



**HAL**  
open science

# A Method on Online Learning Video Recommendation Method Based on Knowledge Graph

Xin Chen, Yuhong Sun, Tong Zhou, Qingtian Zeng, Huafang Qi

► **To cite this version:**

Xin Chen, Yuhong Sun, Tong Zhou, Qingtian Zeng, Huafang Qi. A Method on Online Learning Video Recommendation Method Based on Knowledge Graph. 12th International Conference on Intelligent Information Processing (IIP), May 2022, Qingdao, China. pp.419-430, 10.1007/978-3-031-03948-5\_34 . hal-04178719

**HAL Id: hal-04178719**

**<https://inria.hal.science/hal-04178719v1>**

Submitted on 8 Aug 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

# A Method on Online Learning Video Recommendation Method Based on Knowledge Graph

Xin Chen<sup>1</sup>[0000-0003-4325-8629], Yuhong Sun<sup>1</sup>[0000-0001-6276-8531], Tong Zhou<sup>2</sup>[0000-0002-6374-8310],  
Qingtian Zeng<sup>1</sup>[0000-0002-6421-8223], Huafang Qi<sup>1</sup>[0000-0002-1067-8195]

<sup>1</sup>College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, P.R.China

<sup>2</sup>College of Civil Engineering and Architecture, Shandong University of Science and Technology, Qingdao 266590, P.R.China

Corresponding author: Xin Chen (xxwar@163.com) and Tong Zhou (768279353@qq.com).

**Abstract.** The MOOC platforms usually recommend online course resources to users among a large number of courses on platform whose users are almost collage students or students by selves. However, the entire course as the recommendation results may ignore student interests in some specific knowledge points, and so may reduce student learning interests. The recommendation not courses but online learning videos of course are researched. The connection between a single learning video and other entities is considered to construct learning video knowledge graph. Based on the end-to-end deep learning framework, a convolutional neural network KGCN-LV is applied in learning videos recommendation, which is integrated learning video knowledge graph. Experimental results on public data set MOOCCube show that the learning video recommendation method based on knowledge graph is up to 5% better than the traditional collaborative filtering method. The recommendation performance is effectively improved, and the recommendation effect is interpretable.

**Keywords:** *Recommendation System, Knowledge Graph, Convolutional Neural Network, Learning video*

## 1 Introduction

### 1.1 A SubsectionSample

The popularity and development of MOOC platforms (such as XuetangX, Chinese University MOOC, iCourse, etc.) has promoted development to Online education. Students usually study by the way of online learning on Internet especially during the

COVID-19 pandemic which is gradually becoming a popular education mode. Faced with massive online course resources, it becomes extremely difficult for students to choose the courses they need, which also reduces student interests in learning. The current overall course completion rate for MOOC platforms is less than 5%. In order to capture and understand student interests on the MOOC platforms, researchers have done a lot of work, including course recommendation [1], behavior prediction [2], user intention understanding [3], etc. In these works, the MOOC platform applies a recommendation system to recommend courses to students. However, a course is composed of lots of knowledge points, and a course online usually contains several-learning videos. Each video is focus on a specific knowledge point and the length of time is about 10-20 minutes. The whole course recommendation may ignore student interests in some specific knowledge points. Each course is constructed by different teacher or different university and so there are some similar knowledge points between different courses. For example, if a student wants to learn “the loss function of the neural networks”, the course recommendation maybe recommends the entire course to him or her but the videos about “the loss function of the neural networks” cannot be focused on. He or she has to find the videos corresponding from the course indexes. The whole process is tedious and the student may lose interest. Therefore, MOOC platforms need to capture student learning interests and the recommendation results are not the courses but the specific videos which contains knowledge points that students just need.

