Learning Behavior-Aware Cognitive Diagnosis for Online Education Systems

Yiming Mao[1(✉)], Bin Xu[1,2,3], Jifan Yu[2], Yifan Fang[2], Jie Yuan[2], Juanzi Li[2,3], and Lei Hou[2,3]

[1] Global Innovation Exchange (GIX) Institute, Tsinghua University, Beijing 100084, China
mym19@mails.tsinghua.edu.cn

[2] Department of Computer Science and Technology, BNRist, Tsinghua University, Beijing 100084, China

[3] KIRC, Institute for Artificial Intelligence, Tsinghua University, Beijing 100084, China

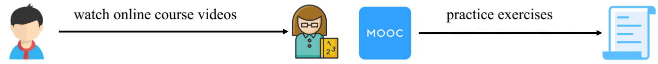
Abstract. Cognitive diagnosis, which aims to diagnose students' knowledge proficiency, is crucial for numerous online education applications, such as personalized exercise recommendation. Existing methods in this area mainly exploit students' exercising records, which ignores students' full learning process in online education systems. Besides, the latent relation of exercises with course structure and texts is still underexplored. In this paper, a learning behavior-aware cognitive diagnosis (LCD) framework is proposed for students' cognitive modeling with both learning behavior records and exercising records. The concept of LCD was first introduced to characterize students' knowledge proficiency more completely. Second, a course graph was designed to explore rich information existed in course texts and structures. Third, an interaction function was put forward to explore complex relationships between students, exercises and videos. Extensive experiments on a real-world dataset prove that LCD predicts student performance more effectively, the output of LCD is also interpretable.

Keywords: Cognitive diagnosis · Intelligent education · Graph neural network

1 Introduction

Online education systems, such as Massive Open Online Course (MOOC), Intelligent Tutoring System (ITS), and Mobile Autonomous School (MAS), can provide a series of computer-aided applications for better tutoring, such as computer adaptive test [12] and knowledge tracing [16,23]. Among these applications, one of the critical applications is to diagnose states of student knowledge, which is called **cognitive diagnosis**. In intelligent education systems, cognitive diagnosis aims to discover students' states from history learning records of students, such as their proficiencies on specific knowledge concepts.

J. Zeng et al. (Eds.): ICPCSEE 2021, CCIS 1452, pp. 385–398, 2021.
https://doi.org/10.1007/978-981-16-5943-0_31



Online courses
V1:Data Struture: Red-black Tree
V2:Operation System:Memory Management
V3:Operation System:File System
V4:Algorithm:Quick Sort

Exercises	Knowledge Concepts	Records
E1:How to use the heap to sort	Heap	✔
E2:Operating system belongs to	Operation System	✖
E3:Red and black trees are equivalent to__ Rank B tree	Red–black Tree、 B Tree	✔
E4:The results of 3 & 5 are__	Programming	✔

Fig. 1. An example of student learning records in online education system.

Massive efforts have been made in the cognitive diagnosis area. Among them, item response theory (IRT) [17], as a typical unidimensional model, considered each student as a single proficiency variable (i.e., latent trait). Comparatively, multidimensional models, such as Deterministic Inputs, Noisy-And gate (DINA) [2] model, characterized each student by a binary latent vector, which described whether or not he had mastered the knowledge concepts with the help of Q-matrix(exercise knowledge concept matrix labeled by educational experts). Furthermore, Neural Cognitive Diagnosis [20] uses artificial neural network to model complex non-linear interactions.

The methods above have been proved effective but still have limitations. A deficiency is that they only consider the interaction between students and exercises and do not explore complete students' learning behaviors in online education systems. Figure 1 shows an example of such a complete learning process of a typical student in Massive Open Online Courses (MOOCs). The student first watches four videos of three courses for his goals and does four exercises as practice. Specifically, he reads the exercise contents, answers them by applying the corresponding knowledge learned from videos, and leaves response records(e.g., right or wrong). The student only answers E_2 incorrectly, which demonstrates that he has a high mastery on knowledge concepts like "Heap", "B Tree" and is weak in "Operation System". We can observe that when students learn in MOOCs, they watch videos first to learn the knowledge they want to master, which helps them practice exercises related to videos.

