2019 IEEE 5th International Conference on Computer and Communications

Learning Path Recommendation Based on Knowledge Tracing Model and Reinforcement Learning

Dejun Cai
Communication University of China
Beijing, China
e-mail: caidejun@cuc.edu.cn

Yuan Zhang and Bintao Dai
Communication University of China
Beijing, China
e-mail: {yzhang, daibintao}@cuc.edu.cn

***Abstract*—In recent years, studies on personalized learning path recommendation have drawn much attentions in E-learning area. Most of the existing methods generate the learning path based on learning costs that are formulated manually by education experts. However, this kind of learning costs cannot record the knowledge level change during the learning process and therefore does not accurately reflect the learning situation of the learner. To tackle this problem, we propose a knowledge tracing method which models learners' knowledge level over time, so that the learners' learning situation can be accurately predicted. Then, we propose a learning path recommendation algorithm based on the knowledge tracing model and Reinforcement Learning. A series of experiments have been carried out against learning resource datasets. Experiments results demonstrate that our proposed method can make sound recommendations on appropriate learning paths in terms of accuracy and efficiency.**

Keywords-personalized learning path; learning recommendation; knowledge tracing; reinforcement learning

I. INTRODUCTION

Recent years have witnessed the tremendous growth of online education. Although online learning can bring learners with convenience, there still exists some challenging problems such as information confusion. For example, the learners are exposed to a large number of learning resources and cannot clearly understand the direction of the next step in the learning process. Most of the current research works try to solve this problem by generating personalized learning paths. The personalized learning path guides students to learn according to the learner's different prior background, preferences, and various learning goals. Each learning path represents a set of KCs (i.e. knowledge components, which may include skills, concepts or facts) that are linked together based on some rules or constraints.

A general framework of learning path recommendation includes two parts: learning cost acquisition and learning path generation. In most cases, the learning costs are formulated by the educational expert at the start of learning, which doesn't change any more. Therefore, the researchers are more focused on the approaches to generate the optimal learning path given the learning costs. The main disadvantage of these approaches is that we must get the learning costs at first and the fixed costs set by experts does not reflect the actual learning situation of the learner. However, the demand for each learner of learning resources is different and always changing during the learning process. In short, the previous works [1][2][3] based on static learning costs cannot provide the learning path to be personalized with the learning process.

To tackle the above problem, we propose to utilize knowledge tracing to model the students' learning situations based on their historical learning trajectories. To generate the personalized learning path for each learner, we propose a reinforcement-learning-based method to recommend learning path. Overall, we name the proposed method as Knowledge Tracing based Knowledge Demand Model (KT-KDM), with which we can recommend the optimal learning path for learners according to the requirements of each knowledge point during the learning process. The main contributions of this paper include:

(1) Knowledge tracing model to reflect the learner's learning demand instead of manually made learning costs.

(2) Reinforcement Learning method to obtain the learning demand of knowledge by simulating the actual learning process, and recommend a personalized learning path.

(3) Demonstration of the superiority of dynamic updating of learning costs against the fixed cost in learning path recommendation.

II. RELATED WORK

A. Learning Path Recommendation

There are many kinds of personalized learning path recommendation algorithms. Based on the data mining method, Zhao X et al. [4] refer to the historical learning choice of similar learners, combine the KCs relationships from the existing knowledge graph, and generate the path. But the data mining way lacks the analysis of the learning resources themselves.

Therefore, some researchers make a recommendation from the perspective of learning resources. [2] [3] focus on the constraints of learning resources such as learning time, learning style, and learning difficulty, thus proposed a universal solution for the learning path. In addition, The intelligent optimization algorithms are applied to generate the path such as Particle Swarm algorithm [1], Bayesian Network method [2], and Immune algorithm [3].

978-1-7281-4743-7/19/$31.00 ©2019 IEEE

B. State-of-the-Art Knowledge Tracing Technology

Bayesian Knowledge Tracing (BKT) is the first model introduced in the e-learning area [5]. It applies the Hidden Markov Model to model the knowledge of students and use the transition probability and the initial learning state to determine whether the learner can learn the KC. Learning Factors Analysis (LFA) [6] is another traditional knowledge tracing model. LFA is a cognitive model that combines a statistical model, human expertise and a combinatorial search. Performance Factors Analysis (PFA) makes an improvement based on LFA, modifying the previous tracing model so that it can be used to select KCs adaptively [7].

