.4218 | 0.3954 | 0.8937 |
| $mp_4$ | 0.6125 | 0.5087 | 0.4893 | 0.8927 | 0.4456 | 0.3123 | 0.3072 | 0.8576 |
| $mp_{1\&2}$ | 0.6748 | 0.5689 | 0.5564 | 0.9194 | 0.5655 | 0.3919 | 0.3699 | 0.9058 |
| $mp_{1\&3}$ | 0.6984 | 0.6134 | 0.6083 | 0.9236 | 0.5969 | 0.4423 | 0.4353 | 0.9185 |
| $mp_{1\&4}$ | 0.6851 | 0.5868 | 0.5732 | 0.9204 | 0.5828 | 0.4025 | 0.3795 | 0.9077 |
| $mp_{2\&3}$ | 0.6927 | 0.6059 | 0.5913 | 0.9227 | 0.6502 | 0.4723 | 0.4439 | 0.9161 |
| $mp_{2\&4}$ | 0.6624 | 0.6106 | 0.5718 | 0.9176 | 0.4934 | 0.3466 | 0.3353 | 0.8586 |
| $mp_{3\&4}$ | 0.6858 | 0.6048 | 0.5883 | 0.9208 | 0.5257 | 0.3697 | 0.3575 | 0.8993 |
| $mp_{1\&2\&3}$ | 0.7279 | 0.6364 | 0.6360 | 0.9311 | 0.7162 | 0.5464 | 0.5168 | 0.9386 |
| $mp_{1\&2\&4}$ | 0.7167 | 0.6222 | 0.6207 | 0.9246 | 0.6727 | 0.5285 | 0.4673 | 0.9199 |
| $mp_{1\&3\&4}$ | 0.7214 | 0.6347 | 0.6301 | 0.9302 | 0.7106 | 0.5463 | 0.5191 | 0.9367 |
| $mp_{2\&3\&4}$ | 0.7125 | 0.6208 | 0.6264 | 0.9274 | 0.6924 | 0.5307 | 0.4916 | 0.9213 |
| $mp_{1\&2\&3\&4}$ | **0.7542** | **0.6708** | **0.6637** | **0.9494** | **0.7851** | **0.6118** | **0.5766** | **0.9583** |

TABLE III: Results of different meta-path combinations

| Method | MOOCCube | | | | XuetangX | | | |
|---|---|---|---|---|---|---|---|---|
| | $HR_{@5}$ | $NDCG_{@5}$ | $MRR$ | $AUC$ | $HR_{@5}$ | $NDCG_{@5}$ | $MRR$ | $AUC$ |
| *Ave*-Fusion | 0.6338 | 0.5387 | 0.5292 | 0.8852 | 0.3923 | 0.2602 | 0.2602 | 0.8506 |
| *Loc*-Fusion | 0.7364 | 0.6504 | 0.6426 | 0.9414 | 0.7247 | 0.5552 | 0.5204 | 0.9475 |
| *Con*-Fusion | **0.7542** | **0.6708** | **0.6637** | **0.9494** | **0.7851** | **0.5833** | **0.5491** | **0.9583** |

*2) Comparison of Different Fusion Functions:* In order to verify the efficiency of the proposed fusion function, we evaluate the recommendation task when using different fusion functions in the two datasets, including *Ave*-Fusion that takes an average of each vector, *Loc*-Fusion that is a location-based fusion [11], and *Con*-Fusion is a concatenation-based fusion designed in our method. Table III turns out that in both datasets, the effect of *Ave*-Fusion is much worse than the others. *Con*-Fusion works the best, the $AUC$ of *Con*-Fusion is 0.8% higher than *Loc*-Fusion in MOOCCube, and 1.08% higher than *Loc*-Fusion in XuetangX, which means that concatenation-based fusion designed in our method can gather more associated information from different meta-paths.