Moreover, existing methods model each exercise as an independent data instance and do not consider latent relations between exercises. For an online course, it is consists of structured videos and exercises. Besides, exercises are also connected by knowledge relations such as prerequisite [14].

To address these challenges, we propose a learning behavior-aware cognitive diagnosis(LCD) framework to predict student performance by taking advantage of full student learning records and the latent relations of exercises. Specifically, we first build a graph consisting of videos, exercises and construct the structure by utilizing course structures and knowledge concepts. Next, we encode students, exercises, videos as vectors and leverage a multi-layer perceptron (MLP) [6] for modeling the complex interactions of student answering exercises. A graph convolutional network(GCN) [9] is used to automatically refine the representation as feature vectors of exercises and videos. Finally, we conduct extensive experiments on a real-world dataset, which is the first dataset to contain video information and exercise text materials. The results show that our approach outperforms other methods. Detailed parameter analysis and discussion are also discussed. In summary, the contributions are as follows:

- We develop a new method named LCD to model the complex interactions of full student learning behaviors in an end-to-end manner.
- We propose two methods to construct a course graph to build relationships between videos and exercises under the graph neural network framework.
- We construct a novel dataset including students watching video records and exercising records with texts and structure information in MOOC.

2 Related Work

Cognitive Diagnosis. Generally, traditional cognitive diagnostic models (CDM) could be roughly divided into discrete models and continuous models. The fundamental discrete CDM is Deterministic Inputs, Noisy-And gate (DINA) [2], represented each student as a binary vector which denoted whether he mastered or not the knowledge concepts required by exercises with a given Q-matrix prior. The DINA-based models are applied to further specific educational scenarios, such as differential item functioning assessment [7], learning team formation [11] and comprehension test validation. Comparatively, the basic continuous method is item response theory (IRT) models [4], which characterize students by a continuous variable, i.e., latent trait, and use a logistic function to model the probability that a student correctly solves a problem. Many variations of CDMs were proposed by combining learning information. For example, learning factors analysis (LFA) [1] incorporated the time factor into the modeling. In recent years, Wu [10] put forward a fuzzy cognitive diagnosis framework to diagnose students' knowledge state in the examination, which considers both subjective and objective questions. Thai [19] leveraged a multi-relational matrix factorization to map students' knowledge proficiency as a hidden vector. Another typical method is the deep learning method. Piech [16] developed a deep knowledge tracing method based on a recurrent neural network to simulate students' learning process and improve the effectiveness of the task of predicting students' achievement. Huang [8] extended this method, and they used text information for deeper semantic mining of exercises. Wang et al. [20] proposed NeuralCDM that uses artificial neural networks to capture relations between students and exercises.

Graph Neural Networks. The GNN [18] is a type of neural network that can operate on graph-structured data. The graph is a type of data structure representing objects and their relationships as nodes and edges. In recent years, various generalization frameworks and significant operations have been developed with successful results in various applications, such as recommender systems [21] and social network analysis [15]. Very few studies have been done in the field of online learning and education. A recent study on college education by Hiromi proposed a Graph-based Knowledge Tracing (GKT) [13] combines the knowledge tracing with a graph neural network. It encodes a learner's hidden knowledge state as embeddings of graph nodes and updates the state in a knowledge graph. However, GKT aims to predict students' scores and does not distinguish between an exercise and the knowledge concepts it contains. Thus it is unsuitable for cognitive diagnosis.

3 Problem Definition

In this section, we give some basic definitions and formulate the problem of learning behavior-aware cognitive diagnosis in the MOOC platform.

Knowledge component is denoted as $K = \{Kn, Kr\}$, where $Kn = \{k_i\}_{i=1}^{|Kn|}$ is the set of knowledge concepts, which are the subjects taught in courses (e.g., "Red-black Tree" in "Data Structure"). $Kr = [(k_i, k_j)]$ is the relation of knowledge concepts. Here we consider the prerequisite relation, which is crucial for students to learn, organize, and apply knowledge [14]. Specifically, if a pair $(k_i, k_j) \in Kr$, k_i is a prerequisite concept of k_j.