C. Knowledge Tracing Based on Deep Learning

In recent years, deep neural networks have also been used in knowledge tracing and generally outperforms traditional methods. Piech et al. [8] proposed Deep Knowledge Tracing (DKT) based on Recurrent Neural Networks (RNN). It models the knowledge of students as they interact with coursework and capture more complex representations of student knowledge. However, the DKT model fails to reconstruct the observed input, and the predicted performance for KCs across time-steps is not consistent. Yeung et al. [10] improved the DKT model by adding regularization terms to solve reconstruction and volatility related problems without reducing prediction accuracy effectively.

III. PROPOSED METHOD

A. Algorithm Overview

The main idea of our proposed is to apply the knowledge tracing technology to model the learner's learning process and then generate a learning path with the reinforcement learning method.

The KT-KDM can be divided into two modules. One module is the Knowledge Demand Model (KDM), the other is Knowledge Tracing Model (KTM) – to characterize learner's learning requirements for different KCs. KTM models the learner learning situation in general based on their historical learning trajectories. And KDM obtains the learner's knowledge mastery degree provided by the KTM and recommends the KCs through predicting the level of knowledge requirements (LKR). Being trained with different feedback, the KDM can learn the relationships between recommendation action and learning effects, thereby determining whether the recommended KC are suitable for the current learner.

After a round of study by the learner, the KTM can provide the new KCs mastery degree. In this recommending process, the environment consists of KTM and learners, KDM is the agent, the learner's knowledge mastery degree is the state, and the recommended KCs is the action perform by the agent. Therefore, the problem is modeled as a Markov decision process, and the reinforcement learning method is applied to solve this problem. In Fig. 3, we show the interaction between the environment and agent. In the following, we will describe the knowledge tracing environment and learning path generation, respectively.

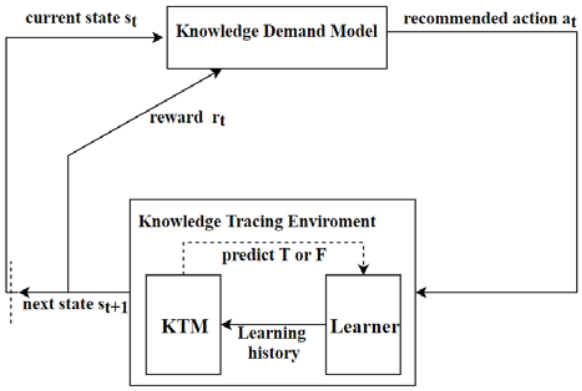

Figure 1. Markov process model of the recommendation learning ath.

B. Knowledge Tracing Model

In the knowledge tracing environment, the KTM interacts with the learners. KTM is mainly used to trace different learners' learning progress and provides corresponding observation data according to the learner's learning history. The learner learns and questions about the KCs recommended by KDM. The feedback of the learner after learning is also fed into KTM together with its previous learning history.

The KTM needs to be trained before it can be used online by real learners. Therefore, in the two stages of model training and online deployment, learners in the knowledge tracing environment have different definitions:

(1) In the training stage, the KTM simulates the real learner and interacts with the KDM. At this stage, the KTM is considered as a high-dimensional student dataset, because the KTM is trained by a mass of students learning log data. For a new recommended KC, the previous KTM acts as a real learner to make learning predictions. Then the new learning result is applied to update the KTM together with historical states, which will generate the next observation of the learning situation for KDM. For example, if the previous KTM records that the mastery degree for KC A is 0.7, then there will a 70% probability of studying this knowledge correctly, and the result is used to update the KTM.

(2) In the online deployment stage, the real user interacts with the model. The user obtains and learns the recommended KCs in the system and learns. The actual learning result is merged into the learning history, and passes it to the KTM for prediction.