Traditional recommendation strategies, such as collaborative filtering (CF), utilize user-item (student-learning video) historical interactions and make recommendations based on the common preferences of potential users (students). However, CF has problems of data sparsity and cold start, which limit the performance of the recommendation. In order to solve this problem, the recommended method considers the integration of auxiliary information (such as social network [4], user project attributes [5], and context [6]) into collaborative filtering, and has achieved good results. On the MOOC platform, we observe that in addition to users and learning videos, there are also relationships between entities of various types, such as schools, teachers, courses and knowledge points. Based on these relationships, rich semantic information between entities can be obtained. Therefore, this paper considers the construction of Learning Video Knowledge Graph (LVKG), which can be used as auxiliary information to capture rich semantic information between different types of entities and incorporate them into the presentation learning process. Convolutional Networks (KGCN-LV), an end-to-end Learning Video Knowledge Graph framework proposed in this paper, aggregates the neighborhood information of a given entity representation when computing it. To conduct high-level modeling to capture students' potential interests. KGCN-LV uses knowledge graph as auxiliary information to recommend learning videos to students, which effectively improves data sparsity and cold start problems in recommendation and makes recommendation interpretable.

The main contributions of this paper are as follows. Firstly, we construct the learning video knowledge graph (LVKG) from the MOOC platform to capture the auxiliary information from entities and connections to identify the potential interactions. Se-

condly, a convolutional neural network based on the knowledge graph of learning video (KGCN-LV) is proposed, and the experiments on MOOCube public dataset contains the methods of KGCN-LV, SVD, LibFM, RippleNet, MKR.

## 2 Related Work

### 2.1 Traditional Recommendation Strategy

Traditional recommendation strategies are mainly collaborative filtering (CF), which considers the historical interaction between students and learning videos and recommends student common preferences based on potential similar interests, such as matrix decomposition method [7] and factor decomposition method libFM [8]. CF has effectiveness and universality in practical recommendation scenes, but it cannot model auxiliary information, such as learning video attributes, student profiles and context. Therefore, the recommendation effect is extremely poor when the interaction matrix between students and courses is extremely sparse. In order to solve this problem, the neural network method is introduced to convert student ID and learning video ID into a general feature vector, which is input into the supervised learning model to predict scores, such as neural network-based collaborative filtering recommendation NCF[9], neural factor decomposition machine NFM[10], and Wide&Deep model [11]. These methods all provide powerful performance for recommendation systems, but they model each student course interaction as an independent data instance without considering the relationships between them, which makes them inadequate to extract collaborative signals from students' collective behavior.

### 2.2 Existing KG-aware methods

As a semantic network, knowledge graph has strong expression ability and modeling flexibility, and can model entities, concepts, attributes and their relationships in the real world [12]. Properly utilizing knowledge graph to model the real world can make computers have a stronger ability to understand the real world. The concept of knowledge graph was proposed and released by Google on May 17, 2012, and it announced to build the next generation of intelligent search engine based on the knowledge graph. The key technologies include extracting entities, entity attribute information and relationship between entities from the Web pages of the Internet [13]. Recently, researchers have proposed several academic knowledge graphs, such as NELL, DBpedia and Google Knowledge Graph and Microsoft Satori. The knowledge graph method used in recommendation systems is now used in films, books, music, news and other fields. Collaborative Knowledge Base Embedding (CKE)[14] combines collaborative filtering (CF) module with knowledge embedding, text embedding and image embedding of projects in a unified Bayesian framework. CKE is suitable for knowledge graph completion and link prediction, but not for recommendation. Based on deep knowledge perception network (DKN)[15], entity embedding and word embedding are regarded as different channels, and then the convolutional neural network (CNN) framework is used to combine them together for news recommendation. However,

before using DKN, entity embedding is required, which results in the lack of end-to-end training mode for DKN. Another disadvantage of DKN is that it contains little information other than text. RippleNet[16] is a model similar to memory network, which propagates users' potential preferences in KG and explores their hierarchical interests. However, the importance of relationships in RippleNet is weakly characterized. MKR[17] is a multi-task deep learning framework for knowledge graph embedding tasks to assist in enhancing recommendation tasks. In the recommendation process, entities and projects update their vector representations to each other. Ignoring the importance of relationships in the knowledge graph.

### 3 Model and Method

A learning video knowledge graph LVKG is constructed to support the KGCN-LV model for the learning video recommendation.