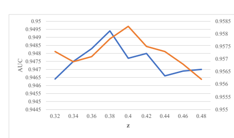


Fig. 3: The performance of different $z$ in two datasets

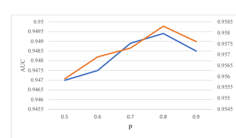


Fig. 4: The performance of different $p$ in two datasets

*3) Evaluation of Model Parameters:* In DCCR, the triad threshold $z$ is an important parameter. Fig. 3 shows how $z$ effect the result. The best results in the two datasets are obtained when $z$ is equal to 0.38 and 0.42, respectively. When $z$ is too small, too much useless correlation information may be got, which will affect the performance of the method. While if $z$ is too large, there will be a lack of correlation information leading to a bad performance.

In addition, in the proposed fusion function, we consider the impact of different values of scale parameter $p \in [0, 1]$, Fig. 4 shows the results. When $p = 0.8$, the best results are obtained in both datasets. If $p$ is too small, the effect will be reduced due to insufficient attention to the representation itself. If too large, it will be hard to fuse the association with the other meta-paths.

### D. Comparison with Other Methods

We compare with the following methods. *MLP* [2] that applies a multi-layer perceptron to user representations and the target knowledge concept representations, *FISM* [1] that is an item-to-item CF method, and conducts the recommendation task with the embeddings of users' history behaviors and the corresponding concept, *NAIS* [7] that is also an CF method with an attention mechanism to distinguishe the weights of different online learning behaviors, *ACKRec* [16] that is an attentional graph neural network in a heterogeneous view. For MLP, FISM, and NAIS, we construct the rating matrix and interaction histories between users and concepts from datasets. For ACKRec, we construct the corresponding features and adjacency matrices as inputs based on its steps. We select the most appropriate parameters to obtain the best results for a fair comparison.

From Tabel IV , it is apparent that the performance of DCCR is much better than MLP, FISM, NAIS in both datasets. The $AUC$ of DCCR is about 5.15% to 8.53% higher than MLP, FISM, and NAIS in MOOCCube, and 7.72% to 10.51% higher in XuetangX. Compared with ACKRec, DCCR extracts the correlation feature of user preference, which leads to a better performance. The $AUC$, $HR$@20, $NDCG$@20 and $MRR$ of DCCR are 2.06%, 6.22%, 3.62% and 4.7% higher than ACKRec in MOOCCube, respectively, and 3.51%, 10.39%, 9.73%, 4.92% higher in XuetangX. It has a lager growth from ACKRec to DCCR in XuetangX than in MOOCCube, which means that the correlation information in XuetangX is more abundant so that DCCR can make a bigger improvement.

TABLE IV: Results of different methods in Mooccube

| Methods | $HR$ | | | $NDCG$ | | | $MRR$ | $AUC$ |
|---|---|---|---|---|---|---|---|---|
| | @5 | @10 | @20 | @5 | @10 | @20 | | |
| **MOOCCube** | | | | | | | | |
| $MLP$ [2] | 0.4335 | 0.5744 | 0.7102 | 0.3562 | 0.3807 | 0.4088 | 0.3335 | 0.8651 |
| $FISM$ [1] | 0.5285 | 0.7411 | 0.7715 | 0.4826 | 0.5033 | 0.5288 | 0.4701 | 0.8684 |
| $NAIS$ [7] | 0.4957 | 0.6235 | 0.8497 | 0.2848 | 0.3651 | 0.4218 | 0.3563 | 0.8979 |
| $ACKRec$ [16] | 0.7125 | 0.8014 | 0.8827 | 0.6326 | 0.6622 | 0.6855 | 0.6015 | 0.9288 |
| **DCCR** | **0.7542** | **0.8539** | **0.9297** | **0.6708** | **0.7026** | **0.7217** | **0.6637** | **0.9494** |
| **XuetangX** | | | | | | | | |
| $MLP$ [2] | 0.3680 | 0.5899 | 0.7237 | 0.2231 | 0.2926 | 0.3441 | 0.2146 | 0.8595 |
| $FISM$ [1] | 0.5849 | 0.7489 | 0.7610 | 0.3760 | 0.4203 | 0.4279 | 0.3293 | 0.8532 |
| $NAIS$ [7] | 0.4112 | 0.6624 | 0.8649 | 0.2392 | 0.3201 | 0.3793 | 0.2392 | 0.8863 |
| $ACKRec$ [16] | 0.6470 | 0.8122 | 0.9255 | 0.4635 | 0.5170 | 0.5459 | 0.4352 | 0.9232 |
| **DCCR** | **0.7851** | **0.9063** | **0.9747** | **0.5833** | **0.6259** | **0.6432** | **0.5491** | **0.9583** |

## IV. Conclusions

This paper proposes a recommendation method named as DCCR which generates the knowledge concept recommendation list for users in MOOCs. DCCR captures the rich entity interactive information in MOOCs by the guide of meta-path and extracts the semantic features from text information along with concepts. It also gathers the correlation features of users among meta-paths by constructing a multi-relational graph. To better consider the association between different meta-paths, a concatenation-based fusion function is proposed to generate the final joint representation of users and concepts. By verifying the effectiveness of DCCR on two public datasets, the experimental results show that DCCR is superior to the existing methods.

## References


[1] S. Kabbur, X. Ning, G. Karypis, Fism: factored item similarity models for top-n recommender systems, In: ACM SIGKDD, 2013, pp. 659–667.

[2] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.S. Chua, Neural collaborative filtering, In: International Conference on World Wide Web, 2017, pp. 173–182.

[3] K. Järvelin, J. Kekäläinenm, Ir evaluation methods for retrieving highly relevant documents, In: ACM SIGKDD, 2017, pp. 41-48.

[4] M. Gori, G. Monfardini, F. Scarselli, A new model for learning in graph domains, In: IEEE International Joint Conference on Neural Networks, 2005, vol. 2, pp. 729–734.

[5] L. Pan, X. Wang, C. Li, J. Li, J. Tang, 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), 2017, pp. 875-884.

[6] A. Elbadrawy, G. Karypis, Domain-aware grade prediction and top-n course recommendation, In: ACM Conference on Recommender Systems, 2016, pp. 183–190.

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

[8] L. Pan, C. Li, J. Li, J. Tang, Prerequisite Relation Learning for Concepts in MOOCs, In: Annual Meeting of the Association for Computational Linguistics, 2017, pp. 1447–1456.

[9] Y. Pang, Y. Jin, Y. Zhang, T. Zhu, Collaborative filtering recommendation for mooc application, Computer Applications in Engineering Education **25**(1), 120–128, 2017.

[10] H. Zhang, M. Sun, X. Wang, Z. Song, J. Tang, J. Sun, Smart jump: Automated navigation suggestion for videos in moocs, In: International Conference on World Wide Web Companion,2 2017 pp. 331–339.

[11] M.T. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation, In: Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1412–1421.

[12] C. King, A. Robinson, J. Vickers, Targeted mooc captivates students, Nature **505**(7481), 26–26, 2014.

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

[14] C. Shi, B. Hu, W.X. Zhao, P.S. Yu, Leveraging meta-path based context for top-n recommendation with a neural co-attention model, In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1531–1540.

[15] S. Vashishth, S. Sanyal, V. Nitin, P. Talukdar, Composition-based multi-relational graph convolutional networks, In: International Conference on Learning Representations, 2020, pp. 1–15.

[16] J. Gong, S. Wang, J. Wang, W. Feng, H. Peng, J. Tang, P.S. Yu, Attentional graph convolutional networks for knowledge concept recommendation in moocs in a heterogeneous view, In: ACM SIGIR, 2020, pp. 79–88.

[17] J. Yu, G. Luo, T. Xiao, Q. Zhong, Y. Wang, J. Luo, C. Wang, L. Hou, J. Li, Z. Liu, J. Tang, J.: Mooccube: A large-scale data repository for nlp applications in moocs, In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3135–3142.

[18] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, In: ICLR Workshop, 2013.

[19] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, In: International Conference on Learning Representations, 2016.