The KTM applied in our approach is the regularized DKT+ model proposed by Yeung [10]. For the two advantages of regularized DKT+ are that: (1) regularized DKT+ provides better prediction accuracy and tracing results, compared with traditional knowledge tracing models BKT [5] and PFA [7]. (2) Regularized DKT+ can reconstruct the observed input, and the predicted output changes smoothly. Compared with the DKT, the regularized model is more correlated with the actual situation of students learning.

C. Learning Path Generation

1) Model

The learning path recommendation part is mainly composed of a knowledge demand model and a weighted random selection function. KDM is essentially a predictive network base on KTM's learning state. The output of the KDM is a vector where each entry represents the LKR of each KC, that is, the predicted recommended probability. Then the weighted random function selects a KC to recommend. In short, the generation part takes the learner's mastery degree as input, predicts the LKR of the learners, and makes the corresponding recommendations.

The base architecture of the KDM is a fully connected neural network. In the last layer of the KDM, softmax is added as the activation function and outputs the probability value of 0 to 1 as for each KC, and the sum of the probabilities is 1. The detail of the network is described in the simulation section.

2) Reward function

The reward of the reinforcement learning method is determined by the action that has been performed, the current state, and the state of the next step. For our learning path recommendation model, the reward value is defined as the encouragement taken by recommending the appropriate KC. Therefore, we design the reward function as the difference between the learner's post-learning mastery degree of the two adjacent recommended KCs. To avoid the model repeatedly recommending the same KC with high reward, we set a penalty for the already recommended KCs.

$$r\_t = \begin{cases} \frac{s_{k,t} - s_{k,t-1}}{n_{k,t}}, s_{k,t} - s_{k,t-1} > 0 \\ s_{k,t} - s_{k,t-1}, s_{k,t} - s_{k,t-1} \leq 0 \end{cases}$$

k epresents the index of latest recommended KC, $s_{k,t}$ represents the learning situation of KC k at time step t, $n_{k,t}$ represents the times the KC k has been recommended at time t.

3) Training method

Algorithm 1 A2C-based learning path recommendation model training process

1: Initialize the environment, get the state space and action space of the environment;
2: Initialize the Actor Policy Network and the Critic Value Estimation Network in A2C;
3: Learner initial learning state, s_0; Learning state of time t, s_t; Recommended KC of time t, a_t; Reward of time t, r_t;
4: **for** $episode = 0 \rightarrow nb\_episodes$ **do**
5: Initialize experience pool S,A,R of this episode
6: Get s_0
7: **while** during the experiment **do**
8: According to the prediction of Policy Network, access to the LKR of each knowledge, P_t
9: Use the weighted ranmdom function to select recommended KC a_t based on P_t
10: From the environment,get s_t, r_t, and determine whether the end of the experiment
11: Store (s_t, a_t, r_t) in S, A, R respectively
12: **end while**
13: Calculate the reward D after penalty, the estimate reward $\widehat{D}$ of the Value Estimation Network output, and calculate $TDerror = D - \widehat{D}$
14: Use $(S, A, TDerror)$ to update the policy network
15: Use (S, D) to update the value estimation network
16: Reset environment
17: **end for**
18: Close the environment

In our model, the observed state is the learning state generated by KTM. There are total N KCs, an N-dimensional vector represents the user learning state, and the value of each dimension represents the degree of the mastery of the KC, and the value range is [0,1], so the value of state space is continuous. However, the input action to the KTM environment represents a recommended KC, which is a discrete value with a range of [0, N-1].

For the problem of continuous state space, discrete action space, and limited experimental length, we apply a reinforcement learning method – Advantage Actor-Critic (A2C) to solve it. The reinforcement learning steps of the model are shown in Algorithm 1.

IV. EXPERIMENTS

A. Dataset

The dataset used to train KTM of our experiment is ASSITment 2009-2010[9], which is collected from the ASSISTment's online teaching system for teaching data from 2009 to 2010, with a range of mathematics in grades 4 to 10. After discarding invalid data, 328291 records were retained for the training dataset, including 4417 students and 110 KCs.

B. Implementation

The software and hardware environment of the simulation experiment are shown in Tab. I. To increase the accuracy of the test results and prevent the overfitting phenomenon caused by limited experimental data, we shuffled the datasets and randomly selected 70% as the training data each time. After five times, we got five knowledge of tracing models. The AUC score of the five models is 0.76232, 0.76835, 0.77319, 0.76560 and 0.76403.