Course corpus is composed by numerous courses, denoted as $M = \{C_i\}_{i=1}^{M}$, where C_i is one course. Each course includes a sequence of videos and exercises, i.e., $C_i = [(c_{ij}, kc_{ij}, t_{ij})]_{j=1}^{|C_i|}$, where the triplet $(c_{ij}, kc_{ij}, t_{ij})$ describes the j-th component of course C_i. t_{ij} refers the type (video or exercise), kc_{ij} (usually labeled by experts) is the set of knowledge concepts the component relates to and c_{ij} indicates its contents (video subtitles or exercise texts). W is the set including all exercises and videos in M and the Q-matrix $Q = \{Q_{ij}\}_{|W| \times |Kn|}$, where $Q_{ij} = 1$ if W_i relates to knowledge concept k_j and $Q_{ij} = 0$ otherwise.

Course graph consists of only one type of node. Each node corresponds to a video or an exercise in W. Formally we represent it as (A, F), where $A \in \{0,1\}^{|W|*|W|}$ is an adjacency matrix and $F \in \mathbb{R}^{|W|*t}$ is the node feature matrix assuming each node has t features.

Student records consist of two types of behaviors: watching videos and doing exercises. Suppose we have a student set S, where students study individually. Formally, we record the learning behaviors of students as $R = R_1 \cup ... \cup R_{|S|}$, where $R_i = [(w_t, r_t)], w_t \in W$ denotes student S_i learns w_t at step t and r_t denotes the response. For exercises, r_t is the corresponding score the student got on w_i and r_t does not exist otherwise.

Learning behavior-aware cognitive diagnosis is defined as bellows: Given the course corpus M, knowledge component K, student set S, and students' response logs R, the goal of our cognitive diagnosis task is to build a model to mine students' proficiency on each knowledge concept. Since there is no ground truth for diagnosis results, following previous works, we adopt the **student performance prediction task** to validate cognitive diagnosis results' effectiveness.

4 Method

The entire structure of the LCD is shown in Fig. 3. Generally, for a cognitive diagnosis system, three elements need to be considered: student features, exercise features, and the interaction function including them [3]. In this paper, we add video features that can be obtained by course graph representation to enhance interaction function. Specifically, we construct a course graph by course structures, knowledge relations, and text features. Then we learn the representation of nodes by a graph convolutional network (GCN). The results are used as one of the exercise features and video features. We use one-hot vectors of the corresponding student, exercise, and video as input and obtain these three diagnostic features for each response log. The interaction function aggregates different features and outputs predicting score that the student can get on the exercise after training. Finally, we get students' proficiency vectors as diagnosis reports. It is worth mentioning that the GCN and the interaction function are trained in an end-to-end fashion.

4.1 Course Graph Construction

In this section, we introduce details to construct the course graph, which includes the adjacency matrix A and node features F. As shown in Fig. 2, First BERT is used to encode all texts of exercises and videos as the value of F. It is worth mentioning that we only use BERT to extract features of texts and do not fine-tune. Next, we propose two approaches named the structure-based method and the knowledge-based method to assign A. Detailed methods are as follows:

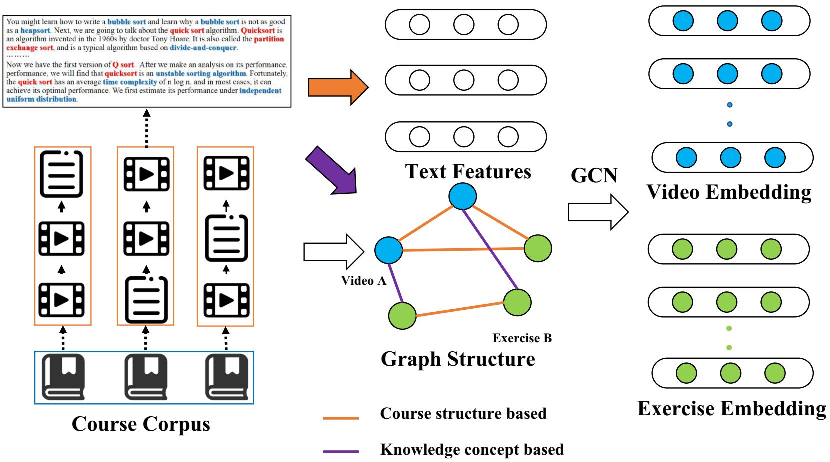


Fig. 2. The framework of course graph learning.

Structure-based Graph is a graph generated by the structured information of courses provided by online education system. We assign the adjacency A^S as:

$$A_{ij}^S = \begin{cases} \frac{MS(w_i,w_j)}{|MC(w_i)-MC(w_j)|}, & |MC(w_i) - MC(w_j)| < \lambda \\ 0, & otherwise \end{cases} \tag{1}$$

where $MS(w_i, w_j) = 1$ if w_i and w_j belong to a same course in M, $MS(w_i, w_j) = 0$ otherwise. $MC(w_i) = n$ indicates w_i is the n-th component of a course in M. That is A_{ij}^S depends on their distance w_i and w_j in the sequence of courses if both them belong to a same course. λ is a parameter to control the threshold.

Knowledge-based Graph is a densely connected graph generated by the knowledge concepts and relations, here we design the weight A^K as:

$$A_{ij}^K = \begin{cases} 1, & kc_i \cap kc_j \neq \emptyset \\ \alpha, & \{(k_x, k_y)|k_x \in kc_i, k_y \in kc_j\} \cap Kr \neq \emptyset \\ 0, & otherwise \end{cases} \tag{2}$$

Where kc_i indicates the set of knowledge concepts of w_i. That is A_{ij}^K is 1 if w_i and w_j contain at least one same knowledge concept; else A_{ij}^K is α if one of knowledge concepts w_i relates to is a prerequisite concept of one which w_j contains; otherwise it is 0. α is a parameter we manually set to control the importance of prerequisite relation.

Finally we assign A as: $A = Normalization(A^S + A^K)$.

4.2 Graph Representation Learning

In this work, we build our model on the graph above to learn useful representations of videos and exercises for the cognitive diagnosis task. Graph neural networks and its variant have achieved promising performance in various challenging tasks. We first briefly review its mechanism and then reveal the revision for this work in detail.

Traditional GNNs architectures can be considered as following general "message-passing" architectures.

$$H^k = M(A, H^{k-1}; \theta^k) \tag{3}$$

Here M is a message propagation function that defines how to aggregate and update node embeddings based on a trainable parameter θ^k. Furthermore, $H \in \mathbb{R}^{n*d}$ is the node embeddings computed after k steps of the message propagation. Here, the H^0 is initialized using node features F in the graph.

A popular variant of GNN is called graph convolutional neural network (GCN), which has demonstrated to be efficient to gather information in a graph through convolution operations. Its message propagation function is as following:

$$H^k = M\left(A, H^{k-1}; \theta^k\right) = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{k-1}W^{k-1}\right) \tag{4}$$

where $\sigma(\cdot)$ denotes a non-linear activation function such as $ReLU$ function. $\tilde{A} = A + I$ is an adjacency matrix with added self-connections. $\tilde{D}$ is a diagonal degree matrix of $\tilde{A}$ and $\tilde{A}^{k-1}$ is a trainable weight matrix of $k-1$ layer. We divide the set of node embedding H^k into two parts: E^k and V^k, which refers the exercise embedding and video embedding.

4.3 Modeling Process

The goal of the LCD modeling process is to combine different features to predict the probability of a typical student correctly answering a typical exercise. Figure 3 illustrates the structure of the LCD. Details are introduced as below.

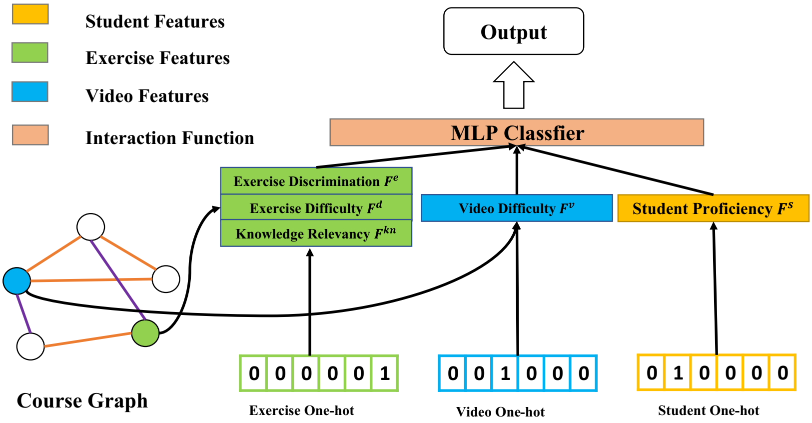


Fig. 3. The architectures of LCD.

Student Features reflect the studying abilities of students in knowledge concepts, which is defined in traditional cognitive diagnosis like DINA [2], IRT [4] and MIRT [19]. One of most important student features is the proficiency on knowledge concepts, which is the goal of cognitive diagnosis task. To better represent knowledge proficiency feature, we design it as explainable vectors similar to DINA and NeuralCDM [20] to guide students' self-assessment rather than the latent trait vectors in IRT and MIRT. Specifically, We use a continuous vector F^s to represent the knowledge proficiency of students, which the size of is the same as the number of knowledge concepts. The formulation of F^s is as below:

$$F^s = \text{sigmoid}(x^s \times B) \tag{5}$$

in which $F^s \in (0,1)^{1\times K}$, $x^s \in \{0,1\}^{1\times N}$ is the one-hot embedding to represent students, $B \in \mathbb{R}^{N\times K}$ is a trainable matrix. sigmoid is the sigmoid function which can control the value of F^s in 0 to 1. Each entry of F^s represents the knowledge level on a specific knowledge concept. For example, $F^s = [0.1, 0.8]$ indicates a low mastery on the first knowledge concept but high mastery on the second.

Exercise Features characterize the trait of exercises. To indicate the relationship between exercises and knowledge concepts, we design the knowledge relevancy vector and denote it as F^{kn}. F^{kn} has the same length of F^s, with each entry represents the relevance of every knowledge concept. In LCD, we use the pre-given Q-matrix Q (exercise knowledge concept matrix labeled by educational experts) to assign F^{kn} for each exercise as:

$$F^{kn} = x^e \times Q \tag{6}$$

where $F^{kn} \in \{0,1\}^{1\times Kn}, x^e \in \{0,1\}^{1\times E^k}$ is the one-hot representation of the exercise. In addition, we design a feature called the exercise difficulty vector F^d to represent the difficulty of each knowledge concept like F^s. F^d directly comes from the exercise embedding E^k. Furthermore, we design the exercise discrimination vector F^e, which represent the ability of distinguishing students. The higher the value of F^e is, the stronger the discrimination ability F^e has. F^e is obtained by a trainable matrix D:

$$F^d = \text{sigmoid}(x^e \times E^k) \tag{7}$$

$$F^e = \text{sigmoid}(x^e \times D), D \in \mathbb{R}^{M\times\beta} \tag{8}$$

In which β is a parameter to control the size of F^e.

Video Features denote the features that characterize the traits of videos. Since a video relates to more knowledge concepts than an exercise, which may get redundant information and reduce performance. We do not consider the knowledge relevancy of videos. Instead, we design a vector F^v to characterize video difficulty, indicating the difficulty for students to understand. Like F^d, F^v is direct from the video embedding V^k:

$$F^v = \text{sigmoid}(x^v \times V^k) \tag{9}$$

where $x^v \in \{0,1\}^{1\times V^K}$ is the one-hot representation of the video which satisfies the following conditions: 1) the knowledge concepts relating to x^e also relate to x^v. 2) Student x^s watched x^v before exercising x^e. It is worth mentioning that if x^v does not exist, we use a zero vector to represent F^v.

Interaction Function. We use Multi-Layer Perceptron (MLP) to obtain the interaction function. The formulations are as below:

$$y = \tau_n\left(\ldots\tau_1\left(F^s, F^{kn}, F^d, F^e, F^v, \lambda_f\right)\right) \tag{10}$$

where τ_n denotes the mapping function of the ith MLP layer and λ_f denotes model parameters of all the interactive layers.

$$\boldsymbol{x} = (F^{kn}\cdot F^s) \oplus F^e \oplus F^d \oplus F^v \tag{11}$$

$$\begin{aligned}\boldsymbol{f}_1 &= \text{sigmoid}\left(\mathbf{W}_1 \times \boldsymbol{x}^T + \boldsymbol{b}_1\right)\\ \boldsymbol{f}_2 &= \text{sigmoid}\left(\mathbf{W}_2 \times \boldsymbol{f}_1 + \boldsymbol{b}_2\right)\\ \boldsymbol{y} &= \text{sigmoid}\left(\mathbf{W}_3 \times \boldsymbol{f}_2 + \boldsymbol{b_3}\right)\end{aligned} \tag{12}$$

where $\oplus$ is the concatenation operation. The loss function of LCD is cross entropy between output y and true label r:

$$loss = -\sum_i \left(r_i \log y_i + (1 - r_i) \log \left(1 - y_i\right)\right) \tag{13}$$

After training, the value of F^s is what we get as the diagnosis result, which denotes the student's knowledge proficiency.

5 Experiments

5.1 Dataset Description

Since there is no open dataset that could provide both watching video records and exercising records with text information, our experimental dataset is derived from MOOCCube [22], a large-scale data repository of over 700 MOOC courses, 100k concepts, 8 million student behaviors with an external resource. We extract 12 computer science courses. Some statistics of the dataset are shown in Table 1. To ensure the experimental results' reliability, we filter the students that both did less than eight exercises and answered less than two exercises incorrectly. We also filter the exercises and videos that no students have done or no contents. In total, we get 271,960 learning behavior records from 2,093 students, 519 exercises, 857 videos with 101 knowledge concepts for diagnostic networks.

Table 1. The statistics of the dataset.

Statistics	Value
# of students	26,969
# of exercises	1,074
# of videos	1,321
# of knowledge concepts	101
# of video records	2,680,833
# of exercise records	116,790

5.2 Experiment Setup

Framework Setting. We now specify the network initializations for LCD models. To extract the semantic representation for exercises and videos, we use the BERT-Base model to get embedding vectors with 768 dimensions. In course graph construction part, we set parameters $\alpha = 0.5$ and $\lambda = 6$. In the video and exercise representation learning part, we set the graph hidden size of 128. The size of the GCN output is 16. In the interaction layer part, The dimensions of the full connection layers are 64, 32, 1 respectively, $\beta = 16$. We initialize the parameters with Xavier initialization [5]. Besides, we set mini-batches as 32 for training and used dropout(with probability 0.6) to prevent overfitting.

Comparison Methods. To compare the performance of our proposed models, we borrow some baselines from various perspectives. Details are as follows:

IRT: A cognitive diagnostic model(CDM) that models student exercising records by a logistic-like function [3].

MIRT: A CDM that extends the latent trait value of each student in IRT to a multi-dimension knowledge proficiency vector with the Q-matrix [17].

NeuralCDM: A CDM that incorporates neural networks to learn the complex exercising interactions [20].

LCD-E: A variant of LCD framework. First, we construct an exercise graph(all nodes are exercises) to replace the course graph. Second, in the modeling process, we use a zero vector to represent video features instead of video embeddings learned by GCN.

In the experiment, we prepare 70% of the data as training data, 10% of the data as validation data, and the remaining 20% as testing data. All models are implemented by PyTorch using Python, and all experiments are run on a Linux server with four 2.4GHz Intel Xeon E5-2640 CPUs and a Geforce 1080Ti GPU.

5.3 Experiment Result

Student Performance Prediction. Here, we conduct experiments on the Student performance prediction task, i.e., predicting students' scores over each subjective or objective problem to indirectly evaluate the effectiveness of LCD, since we cannot obtain the real knowledge proficiency of students. Considering that all the exercises we used in our dataset are objective, we use evaluation metrics from both the classification and regression aspects, including accuracy, RMSE (root mean square error), and AUC (area under the curve).

Table 2. Experimental results on student performance prediction.

Model	Accuracy	RMSE	AUC
IRT	$63.1 \pm .1$	$46.7 \pm .1$	$60.3 \pm .1$
MIRT	$77.0 \pm .2$	$39.9 \pm .1$	$74.9 \pm .1$
NeuralCDM	$77.2 \pm .2$	$40.0 \pm .1$	$74.2 \pm .1$
LCD-E	$77.5 \pm .1$	$39.7 \pm .1$	$75.4 \pm .1$
LCD	$\mathbf{78.0} \pm \mathbf{.1}$	$\mathbf{39.5} \pm \mathbf{.1}$	$\mathbf{75.6} \pm \mathbf{.1}$

Table 2 shows the experimental results of all models on student performance prediction task. The error bars after '±' are the standard deviations of 5 evaluation runs for each model. We observe the performance in the following aspects:

For LCD-E method, it outperforms other baseline methods in terms of accuracy, RMSE, and AUC, which demonstrates that our method can make full use of the relationship between exercises to benefit the prediction performance.

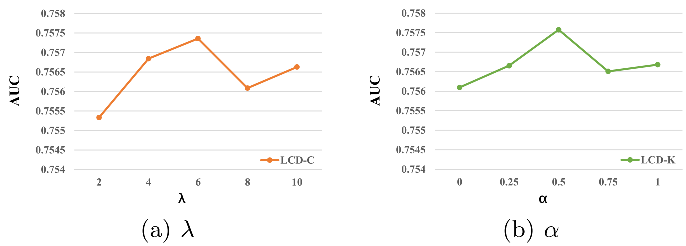
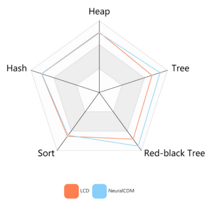


Fig. 4. The study of different graph methods with parameter influence.



Videos	Exercises	real	NeuralCDM	LCD
V1: Data Structure: IO of B Tree Concepts: IO, B Tree, Tree, heap, computation, V2: NULL V3: Data Structure: Exercise guidance: Sort Concepts: sort, class, arrange, complexity, computation, sequence, vector	E1: How to use the heap to sort… Concepts: heap	✓	✓	✓
	E2: S is the space of all possible entries, A is the space of all available addresses (|A| < |S|), h is a hash function, then… Concepts: hash	✗	✓	✗
	E3: And which sort has the asymptotic time complexity with it? A: Selection sort C: Bubble sort B: Insertion sort D: Merge sort Concepts: sort	✓	✗	✓

Fig. 5. Visualization of a student's proficiency on knowledge concepts and the responses of three exercises.

For LCD method, it generates better results than LCD-E, indicating the effectiveness of diagnosing student's knowledge states on both watching video behaviors and exercising behaviors than only considering exercising records.

Course Graph Analysis. We investigate the parameters that may influence the performance, including structure-based parameter λ in Eq. 1 and knowledge-based parameter α in Eq. 2.

Figure 4(a) visualizes the performance with the increasing values of λ from 2 to 10. As λ increases, the performances of LCD-C firstly increase but decrease when λ surpasses 6, and it gets the lowest performance while $\lambda = 2$ (each video and exercise only connects with neighbors). Figure 4(b) shows it gets the best performance when $\alpha = 0.5$, which demonstrates that the prerequisite relation has a positive but weak influence.

Case Study. Here, we present an example of the cognitive diagnosis of student knowledge proficiency. As shown in Fig. 5, the radar chart shows a typical student's concept proficiency report diagnosed by NeuralCDM and LCD. The right side is some student records in the online education system. V_i is the video with the same concept as E_i and the student learned before practicing E_i. We can observe that LCD predicts all three exercises correctly, but NeuralCDM gets wrong results on E_2 and E_3 because NeuralCDM does not consider the video information which the student watched. For E_2, although the results of the student diagnosis report by NeuralCDM and LCD are similar, the student does not watch the video about "hash", which makes him have a higher probability of getting a wrong answer. As for E_3, the student watched the video named "Data Structure: Exercise guidance: Sort" before answering E_3, which helps him familiar with the exercises about "sort". Thus he answers it correctly. From the case, we can see that LCD provides a right way for result analysis and model explanations, which is also meaningful in educational applications.

5.4 Discussion

From the experimental results, we can observe that LCD outperforms all baselines. The course graph analysis shows the influence of course structure and knowledge concepts. The case study indicates that LCD can get an interpretive analysis result, which better characterizes the students' knowledge states.

On the other hand, there are still some directions for improvement. Although the learning behavior-aware method we designed is effective, manually-labeled knowledge concepts exercises and videos relate to may be deficient because of inevitable errors and subjective bias. We will design an efficient algorithm in the future to extract knowledge concepts more accurately. Besides, inevitably, students gain and forget the knowledge they learn, especially in online self-learning circumstances. We will make an effort to design reward mechanisms while monitoring the behaviors of students automatically.

6 Conclusion

In this paper, to investigate the influence of student learning behaviors in the cognitive diagnosis task, we propose a learning behavior-aware framework modeling both student's exercising records and watching video records. First, we constructed a graph to build the relationship between videos and exercises and refined the representation by a graph convolutional network (GCN). Next, we propose a novel interaction function to explore complex relationships between students, exercises, and videos. Experimental results on a large scale real-world dataset validate the effectiveness and the interpretation of the proposed method. Promising future directions would be to investigate other important features (guessing features) in the student learning process.

Acknowledgement. This work is supported by the National Key Research and Development Program of China (2018YFB1005100). It also got partial support from National Engineering Laboratory for Cyberlearning and Intelligent Technology, and Beijing Key Lab of Networked Multimedia.

References

1. Cen, H., Koedinger, K., Junker, B.: Learning factors analysis – a general method for cognitive model evaluation and improvement. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 164–175. Springer, Heidelberg (2006). https://doi.org/10.1007/11774303_17

2. De La Torre, J.: Dina model and parameter estimation: a didactic. J. Educ. Behav. Stat. **34**(1), 115–130 (2009)
3. DiBello, L.V., Roussos, L.A., Stout, W.: 31a review of cognitively diagnostic assessment and a summary of psychometric models. Handb. Stat. **26**, 979–1030 (2006)
4. Embretson, S.E., Reise, S.P.: Item Response Theory. Psychology Press (2013)
5. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010)
6. Hornik, K., Stinchcombe, M., White, H., et al.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**(5), 359–366 (1989)
7. Hou, L., la Torre, J.D., Nandakumar, R.: Differential item functioning assessment in cognitive diagnostic modeling: application of the Wald test to investigate DIF in the DINA model. J. Educ. Measur. **51**(1), 98–125 (2014)
8. Huang, Z., Yin, Y., Chen, E., Xiong, H., Su, Y., Hu, G., et al.: EKT: exercise-aware knowledge tracing for student performance prediction. IEEE Trans. Knowl. Data Eng. **30**, 100–105 (2019)
9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
10. Liu, Q., et al.: Fuzzy cognitive diagnosis for modelling examinee performance. ACM Trans. Intelli. Syst. Technol. (TIST) **9**(4), 1–26 (2018)
11. Liu, Y., et al.: Collaborative learning team formation: a cognitive modeling perspective. In: Navathe, S.B., Wu, W., Shekhar, S., Du, X., Wang, X.S., Xiong, H. (eds.) DASFAA 2016. LNCS, vol. 9643, pp. 383–400. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-32049-6_24
12. Martin, A.J., Lazendic, G.: Computer-adaptive testing: implications for students' achievement, motivation, engagement, and subjective test experience. J. Educ. Psychol. **110**(1), 27 (2018)
13. Nakagawa, H., Iwasawa, Y., Matsuo, Y.: Graph-based knowledge tracing: modeling student proficiency using graph neural network. In: 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI), pp. 156–163. IEEE (2019)
14. Pan, L., Li, C., Li, J., Tang, J.: Prerequisite relation learning for concepts in MOOCs. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1447–1456 (2017)
15. Peng, H., et al.: Fine-grained event categorization with heterogeneous graph convolutional networks. arXiv preprint arXiv:1906.04580 (2019)
16. Piech, C., et al.: Deep knowledge tracing. In: Advances in Neural Information Processing Systems, pp. 505–513 (2015)
17. Reckase, M.D.: Multidimensional Item Response Theory Models, pp. 79–112. Springer, New York (2009). https://doi.org/10.1007/978-0-387-89976-3
18. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Trans. Neural Netw. **20**(1), 61–80 (2008)
19. Thai-Nghe, N., Schmidt-Thieme, L.: Multi-relational factorization models for student modeling in intelligent tutoring systems. In: 2015 Seventh International Conference on Knowledge and Systems Engineering (KSE), pp. 61–66. IEEE (2015)
20. Wang, F., et al.: Neural cognitive diagnosis for intelligent education systems. arXiv preprint arXiv:1908.08733 (2019)
21. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: 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)

22. Yu, J., 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)
23. Zhang, J., Shi, X., King, I., Yeung, D.Y.: Dynamic key-value memory networks for knowledge tracing. In: Proceedings of the 26th International Conference on World Wide Web, pp. 765–774 (2017)