#### 3.1 Problem Formulation

The single learning video recommendation problem is to recommend learning videos (projects) to students (users). In the course recommendation scenario, there is a group of  $M$  students as  $\mathcal{U} = \{u_1, u_2, \dots, u_M\}$  and a group of  $N$  learning videos  $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ . The interaction matrix is constructed according to the implicit feedback of students, which is expressed as  $Y = \{y_{uv} | u \in \mathcal{U}, v \in \mathcal{V}\}$ , if students interact with the learning video (such as learning, browsing, etc.),  $y_{uv} = 1$ , otherwise  $y_{uv} = 0$ , which is

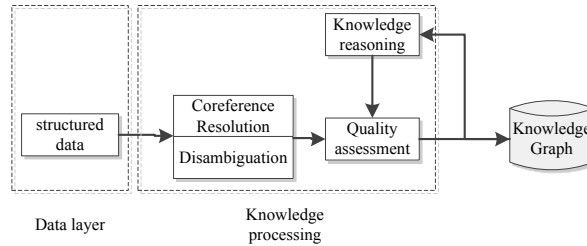
$$y_{uv} = \begin{cases} 1, & \text{if interaction}(u, v) \text{ is observed} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The learning video knowledge graph is composed of a large number of entity-relation-entity triples  $(h, r, t)$ , where  $h \in \mathcal{E}, r \in \mathcal{R}, t \in \mathcal{E}$  represent the head entity, relation and tail entity in the knowledge graph.  $\mathcal{E}$  and  $\mathcal{R}$  respectively represent the entity set and relationship set in the knowledge graph. For example, the triple (V\_e88505, teacher, T\_Liu Boli) means that the teacher of the learning video 'V\_e88505' is Liu Boli.

The purpose of the learning video recommendation problem is to predict whether the student  $u$  is potentially interested in the learning video  $v$  that has not been interacted before, given the interaction matrix  $Y$  and the knowledge graph  $\mathcal{G}$ . The goal of the model is to learn  $\hat{y}_{uv} = F(u, v; \Theta)$ ,  $\hat{y}_{uv}$  represents the possibility that student  $u$  clicks on the learning video  $v$ , and  $\Theta$  represents the model parameter of the function  $F$ .

### 3.2 Construction of KG for Learning Videos

This paper adopts a bottom-up construction method to add structured data in the MOOCCube dataset[18] to the data layer, and then process these knowledge elements to construct a knowledge graph LVKG.KG construction method is shown as Fig 1.



**Fig. 1.**KG construction method

The data used in this article already has a certain data structure. For preliminary knowledge representation, knowledge processing is required. For example, for the same entity, there may be different descriptions, and the different descriptions of entities can be accurately and completely expressed by the coreference resolution. When constructing a knowledge graph, entities with the same name may be ambiguous. For example, the entity "operating system" can be either a knowledge point or the name of a computer course. The entity disambiguation is used to accurately identify it by understanding the context. After data knowledge processing, a series of examples are obtained. In order to obtain a more accurate knowledge system than before, quality evaluation and knowledge reasoning are carried out, and finally the construction of LVKG is completed.

For the actual application scenario of the MOOC platform, the extracted entities include video id, course id, school name, teacher name, and related knowledge points. The relationship includes the course, school, teacher, and knowledge points to which the video belongs. The data fragment is shown as follows as table 1. According to the knowledge graph construction method, the triple form of the knowledge graph LVKG is obtained as shown as Table 2. For example, a triple (V-297bc3, knowledge point, KMP algorithm) means that 'KMP algorithm' is the knowledge point contained in video 'V-297bc3'.

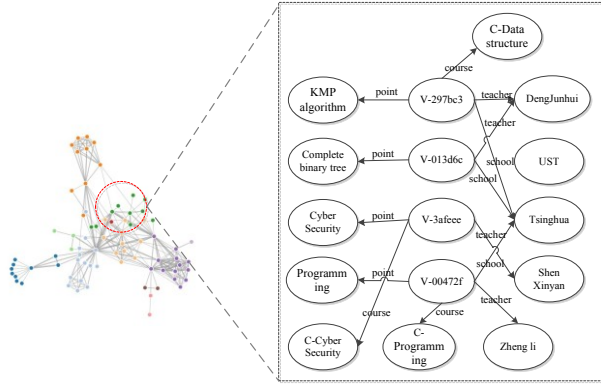
**Table 1.**MOOCCube Data fragment

Video	Point	Teacher	School	Course
V_a37950	KMP algorithm	T_DengJunhui	S_TsinghuaX	Data structure
V_7b6672	White box testing	T_JohnGuttag	S_MITx	Software test
V_4ab7bf	Encoder	T_LiuBoli	S_HBNU	Programming
V_c1857a	Concurrency	T_Chen Yu	S_TsinghuaX	Programming
V_40268a	Operating system	T_Xiang Yong	S_TsinghuaX	Operating system
...	...	...	...	...

**Table 2.**Triple of KG.

Head entity	Relation	Tail entity
V_a37950	points	KMP algorithm
V_7b6672	teacher	T_JohnGutttag
V_4ab7bf	teacher	T_LiuBoli
V_c1857a	school	S_TsinghuaX
V_40268a	course	Operating syste
V_40268a	teacher	T_Xiang Yong
...	...	...

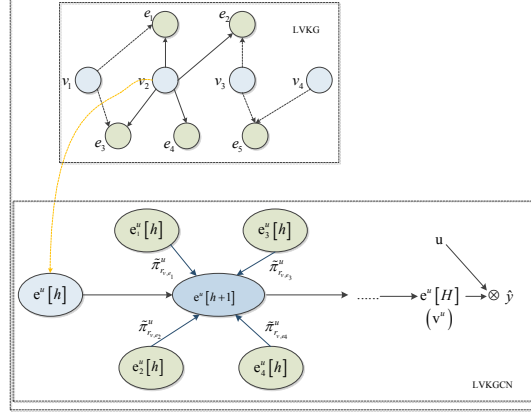
According to the obtained triples, we use the graph database Neo4j to visualize the knowledge graph LVKG shown in Fig. 2.

**Fig. 2.**Learning Video Knowledge Graph-LVKG

### 3.3 Framework

In this paper, the KGCN-LV model is proposed to implement student-oriented learning video recommendation. The model captures the high-order connections between entities in the knowledge graph, aggregates the entity representation of learning video and its neighborhood nodes (green nodes), and forms the representation of the next iteration (dark blue nodes). The KGCN-LV model is shown in Fig. 3.





**Fig. 3.** Framework of KGCN-LV

In the first-order connection of KGCN-LV, for student  $u$  and learning video (entity)  $v$ , use  $N(v)$  to represent the entity set directly connected to  $v$ , and  $r_{e_i, e_j}$  to represent the relationship between  $e_i$  and  $e_j$ . The model uses the function  $g: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$  to calculate the student's interest in the relationship. For example, in the learning process, the student pays more attention to which teacher teaches. Here, the inner product of the student and the relationship is used to express the interest.

$$\pi_r^u = g(\mathbf{u}, \mathbf{r}), \quad (2)$$

$\mathbf{u} \in \mathbb{R}^d$  and  $\mathbf{r} \in \mathbb{R}^d$  are the vector representation of the student  $u$  and the relation  $r$ , and  $d$  is the vector dimension. In order to characterize the neighboring structure of the learning video, the linear combination of its neighbors is expressed as:

$$\mathbf{v}_{N(v)}^u = \sum_{e \in N(v)} \tilde{\pi}_{r_{v,e}}^u \mathbf{e}, \quad (3)$$

and  $\tilde{\pi}_{r_{v,e}}^u$  is the normalized representation of the user's interest in the relationship,

$$\tilde{\pi}_{r_{v,e}}^u = \frac{\exp(\pi_{r_{v,e}}^u)}{\sum_{e \in N(v)} \exp(\pi_{r_{v,e}}^u)}, \quad (4)$$

where  $\mathbf{e}$  is the vector representation of the entity  $e$ . When calculating the neighborhood representation of the entity,  $\tilde{\pi}_{r_{v,e}}^u$  is used as a personalized filter to capture the learning interest of students. Finally, the entity representation and its neighborhood representation are aggregated into a single vector, and the two representation vectors are added together to perform a nonlinear transformation.

The final representation of an entity is its own representation and its neighborhood representation. In order to maintain computational efficiency, a fixed-size neighbor is used for each entity. The first-order representation is extended to the multiple-order, and user preferences are explored in more depth. The formal description of the above

steps is presented in Algorithm 1.  $H$  denotes the maximum depth of receptive field (orequivalently, the number of aggregation iterations), and a suffix  $[h]$  attached by a representation vector denotes  $h$ -order. For a given user-item pair  $(u, v)$  (line 2), we first calculate the receptive field  $\mathcal{M}$  of  $v$  in an iterative layer-by-layer manner (line 3,11-16).The final  $H$ -order entity representation is denoted as  $v^u$  (line 5), which is fed into a function  $F$  together with user representation  $u$  for predicting the probability  $\hat{y}_{uv}$ .

---

**Algorithm 1:** KGCN-LV algorithm

---

**Input:** Interaction matrix  $Y$  and the knowledge graph  $\mathcal{G}$ .

**Output:** Prediction function  $\hat{y}_{uv} = F(u, v; \Theta)$

```

1 while KGCN not converge do
2   for  $(u, v)$  in  $Y$  do
3      $\mathcal{M}[i]_{i=0}^H \leftarrow$  Get Receptive Field( $v$ );
4      $e^u[0] \leftarrow e, \forall e \in \mathcal{M}[0]$ ;
5      $v^u \leftarrow e^u[H]$ ;
6     Calculate predicted probability  $\hat{y}_{uv}$ ;
7     Update parameters by gradient descent;
8   end for
9 end while
10 return  $F$ 
11 Function Get Receptive Field( $v$ );
12    $\mathcal{M}[H] \leftarrow v$ 
13   for  $h = H - 1, \dots, 0$ 
14      $\mathcal{M}[h] \leftarrow \mathcal{M}[h + 1]$ 
15   end for
16 return  $\mathcal{M}[i]_{i=0}^H$ ;

```

---

In order to improve computational efficiency, a non-negative sampling strategy is used during training [19], and the complete loss function is as follows:

$$\mathcal{L} = \sum_{u \in \mathcal{U}} (\sum_{v: y_{uv}=1} \mathcal{J}(y_{uv}, \hat{y}_{uv}) - \sum_{i=1}^{T^u} \mathbb{E}_{v_i \sim P(v_i)} \mathcal{J}(y_{uv_i}, \hat{y}_{uv_i})) + \lambda \|\mathcal{F}\|_2^2, \quad (5)$$

Where  $\mathcal{J}$  is the cross-entropy loss function,  $P$  is the negative sampling distribution, and  $T^u$  is the negative sampling number of student  $u$ . In this article,  $T^u = |\{v: y_{uv} = 1\}|$  and  $P$  obey a uniform distribution, the last term is the L2 regularization term to prevent overfitting.  $\lambda$  is the balance parameter.

## 4 Experiment

### 4.1 Dataset

The experiment uses the Q&A dataset in MOOCCube. The MOOCCube dataset is a large-scale data repository containing more than 700 MOOC courses, 38k videos, 200,000 students, 100k points, and 300k relationship examples. The large data set supports model experiments well. The twelve courses in the Q&A dataset all come

from the field of computer science. They cover different levels of computer science courses.

According to the recommendation algorithm, the four attributes of the data set, such as video id, teacher, course id, school, and knowledge points, used in the experiment are extracted. The basic statistics of the data set are shown in Table 3.

**Table 3.** Basic statistics of the dataset.

dataset	#students	#videos	#interactions	#entities	#relation	#KG triples
MOOCCube	4361	1401	450724	4700	4	8777

The experimental environment is Intel i7-6700HQ CPU @ 2.60 GHz, 16 GB memory, Windows 10 operating system, and the software tools include PyCharm, Anaconda3 and TensorFlow framework.

## 4.2 Baselines

SVD[7] decomposes the high-dimensional student course scoring matrix into low-dimensional student feature vector matrix, course feature matrix and diagonal matrix of singular values. The parameters can be updated according to the existing scoring data, once the student eigenvector matrix and the course eigenvector matrix is acquired. SVD takes the student's course scoring matrix as input.

LibFM[8] is a widely used feature-based decomposition model that takes the original features of students and courses as the input of LibFM. The dimension is  $\{1, 1, 8\}$ , and the number of training epochs is 50.

RippleNet[16] is a method similar to a memory network, which can spread student's preferences on the KG for a recommendation. The hyperparameters are set to  $d=8, H=2, \lambda_1 = 10^{-6}, \lambda_2=0.01, \eta = 0.02$ .

MKR [17] is a multi-task feature learning method for knowledge graph enhancement recommendation, and knowledge graph embedding tasks can be used to assist the recommendation.

## 4.3 Experiments Setup

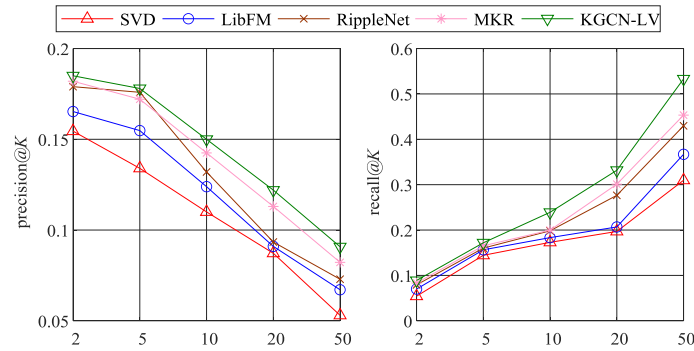
The experiment adopts a negative sampling strategy to establish a student-learning video interaction matrix. This matrix is a sparse matrix. The positive examples are examples of students interacting with the learning video, and the negative examples are sampled from examples that the students have not interacted with. The ratio of training set, validation set to test set is 6:2:2. Each experiment is repeated four times, and the average performance is reported. The method is evaluated in two experimental scenarios: (1) In click-through rate (CTR) prediction, we apply the trained model to predict each interaction in the test set. We use AUC and ACC to evaluate CTR prediction. (2) In top-K recommendation, we use the trained model to select K items with highest predicted click probability for each user in the test set, and choose Precision@K, Recall@K to evaluate the recommended sets. All trainable parameters are optimized by Adam algorithm. For KGCN-LV, set functions  $g$  and  $f$  as inner product,  $K = 4$ , aggregator as sum, batch size=65536, and other hyperparameters  $H = 2$ ,  $d=32$ ,  $\lambda = 10^{-7}$  are validation set is determined by optimizing AUC.

#### 4.4 Results

CTR prediction and top-K recommendation results of all methods are presented in Table 4 and Fig.4 respectively. It can be observed from the experimental results: (1) The experimental effect of MKR and RippleNet models that add knowledge graph to learning video recommendation is 3% higher than that of SVD and LibFM, and they can perform well even in the case of sparse student course interaction. This shows that the use of knowledge graphs as auxiliary information captures rich semantic information between different types of entities, and effectively improves the data sparseness and cold start problems in the recommendation system. (2) KGCVN-LV performed the best, improved by about 1% compared to the strongest baseline MKR. MKR is a multi-task learning method recommended for knowledge graph enhancement. It does not aggregate the neighborhood information between knowledge graph entities, but only represents the entity itself. KGCVN-LV aggregates the neighborhood information of learning video entities and enriches the representation of the entities themselves. This shows that when computing the representation of a given entity, the neighborhood information is aggregated to perform high-level modeling to capture the potential interest of students and enhance the recommendation effect.

**Table 4.** Learn videos click rate prediction probability.

Model	AUC	ACC
SVD	0.9322	0.9113
LibFM	0.9303	0.9045
RippleNet	0.9611	0.9246
MKR	0.9665	0.9270
KGCVN-LV	<b>0.9753</b>	<b>0.9319</b>



**Fig. 4.** top-K recommendation *Persicion@K* and *Recall@K*

## 5 Conclusions and Future Work

In order to solve the problem that student interest in specific knowledge points may be ignored for course recommendation on the MOOC platform, as well as the data sparsity and cold start problems of traditional recommendation algorithms CF. A single online learning video recommendation method using knowledge graph is researched. The connection between a single learning video and other entities, such as courses, teachers, schools, knowledge points is considered to build a learning video knowledge graph LVKG. A convolution neural network KGCN-LV is proposed, which integrates learning video knowledge graphs. By gathering neighborhood information, it can obtain potential personalized student interests. Through experiments on MOOCube, KGCN-LV is always better than the baseline. In future work, the establishment of a student-end knowledge graph would be considered to improve the recommendation performance, and the optimized algorithm is designed to combine the knowledge graphs between the student-end and the course-end.

## Acknowledgement

This work is partially supported by the Open Fund of the National Virtual Simulation Experimental Teaching Center for Coal Mine Safety Mining (SDUST 2019), the 2018 Postgraduate Tutors' Guidance Ability Improvement Project of Shandong Province (SDYY18084), the Teaching Reform Research Project of the Teaching Steering Committee of Electronic Information Specialty in Higher Education and Universities of the Ministry of Education, the Special Project of China Association of Higher Education, the Education and Teaching Research Project of Shandong Province, and Excellent Teaching Team Construction Project of Shandong University of Science and Technology.

## References

1. Zhang J, Hao B, Chen B, Li C, Chen H, and Sun J. Hierarchical reinforcement learning for course recommendation in MOOCs[C]//The Thirty-Third AAAI Conference on Artificial Intelligence, 2019. DOI: 10.1609/aaai.v33i01.3301435.
2. Shu-Tong Xie, Zong-Bao He, Qiong Chen, Rong-Xin Chen, Qing-Zhao Kong, Cun-Ying Song, "Predicting Learning Behavior Using Log Data in Blended Teaching", *Scientific Programming*, vol. 2021, Article ID 4327896, 14 pages, 2021. <https://doi.org/10.1155/2021/4327896>
3. Zhang H, Song Z, Sun M, Tang J, Wang X, and Sun J. Smart jump: Automated navigation suggestion for videos in MOOCs[C]//International World Wide Web Conference Committee, 2017. DOI: 10.1145/3041021.3054166.

4. Liu J, Fu L, Wang X, Tang F, and Chen G. Joint Recommendations in Multilayer Mobile Social Networks[J]. IEEE Transactions on Mobile Computing. 2020, DOI: 10.1109/TMC.2019.2923665.
5. Wang H., Zhang F, Hou M, Xie X, Guo M, and Liu Q. SHINE: Signed heterogeneous information network embedding for sentiment link prediction[C]// Proceedings of the 11th ACM International Conference on Web Search and Data Mining, 2018,2018-Febua(592-600)
6. Xie R, Qiu Z, Rao J, et al. Internal and Contextual Attention Network for Cold-start Multi-channel Matching in Recommendation[C]//IJCAI. 2020: 2732-2738.
7. Matrix N F. SVD Recommendation Model Based on Positive and Negative Feedback Matrix[J].2015.
8. Rendle S. Factorization machines with libFM[J].ACM Transactions on Intelligent Systems and Technology, 2012. DOI: 10.1145/2168752.2168771.
9. He X, Liao L, Zhang H, Nie L. Hu X, and. Neural collaborative filtering[C]//26th International World Wide Web Conference, WWW 2017, 2017.
10. He X. and Chua T S. Neural factorization machines for sparse predictive analytics[C]//International Conference on Research and Development in Information Retrieval, 2017.DOI:10.1145/3077136.3080777
11. Cheng H T, et al. Wide & deep learning for recommender systems[J]. ACM International Conference Proceeding Series, 2016. DOI: 10.1145/2988450.2988454.
12. Qi G L. The Research Advances of Knowledge Graph[J]. Technology Intelligence Engineering, 2017.DOI:10.3772/j.issn.2095-915x.2017.01.002
13. Qiao L. Knowledge Graph construction techniques[J] Journal of Computer Research and Development, 2016, 53(3):582-600
14. Zhang F, Yuan N J, Lian D, Xie X, Ma W Y. Collaborative knowledge base embedding for recommender systems[C]//Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, 13-17(353–362).
15. Wang H, Zhang F, Xie X, Guo M. DKN: Deep knowledge-aware network for news recommendation[C]//International World Wide Web Conference Committee, 2018, 2(1835–1844). DOI: 10.1145/3178876.3186175.
16. Wang H, et al. RippleNet: Propagating user preferences on the knowledge graph for recommender systems[C]//International Conference on Information and Knowledge Management, 2018, 1(417–426). DOI:10.1145/3269206.3271739
17. Wang H, Zhang F, Zhao M, Li W, Xie X, Guo M. Multi-task feature learning for knowledge graph enhanced recommendation[C]//International World Wide Web Conference Committee, 2019, 2(2000–2010). DOI: 10.1145/3308558.3313411.
18. Yu J, et al. MOOCCube: A Large-scale Data Repository for NLP Applications in MOOCs[C]//Annual Conference of the Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.acl-main.285
19. Wang H, Zhao M, Xie X, Li W, Guo M. Knowledge graph convolutional networks for recommender systems[C]//World Wide Web Conference, 2019. DOI: 10.1145/3308558.3313417.