TABLE I. THE SOFTWARE AND HARDWARE ENVIRONMENT

Software or hardware	**Configuration/version**
CPU	Intel(R) Xeon(R) CPU E5-1620
Memory	16GB
Operating system	Windows 10 Pro
Python	3.6.3
TensorFlow	1.12.0

Considering the KTM plays two important roles in our algorithm: (1) Learning state representation, which represents the learners' mastery degree. (2) Virtual learner, which acts as a real learner and interacts with the learning path recommendation model. Therefore, one of the models was selected to represent the learning state and was named KT1. Moreover, the remaining four models were used as virtual learners for testing and named KT2, KT3, KT4, KT5, respectively.

In the Tab. II, the core of the KDM architecture is fully connecting layer, and we applied the parameters provided by the Yeung [10] when training KTM, λ_r = 0.10, λ_{w1}=0.003, λ_{w2}=3.0.

TABLE II. LEARNING PATH MODEL NETWORK PARAMETERS

Parameter	value
Reward penalty(γ)	0.99
Learning rate	0.0001
Consecutive frames	20
Shared network structure	
layer 1(input layer)	dimension: equivalent to the number of KCs
layer 2(FC layer)	dimension: equivalent to the number of KCs, AF: relu
layer 3(FC layer)	dimension: 128, AF: relu
Policy network structure	
layer 1(FC layer)	dimension: 128, AF: relu
layer 2(output layer)	dimension: equivalent to the number of KCs, AF: softmax
Value estimation network	
layer 1(FC layer)	dimension: 128, AF: relu
layer 2(FC layer)	dimension: 1, AF: linear

C. *Reference Algorithm*

We compare the proposed learning path recommendation algorithm with two benchmark algorithms.

(1) Greedy algorithm

The greedy algorithm takes KTM to simulate all the actions of recommended KCs and selects the KC that students will get a higher total mastery degree after learning.

(2) Random algorithm

The random algorithm guides the learner to make random learning choice. It regards the learning states given by knowledge tracing environment and recommends the KCs from all learning resources.

D. *Results*

KT1 is the state representation model, which is tested by the other four KTM mentioned above. For each model, we perform ten experiments and take the mean value of the results. The maximum learning length for each experiment is set to 300.

The two metrics are the average reward and the total mastery degree. The value of the average reward is the average for each recommendation round of rewards, which characterizes the progress of the learning process. The higher the average reward value, the more significant the learners' progress has made. The value of total mastery degree is the summary of the predicted probability of all KCs for whether the learner can mastery them correctly. It reflects the overall learning situation after learning of the learners, which is the higher, the better. The results of the two metrics are as shown in Tab. III.

According to the results, our KT-KDM method gets better performance than the random algorithm in average reward and total mastery degree and performs strictly to the result of the greedy algorithm. Since the greedy algorithm performs all possible actions and selects the choice that leads to the highest mastery degree value, so the result of the greedy method always performs better. The main drawback of the greedy algorithm is its higher computing cost. Because the algorithm needs to traverse the KTM to obtain the learning effect with every recommendation situation and make the optimal choice, it takes much running time than the proposed algorithm for its computational complexities are numerous. Finally, in the operating environment shown in Tab. I, the comparison of the running time among three algorithms is shown in Tab. IV.

In Tab. IV, the greedy algorithm takes much time in computing for its complexity. the KT-KDM takes a much less running time than the greedy algorithm, and the running time is close to the random algorithm. But the recommended performance of the KT-KDM verge on the greedy algorithm. Comprehensively considering the running time and recommendation efficacy, the KT-KDM is practical and effective.

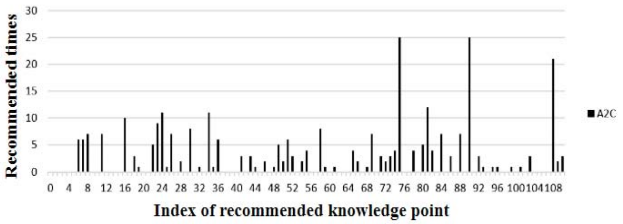

Figure 2. Statistical analysis of the recommended KCs in a learning path.

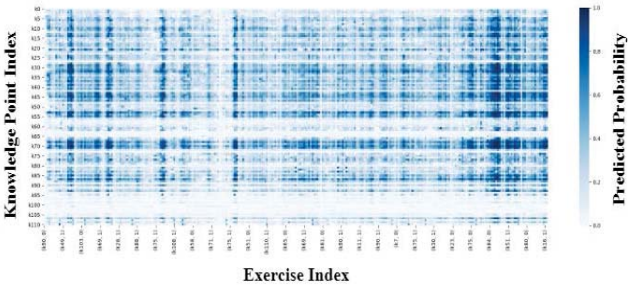

Figure 3. The heatmap records the learners mastery degree of each KC during the training process.

Besides, the KCs recommended are supposed to cover all the learning resources as much as possible in the learning process, instead of just recommending several of them. Based on this, we made a statistical analysis of the KCs in the whole learning process. In Fig. 2, the KCs recommended in the entire process cover most of the KC, and the distribution of recommended times is relatively uniform. Although there are a few KCs that have never been recommended or recommended more often, the absence is the same as the student actual learning sequence in the original dataset.

TABLE III. THE RESULT OF LEARNING ALGORITHMS

Algorithm	Average of reward					Total mastery degree				
	KT2	KT3	KT4	KT5	MEAN	KT2	KT3	KT4	KT5	MEAN
KT-KDM	7.135	8.098	4.899	5.786	6.480	72.193	74.611	70.786	70.905	72.124
GA	2.883	9.513	8.723	7.856	7.244	69.225	75.572	75.143	69.974	72.479
RA	6.116	4.923	4.272	5.186	5.124	62.802	66.899	62.421	67.867	64.997

TABLE IV. RUNNING TIME COMPARISON OF THREE ALGORITHMS

algorithm	**KT-KDM**	**GA**	**RA**
Total running time(s)	60.79	12468.47	57.13
Average time per episode(s)	6.08	1246.85	5.71
Average time per round(s)	20.26	4156.16	19.04

In order to visualize the learning situation, we draw the heatmap that records the mastery of each KC during the training process. In Fig. 3, Exercise Index is the abscissa. The index of KC is the ordinate. The brightness of the blue color in the heat map represents the mastery of knowledge. The darker the color, the better the student masters this KC. In the initial stage, the mastery degree fluctuates sharply in the early stages of training. After a period of training, the learner's knowledge mastery has steadily increased, and then the learners' mastery degree has been dramatically improved compared with the start stage.

These experiments have confirmed that the proposed method can effectively improve their learning efficacy for the learners who are modeled by KTM. The model can represent the learner's learning requirements and recommend a suitable learning path without requiring the existing learning costs. Moreover, the method can take less computing time to generate a leaning path that covers most key knowledge.

V. CONCLUSION AND FUTURE WORK

In this paper, we have presented an integral approach of recommending learning path based on the knowledge tracing method and reinforcement learning method. Based on the theme of modeling the learners, we apply knowledge tracing method to tracing the learning process dynamically. Moreover, we take the reinforcement learning method to build a model determining what to learner according to the knowledge tracing result. Our method can dynamically represent the learner's learning requirements and provide a personalized learning path, which is more in line with the learner's actual learning situation.

Our study establishes a novel framework for learning path recommendation, which skips the steps of learning costs acquisition. The results of this investigation show that our method can improve the learners' learning efficiency, which made an excellent performance in recommending.

A limitation of this study is that KTM and KDM are independent of each other, resulting in the inability to train at the same time. In fact, we must start training KDM after the training of KTM is completed. The online learning that trains both models at the same time remains the future work.

REFERENCES

[1] Wu L, Fang Q, "Learning Path Optimization Based on Improved Particle Swarm Optimization Method," Journal of Systems Science and Mathe matical Sciences. 2016(12): 2272-2281.

[2] N. Anh, V. H. Nguyen, and H. SI DAM, "Constructing a Bayesian belief network to generate learning path in adaptive hypermedia system," Journal of Computer Science and Cybernetics, vol. 24, Jan. 2008..

[3] C. Bian, S. Dong, C. Li, Z. