A Collaborative Graph Convolutional Networks and Learning Styles Model for Courses Recommendation

Junyi Zhu[1], Liping Wang[1], Yanxiu Liu[1,2], Ping-Kuo Chen[3](✉),
and Guodao Zhang[4](✉)

[1] College of Computer Science and Technology, Zhejiang University of Technology,
Hangzhou 310023, China
[2] School of Data and Computer Science, Shandong Women's University, Jinan 250300, China
[3] Great Bar University, Dongguan 523000, China
a1104100@ms23.hinet.net
[4] School of Media and Design, Hangzhou Dianzi University, Hangzhou 310018, China
guodaozhang@zjut.edu.cn

Abstract. With the rise of Massive Open Online Courses (MOOCs) and the deepening of lifelong learning, there is a growing demand for learners to learn on online learning platforms. The vast amount of course resources provides learners with massive and easy access while posing challenges in terms of personalized and precise selection. Traditional recommendation models have room for improvement in performance and interpretability in massive open online course scenarios while under-utilizing the potential interaction signals in user-course interactions and ignoring the impact of the user's learning style as a learner. In order to solve the above problems, this paper proposes a collaborative graph convolutional networks and learning styles model for courses recommendation (CGCNLS). First, the course prediction rating is obtained by propagating the learner-course interaction information recursively through the graph convolutional networks; further, a course and learning styles matching scale is created to calculate the course learning styles similarity score; finally, the course prediction rating is combined with the course learning styles similarity score to make personalized course recommendations. The experimental results show that the model proposed in this paper can effectively recommend courses for learners and outperforms the baseline approach in terms of Precision, Recall, and NDCG performance metrics.

Keywords: Graph neural networks · Learning styles · Course recommendation · Collaborative models

---

This work was supported by the National Natural Science Foundation of China grant numbers 71872131 and Starting Research Fund of Great Bay University under grant YJKY220020 and Research Foundation of Hangzhou Dianzi University (KYS335622091; KYH333122029M).

H. Gao et al. (Eds.): CollaborateCom 2022, LNICST 460, pp. 360–377, 2022.
https://doi.org/10.1007/978-3-031-24383-7_20

1 Introduction

Nowadays, Massive Open Online Courses (MOOCs) are attracting the interest of many learners as an emerging educational model [1, 2]. With many different MOOC platforms in place, not only is the cost of learning reduced for learners, but they can also access quality courses from top universities around the world. Online education has greatly promoted the development of part-time learners. Still, those learners tend to be self-directed, lack guidance, and learn inefficiently, making personalized course recommendations particularly important for improving their learning efficiency. Therefore, scholars [3, 4] have devoted themselves to studying personalized learning resource systems, and customized learning resource recommendations include educational courses, learning paths, exercises, learning peers, etc.

In general, course recommendation methods are divided into course recommendation models based on traditional algorithms and course recommendation models based on deep learning techniques. Traditional course recommendation models use collaborative filtering (CF) techniques, which measure the similarity between users or courses to predict and recommend content that may interest users. In addition, with the development of deep learning techniques, deep networks are gradually being applied to educational resource recommendation. Many effective deep network recommendation models have been proposed to model user preferences in different ways. Deep learning techniques usually represent users and items as low-dimensional Embedding vectors [5–7] and iteratively optimize the parameters of the deep network and the Embedding vectors of users and items based on user behavior data or item information. Then, personalized recommendations are made by calculating the prediction rating of users and items or by feeding the embedding into the deep network to obtain the prediction rating of users for items to be recommended. Although the above methods are effective, the process of generating embedding vectors of users and courses lacks information to encode the key collaboration information, which is hidden in the interaction behavior of users and courses. This information can reveal the behavioral similarity between users/items. Specifically, CF approaches use only descriptive features (e.g., IDs and attributes) to construct embedding functions without considering the effects arising from user-course interactions. These interactions are used only to define the objective function for model training (e.g., inner product) without applying them to the process of generating embedding vectors.

With the development of online open courses and information technology, a large amount of data is easily accessible and stored, and with it comes the challenge of learners' privacy. Process behavior data mainly refers to human-computer interaction data, i.e., information about learners' course selection, learners' viewing of learning resources, etc., which can effectively protect personal privacy. Piao et al. [8] used processive data to construct Meta-Path-based graph convolutional networks for learning resource recommendation. Sheng et al. [9] used to process data to construct a heterogeneous information network-based model for online course recommendation and achieved good performance. It follows that learner modeling through processual behavioral data can effectively address individualized learner needs, i.e., recommending courses or learning resources that meet learners' needs.

In summary, this paper proposes a collaborative graph convolutional networks and learning styles course recommendation model to solve the above problem. The main work is as follows:

1. Construct an end-to-end graph convolutional network model to propagate learner-course interaction information in a recursive form, and apply higher-order interaction information to the embedding vector generation process to better compute the corresponding predicted course rating of learners.
2. To create a matching scale of course and learning styles based on pedagogical learning styles theory, and to obtain learners' corresponding course learning styles similarity score by cosine similarity calculation.
3. Collaborative prediction rating are obtained by graph convolutional network course prediction rating and course learning styles similarity score for personalized course recommendation. Multi-group experimental analysis in the real-world dataset shows that the course learning styles similarity score mechanism proposed in this paper can effectively improve the model performance, and the performance of the CGCNLS model is significantly better than that of the benchmark models.

The remainder of this paper is organized as follows. Section 2 reviews the latest relevant work. Section 3 describes the proposed approach in this paper. In Sect. 4, the experimental results and analysis are shown. In Sect. 5, a summary and outlook of this paper are presented.

2 Related Work

Learning style-based recommendation models have been a hot research topic in recent years [10–12]. For example, Hajri et al. [13] create learner learning style profiles based on learner profiles and dynamically provide learning resource recommendations to learners based on MOOC attributes. Sanjabi et al. [14] conducted a study on the personalization of e-learning environments based on the Kolb learning style model. Yan et al. [15] integrate learning style features into collaborative filtering algorithms for association rule mining. Sensuse et al. [16] surveyed personalization strategies based on Felder-Silverman learning styles and their impact on learning, finding insufficient theoretical research and a lack of user relevance studies in the literature. These methods have experimentally demonstrated the effectiveness of learning styles for the problem of learning resource recommendation. However, they are limited by the fact that they rely heavily on manual adjustment of parameters due to their traditional manual approach to students' learning styles and the selection of educational resources.

The rise of graph neural networks, especially the success of models such as graph convolutional network GCN [17], graph attention network GAT, and graph representation learning GraphSage [18], has led to the rapid development of graph neural network recommendation [19, 20]. For example, Xu et al. [21] proposed an algorithm combining knowledge graph and collaborative filtering (FKGCF), which utilizes not only the user's evaluation information of the course but also the semantic information of the course itself for course recommendation. Jibing Gong et al. [22] proposed a heterogeneous perspective of attention graph convolutional network for MOOC-oriented knowledge concept recommendation by an adaptive attention mechanism that incorporates contextual information from different meta-paths to capture students' different interests and make effective recommendations. Although these models have been successful, they ignore the user as a learner and suffer from poor performance in the course recommendation problem. Therefore, this paper proposes a collaborative graph convolutional network and learning style course recommendation model to solve the above problems.

3 The Proposed Model

Currently, recommendation algorithms based on graph convolutional networks usually use historical user interaction data without considering the influence of the learning styles possessed by the learners themselves. To address those problems, this paper proposes an algorithm called collaborative graph convolutional networks and learning styles course recommendation model (CGCNLS). The algorithm propagates learner-course interaction information in a recursive form, applies higher-order interaction information in the embedding vector generation process. And integrate the learning styles similarity of the recommended objects into the course prediction rating of the graph convolutional network, compensating for the shortcomings of existing recommendation algorithms that ignore the learning styles of the learners themselves. The algorithm model is shown in Fig. 1.

3.1 Problem Definition: Course Recommendation Models

The goal of the course recommendation algorithm is to predict the learners' rating for the untaken course:

$$\hat{y}_{u,c} = F(D_u, D_c) \tag{1}$$

To be specifically, using a prediction function F to estimate the likelihood that a user u will favor a course c, given the data D_u and D_c, to describe the user u and course c.

3.2 Course Learning Styles Similarity Score

Learning Styles Profile. To clearly illustrate the course learning styles similarity score in this paper, we first introduce two definitions.

Definition 1 (learner profile): It is assumed that learner learning styles are represented by a real-valued vector LS_u from 0 to 1 in Eq. 1, where rea, tra, soc denote realistic, traditional, and social learning styles respectively. Equation 1 is as follows:

$$LS_u = (rea, tra, soc) \tag{2}$$

Examples of course-learning styles vectors are shown in Table 1, and these vectors were calculated using the information in Table 3 through the process shown in Fig. 2.

Definition 2 (course profile): Online courses cover many theoretical concepts. Different theoretical concepts belong to different areas of expertise, and different areas of

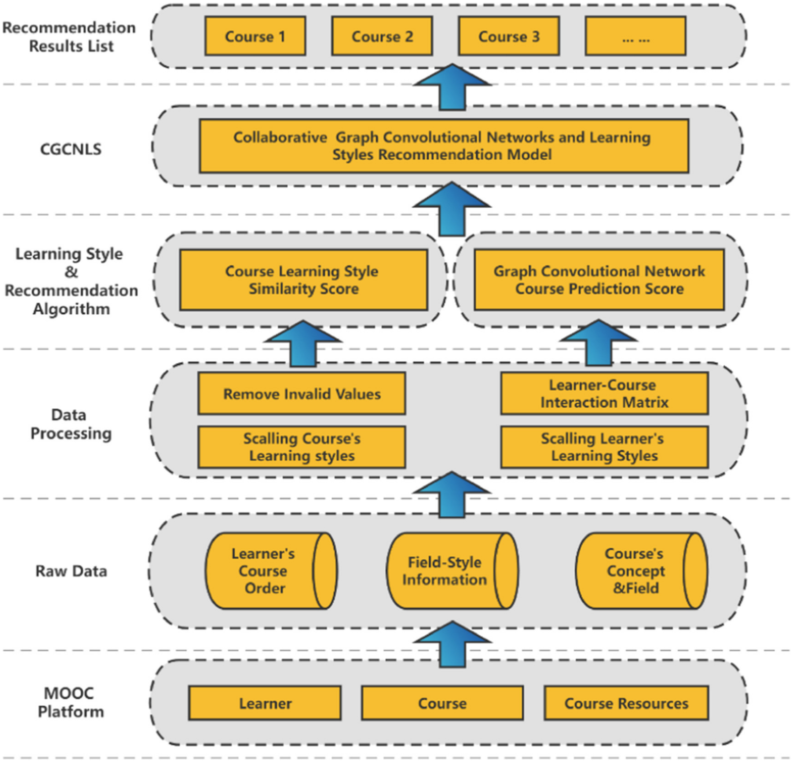


Fig. 1. CGCNLS model.

Table 1. Examples of learner learning styles vectors

Learner	rea	tra	soc
Learner 1	0.1029	0.7402	0.1569
Learner 2	0.8142	0.1858	0.0000
Learner 3	0.0000	0.2137	0.7863
…	…	…	…

expertise have their corresponding learning styles. Therefore, according to the correlation between course-concept-learning styles, the course profile can be represented by a learning styles vector, as in Eq. 2, which indicates the learning styles category that the course fits.

$$LS_c = (rea, tra, soc) \tag{3}$$

Examples of course-learning styles vectors are shown in Table 2, and these vectors were calculated using the information in Table 3 through the process shown in Fig. 2.

Table 2. Example of course learning styles vector

Course	rea	tra	soc
Course1	0.7532	0.1265	0.1203
Course2	0.2105	0.5789	0.2106
…	…	…	…

Calculating Learning Styles Vectors. Table 3 shows some examples of a field-learning styles matching measure based on an authentic questionnaire and guidance from educational professionals.

Table 3. Field-learning styles type matching metric representation example

Filed	Learning styles type
Computer Science and Technology	Realistic
Agronomy	Realistic
Mechanics	Traditional
Mathematics	Traditional
Psychology	Social
Pedagogy	Social
…	…

The learner learning styles vectors in Table 1 and the course learning styles vectors in Table 2 are calculated as shown in Fig. 2.

Calculating Learning Styles Similarity Score. Based on the learner learning styles vector and the course learning styles vector, the cosine similarity was used to calculate the learning styles similarity between the learner and the course, and obtain the course learning styles similarity score $Sim_{cls}(u, c)$, as shown in Eq. 3.

$$Sim_{cls}(u, c) = \frac{\sum_{i=1}^{n}\left(LS_{u_i} \times LS_{c_i}\right)}{\sqrt{\sum_{i=1}^{n}\left(LS_{u_i}\right)^2} \times \sqrt{\sum_{i=1}^{n}\left(LS_{c_i}\right)^2}} \tag{4}$$

where LS_{u_i} denotes the i-th value in the learner learning styles vector and LS_{c_i} denotes the i-th value in the course learning styles vector. The learning styles similarity score will be integrated into the subsequent collaborative prediction rating.

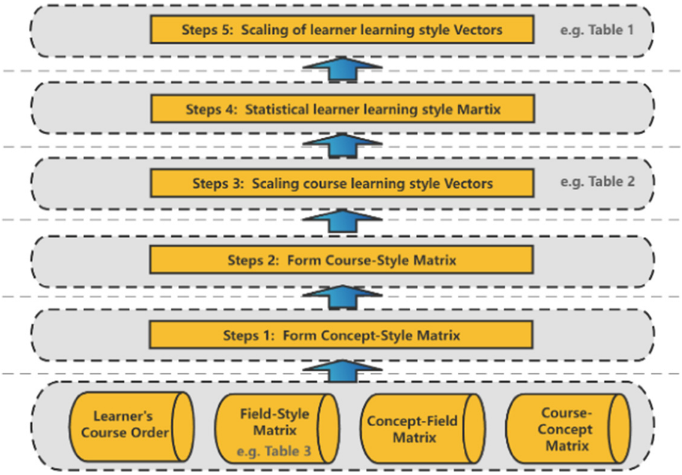


Fig. 2. Learning styles vector calculation process diagram

3.3 Graph Convolutional Network Course Prediction Rating

As with current recommendation models, we describe the learner u (course c) using the embedding vector $e_u \in \mathbb{R}^{dim}\left(e_c \in \mathbb{R}^{dim}\right)$, where *dim* denotes the size of the embedding dimension. This process is seen as constructing a matrix of parameters as an embedding look-up table:

$$E = [\ \underbrace{e_{u_1}, \cdots, e_{u_N}}_{\text{learners embeddings}}\ ,\ \underbrace{e_{c_1}, \cdots, e_{c_M}}_{\text{courses embeddings}}\ ] \tag{5}$$

The main idea of graph convolutional networks is to propagate learner-course interaction information over a learner-course interaction graph, encoding higher-order interaction information into the embedding vector generation process. The graph convolutional network course prediction rating model consists of three main components: higher-order embedding propagation layer, embedding aggregation layer and prediction rating layer.

Higher-Order Embedding Propagation Layer.
As shown in Fig. 3, based on the above idea, we can encode the embedding information from learner u's connected course c as learner u's first-order embedding information for enhancing learner u's own embedding information by the following form. Definition as follows.

$$e_u^{(1)} = \sum\nolimits_{c \in \mathcal{N}_u} \alpha_{uc} e_c \tag{6}$$

where $e_u^{(1)}$ denotes the first-order connectivity information for learner u, e_c denotes the initial embedding of course c, and α_{uc} denotes the decay coefficient for each propagation on edge (u, c). In this paper, we adopt the same idea as GCN[17] and set α_{uc} to $\frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_c|}}$, where $|\mathcal{N}_u|$ and $|\mathcal{N}_c|$ denote the number of first-hop neighbors of the learner u and the course c. α_{uc} not only takes on the function of equalizing how much the historical course c contributes to the preferences of the constituent learner u, but also in the process of embedding propagation different embedding information can be decayed as the path length changes.

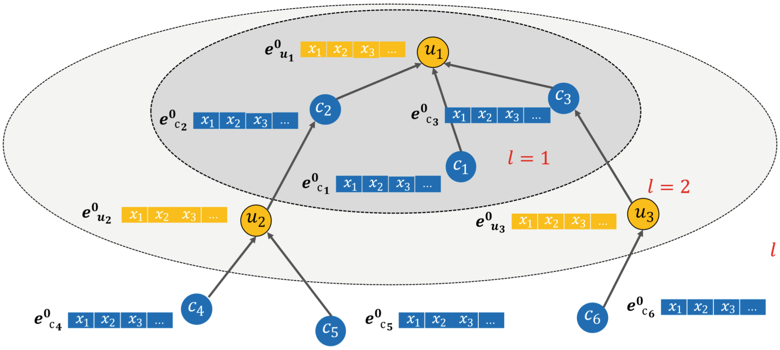


Fig. 3. An illustration of the learner-course interaction graph and the high-order interactive information

According to the form of propagation of first-order embedded information, the form of l-order embedded information for learner u in this method is as follows:

$$\mathrm{e}_u^{(l)} = \sum\nolimits_{c\in\mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_c|}} \mathrm{e}_c^{(l-1)} \tag{7}$$

Similarly, an expression for the l-order higher-order embedding information of course c can be obtained in the form:

$$\mathrm{e}_c^{(l)} = \sum\nolimits_{u\in\mathcal{N}_c} \frac{1}{\sqrt{|\mathcal{N}_c|}\sqrt{|\mathcal{N}_u|}} \mathrm{e}_u^{(l-1)} \tag{8}$$

Embedding Aggregation Layer.

As shown in Fig. 4. After performing the embedding propagation of the L-layer, multiple higher-order information $\left\{\mathrm{e}_u^{(0)}, \cdots, \mathrm{e}_u^{(L)}\right\}$ of the learner node u was obtained, and the same for the course node. The outputs of the different layers emphasise the connectivity information of different orders. Therefore, an aggregation mechanism is used to aggregate the embedding information of each order into a single vector, as shown in Eqs. 8 and 9:

$$e_u^* = \sum\nolimits_{l=0}^{L} p_l \mathrm{e}_u^{(l)} \tag{9}$$

$$e_c^* = \sum\nolimits_{l=0}^{K} p_l \mathrm{e}_c^{(l)} \tag{10}$$

where p_l is 1/(L + 1) to balance the effect of each layer of embedding propagation on the final embedding representation. The node embedding representation obtained

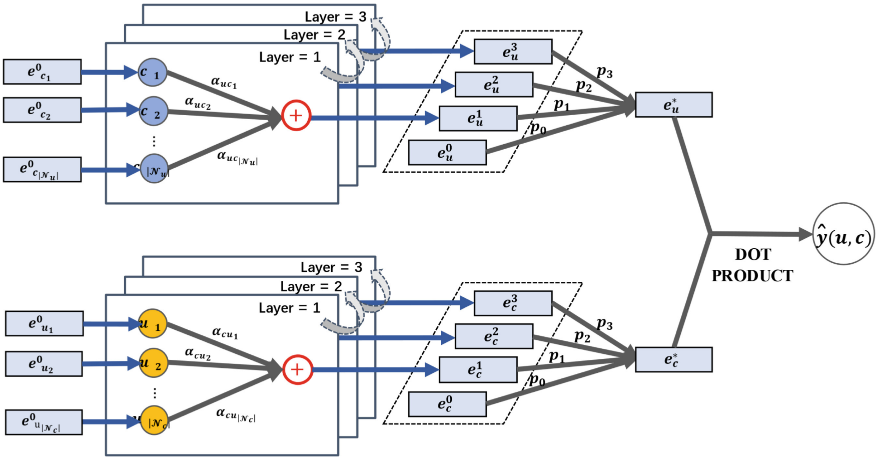


Fig. 4. An illustration of GCN model architecture

by aggregating the embedding information at each level in this way contains different semantic information in the graph structure, and the final embedding representation obtained will be more comprehensive.

Prediction Rating Layer. Finally, we perform inner product operations on the aggregated embedded representations of learner u and course c to predict the predicted rating of learner u for that course c:

$$\hat{y}(u,c) = \mathrm{e}_u^{*\top}\mathrm{e}_c^{*} \tag{11}$$

where $\hat{y}$ will be integrated into the subsequent collaborative prediction rating, which is used to generate the recommendation list.

Loss Functions and Trainers. The only parameter that needs to be trained in the whole graph collaborative filtering model is the embedding lookup table described earlier, i.e. the embedding representation of layer 0. To optimize the graph collaborative filtering model, the BPR loss function is chosen, which calculates the overall loss of the model by assuming that learners should have higher prediction values for learned courses than for unlearned courses.

$$L_{BPR} = -\sum\nolimits_{(u,i,j)\in O} \ln \sigma\left(\hat{y}_{ui} - \hat{y}_{uj}\right) + \lambda\|\Theta\|^2 \tag{12}$$

where $O = \left\{(u,i,j) \mid (u,i) \in \mathcal{R}^+, (u,j) \in \mathcal{R}^-\right\}$ denotes the training set, where $\mathcal{R}^+$ denotes the set of courses that learners have learned. $\mathcal{R}^-$ is the set of courses that learners have not learned through a random negative σ is the sigmoid function; λ is used to control the L2 regularization strength, and Θ denotes the parameters of the model as a whole, i.e. the layer 0 embedding representation. We use the mini-batch Adam optimizer to optimize and update the model parameters.

3.4 Collaborative Prediction Rating

The prediction rating $\hat{y}_{CGCNLS}(u, c)$ of the CGCNLS model is a fusion of the graph convolutional network course prediction rating $\hat{y}(u, c)$ and the learning styles similarity score $Sim_{cls}(u, c)$.

For each learner u, after obtaining a list of predicted ratings of the course by learner u through the graph convolutional network recommendation algorithm $\{\hat{y}(u, c_1), \hat{y}(u, c_2), \ldots, \hat{y}(u, c_{|C|})\}$, the predicted ratings $\hat{y}(u, c)$ obtained from the graph convolutional network model are optimized by calculating the learning styles similarity between learner u and course c. The optimization equation is as follows.

$$\hat{y}_{CGCNLS}(u, c) = \hat{y}(u, c) + \varepsilon \times Sim_{cls}(u, c) \tag{13}$$

where ε is the predictive scoring collaborative weights. Based on the collaborative predicted rating, the Top-K courses of the collaborative predicted rating are selected as the final course recommendation results $R(u)\{c_1, c_2, \ldots, c_{Top-K}\}$.

4 Experiment Process

To evaluate the performance of our proposed collaborative graph convolutional network and learning style course recommendation model, we conducted experiments to answer the following research questions.

RQ1: How does our proposed CGCNLS course recommendation model perform compared to some existing baseline approaches?
RQ2: What is the impact of different co-weighting factors ε on the performance of the model?
RQ3: What is the performance of our proposed model when changing hyperparameters?

4.1 Experimental Environment and Data Set Processing

This paper conducted experiments on the real-world dataset MOOCCube [23] collected by XuetangX. In this paper, users with 10–20 course subscriptions were selected, and a total of 5738 users and 649 courses were used for the experiment. Detailed statistical information on these data is shown in Table 4. Each positive instance is paired with a randomly sampled negative instance during the training process. During testing, each historical course in the test set is considered a target course, and the corresponding course for the same user in the training set is considered a historical course. Finally, all courses of the user except the training set are rated, and the Top-K course recommendation list is obtained by sorting according to the predicted ratings.

Table 4. Statistical table of data sets

Dataset	#Courses	#Users	#Interactions
MOOCCube	649	5738	72148

To more accurately measure the performance of the algorithm, this paper uses five-fold cross-validation. That is, the experimental data are randomly grouped into 5 parts, one of which is used as the test set and the other 4 parts are used as the training set. A total of five tests were conducted and the average result of the five tests was used as the final evaluation result of the algorithm. All experiment results are obtained on a machine with Python 3.8, 3.80 GHz CPU, 12 GB of Video Memory and 32 GB of RAM。

4.2 Evaluation Metrics

Several metrics widely used in recommender system evaluation are used to measure the performance of our proposed model in different aspects, including the precision rate of Top-K items, the recall rate of Top-K items, and the normalized discounted cumulative return of Top-K items. In our experiments, we set the Top-K to 5, 10, 20, and 30.

Let R(u) represent the list of recommendations calculated by the model for the user based on the user's behavior on the training set, and T(u) represent the list of target courses for the user on the test set.

Precision@K is the calculation of how many courses in the predicted recommendation list are actually of interest to the user. The definition is as follows.

$$\text{Precision} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \tag{14}$$

Recall@K is a calculation of how many of the courses in a user's true favorite list are predicted by the recommendation algorithm. The definition is as follows.

$$\text{Recall} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \tag{15}$$

where R(u) represents the list of recommendations calculated by the model for the user based on the user's behavior on the training set, and T(u) represents the list of target courses for the user on the test set. Another ranking metric we use in our evaluation is the Normalized Discounted Cumulative Gain NDCG@K, which measures the performance of the retrieval system based on the hierarchical relevance of the retrieved entities and is a precision-based metric. The definition is as follows:

$$DCG_u@K = \sum\nolimits_{k=1}^{K} \frac{2^{rel_u^i} - 1}{log_2(i+1)} \tag{16}$$

$$IDCG_u@K = \sum\nolimits_{k=1}^{K} \frac{1}{log_2(i+1)} \tag{17}$$

$$NDCG@K = \frac{1}{|U|}\sum\nolimits_{u=1}^{U}\frac{DCG_u@K}{IDCG_u@K} \tag{18}$$

where $DCG_u@K$ indicates the discounted cumulative gain of the Top-K recommendation list for the user u, rel_u^i indicates the relevance (0 or 1) of the i-th recommendation result to user u, and $IDCG_u@K$ indicates the maximum discounted cumulative gain under ideal conditions.

4.3 Comparison with Baseline Method (RQ1)

To validate the performance of our proposed method CGCNLS, we will compare it with the following baseline approach.

GMF [24] decomposes the scoring matrix R into a user matrix U and an item matrix I. The product of U and I get closer to the true scoring matrix in a continuous iterative training.

NeuMF [24] combines GMF and MLP to operate on the embedding of target users and candidate courses to find the predicted ratings of users on candidate courses.

FISM [25] is an item-based collaborative filtering method for recommendations based on the average embedding of all historical courses of the user and the embedding of the target course.

NAIS [26] is also an item-based collaborative filtering method but distinguishes the weights of different historical courses through an attention mechanism.

NGCF [19] is the current baseline algorithm for recommendations based on graph neural networks.

Table 5 shows the experimental results of comparing our proposed model with some baseline methods on offline datasets. The experimental comparison shows that our proposed CGCNLS outperforms other baselines in all evaluation metrics, indicating the effectiveness of CGCNLS in MOOC course recommendations. Compared with the traditional neural network model, our graph convolutional network can better learn to model the complex interactions between learners and courses, thus improving accuracy, recall, and normalized discount cumulative gain. Compared with the graph neural network baseline model, our proposed collaborative learning style strategy is somewhat advanced and further enhances the overall interpretability of the model while ensuring its performance.

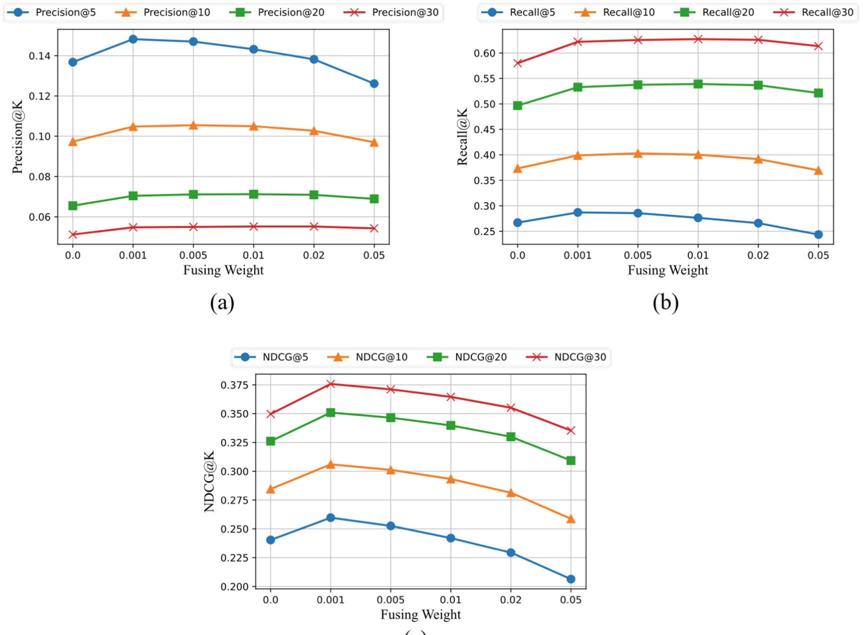
Table 5. Results obtained with different models on the MOOCCube dataset

Model	Precision@Top-K				Recall@Top-K				NDCG@Top-K			
	K = 5	K = 10	K = 20	K = 30	K = 5	K = 10	K = 20	K = 30	K = 5	K = 10	K = 20	K = 30
GMF	0.1221	0.0896	0.0640	0.0505	0.2372	0.3411	0.4862	0.5691	0.2119	0.2551	0.3048	0.3291
NueMF	0.1269	0.0898	0.0631	0.0499	0.2470	0.3446	0.4766	0.5653	0.2204	0.2542	0.3043	0.3285
FISM	0.1272	0.0878	0.0628	0.0496	0.2443	0.3335	0.4749	0.5599	0.2173	0.2504	0.3019	0.3249
NAIS	0.1324	0.9561	0.0647	0.0496	0.2554	0.3651	0.4903	0.5603	0.2254	0.2721	0.3119	0.3241
NGCF	0.1328	0.0955	0.0665	0.0525	0.2588	0.3668	0.5054	0.5952	0.2312	0.2757	0.3221	0.3476
CGCNLS (ours)	**0.1470***	**0.1055***	**0.0711***	**0.0550***	**0.2856***	**0.4029***	**0.5375***	**0.6254***	**0.2526***	**0.3012***	**0.3465***	**0.3711***

We use **bold** to mark the best performance and underline to indicate the best performance other than CGCNLS.

4.4 Experiments of Different Collaborative Weight Factor ε (RQ2)

The predictive scoring co-weight ε controls the proportion of learning style similarity score in the final collaborative predictive rating and is a key factor in the proposed algorithm. We vary the predictive scoring co-weights ε in the set of {0, 0.001, 0.005, 0.01, 0.02, 0.05} for the experiments. In addition, layer number l is set to 3, embedding dimension d is set to 64, and compared under different list lengths Top-K. The final experimental results are shown in Fig. 5. We can see that different weights ε settings affect the model differently. When the weight ε is less than 0.02, the evaluation indexes of the proposed algorithm are better than our proposed single graph convolutional network model. When the weight ε is 0.005, the accuracy and recall of the proposed method reach the best value. The results prove that The cooperative graph convolutional network and learning style approach can make our model more reliable and accurate.



(a) (b)

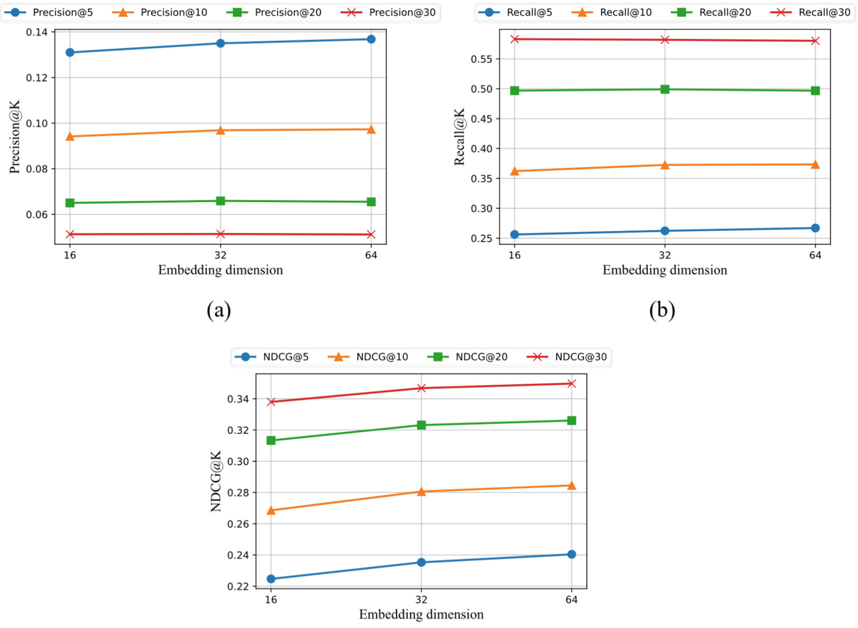
(c)

Fig. 5. Comparison of results for different prediction score fusion weights ε

4.5 Parameter Settings (RQ3)

Here, we investigate the sensitivity of different parameters and report the results of CGCNLS under each of them. As shown in Fig. 6 and Fig. 7.

Experiments of Different GCN Embedding Dimension *d*. When training a predictive scoring model for graph convolutional network, using different embedding dimension d produces different results. We vary the embedding dimension d in the set of {16, 32, 64}, in addition, the layer number l is set to 3, and the prediction scoring co-weight ε is set to 0.0, and conduct experiments at different recommendation list lengths Top-K. The final experimental results are shown in Fig. 6. We can see that the model has better evaluation indexes when d is 64, the overall performance of the model is stable when changing the embedding dimension, and the model performance gradually increases with dimensionality.

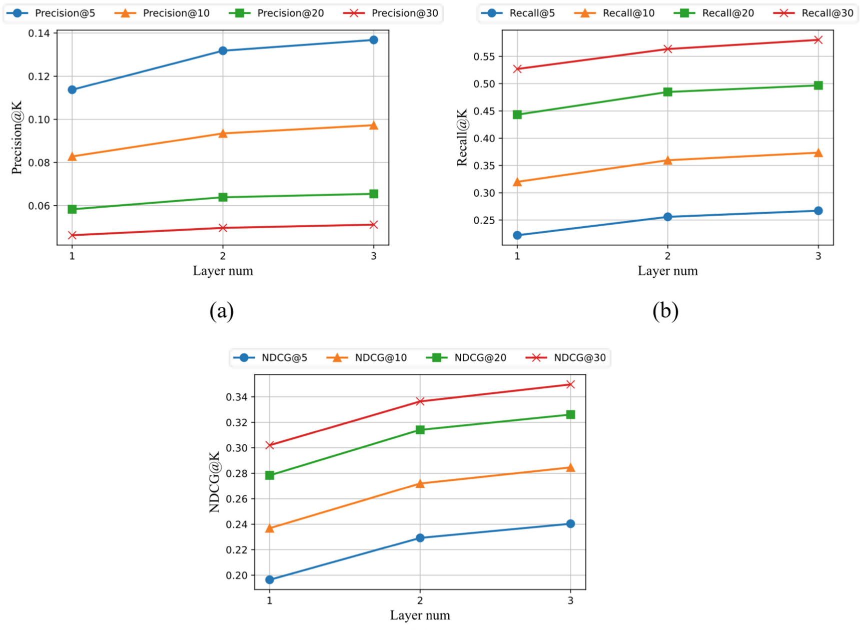


(a) (b)

(c)

Fig. 6. Comparison of the results of the different embedding dimensions d

Experiments of Different GCN Layers *l*. Meanwhile, the higher-order information contained in the final embedding representation of the learner (course) obtained for different Graph convolutional network layers (GCNL) is different. We vary the layer number l in the set of {1,2,3}. In addition, the embedding dimension d is set to 64, the prediction scoring co-weight ε is set to 0.0, and the experiments are conducted at different recommendation list lengths Top-K. The final experimental results are shown in Fig. 7. We can see that different layers of the graph convolutional network have a large impact on the model performance, and the model performs best when $l = 3$.



(a) (b)

(c)

Fig. 7. Comparison of the results of the different number of GCN layers l

5 Conclusion

In this paper we propose a collaborative graph convolutional network and learning style course recommendation model (CGCNLS), which uses a graph convolutional network to learn embedding information that can effectively represent the relevance differences between learners and courses. We also introduce a collaborative weighted to synergize course prediction score and learning styles similarity score. The accuracy of the recommendations is improved while considering the learning styles of the learners. We conducted extensive comparative and ablation experiments on the public dataset, and the experimental results show that the performance of CGCNLS is advanced compared to the baseline methods.

References

1. Khanal, S.S., Prasad, P.W.C., Alsadoon, A., Maag, A.: A systematic review: machine learning based recommendation systems for e-learning. Educ. Inf. Technol. **25**(4), 2635–2664 (2019). https://doi.org/10.1007/s10639-019-10063-9
2. Guruge, D.B., Kadel, R., Halder, S.J.: The state of the art in methodologies of course recommender systems—a review of recent research. Data **6**(2), 18 (2021)
3. Khalid, A., Lundqvist, K., Yates, A.: A literature review of implemented recommendation techniques used in massive open online courses. Expert Syst. Appl. **187**, 115926 (2022)
4. Qiu, F., Zhu, L., Zhang, G., et al.: E-learning performance prediction: mining the feature space of effective learning behavior. Entropy **24**(5), 722 (2022)

5. Wu, L., He, X., Wang, X., et al.: A survey on accuracy-oriented neural recommendation: from collaborative filtering to information-rich recommendation. IEEE Transactions on Knowledge and Data Engineering (2022)
6. Wu, L., He, X., Wang, X., et al.: A survey on neural recommendation: from collaborative filtering to content and context enriched recommendation (2021). arXiv preprint arXiv:2104.13030
7. Gao, C., Wang, X., He, X., et al.: Graph neural networks for recommender system. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, pp. 1623–1625 (2022)
8. Piao, G.: Recommending knowledge concepts on MOOC platforms with meta-path-based representation learning. In: Proceedings of The 14th International Conference on Educational Data Mining (EDM21), pp. 487–494 (2021)
9. Sheng, D., Yuan, J., Xie, Q., et al.: ACMF: an attention collaborative extended matrix factorization based model for MOOC course service via a heterogeneous view. Futur. Gener. Comput. Syst. **126**, 211–224 (2022)
10. Truong, H.M.: Integrating learning styles and adaptive e-learning system: current developments, problems and opportunities. Comput. Hum. Behav. **55**, 1185–1193 (2016)
11. Gope, J., Jain, S.K.: A learning styles based recommender system prototype for edX courses. In: 2017 International Conference on Smart Technologies for Smart Nation (SmartTechCon), pp. 414–419. IEEE (2017)
12. Laksitowening, K.A., Yanuarifiani, A.P., Wibowo, Y.F.A.: Enhancing e-learning system to support learning style based personalization. In: 2016 2nd International Conference on Science in Information Technology (ICSITech), pp. 329–333. IEEE (2016)
13. Hajri, H., Bourda, Y., Popineau, F.: Personalized recommendation of open educational resources in MOOCs. In: McLaren, B.M., Reilly, R., Zvacek, S., Uhomoibhi, J. (eds.) CSEDU 2018. CCIS, vol. 1022, pp. 166–190. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21151-6_9
14. Sanjabi, T., Montazer, G.A.: Personalization of E-learning environment using the kolb's learning style model. In: 2020 6th International Conference on Web Research (ICWR), pp. 89–92. IEEE (2020)
15. Yan, L., Yin, C., Chen, H., Rong, W., Xiong, Z., David, B.: Learning resource recommendation in e-learning systems based on online learning style. In: Qiu, H., Zhang, C., Fei, Z., Qiu, M., Kung, S.-Y. (eds.) KSEM 2021. LNCS (LNAI), vol. 12817, pp. 373–385. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-82153-1_31
16. Sensuse, D.I., Hasani, L.M., Bagustari, B.: Personalization strategies based on Felder-Silverman learning styles and its impact on learning: a literature review. In: 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE), pp. 293–298. IEEE (2020)
17. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks (2016). arXiv preprint arXiv:1609.02907
18. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 1025–1035 (2017)
19. Wang, X., He, X., Wang, M., et al.: Neural graph collaborative filtering. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 165–174 (2019)
20. Ying, R., He, R., Chen, K., et al.: Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 974–983 (2018)

21. Xu, G., Jia, G., Shi, L., et al.: Personalized course recommendation system fusing with knowledge graph and collaborative filtering. Computational Intelligence and Neuroscience 2021 (2021)
22. Gong, J., Wang, S., Wang, J., et al.: Attentional graph convolutional networks for knowledge concept recommendation in MOOCs in a heterogeneous view. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 79–88 (2020)
23. Yu, J., Luo, G., Xiao, T., et al.: MOOCCube: a large-scale data repository for NLP applications in MOOCs. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 3135–3142 (2020)
24. He, X., Liao, L., Zhang, H., et al.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182 (2017)
25. Kabbur, S., Ning, X., Karypis, G.: Fism: factored item similarity models for top-n recommender systems. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 659–667 (2013)
26. He, X., He, Z., Song, J., et al.: Nais: neural attentive item similarity model for recommendation. IEEE Trans. Knowl. Data Eng. **30**(12), 2354–2366 (2018)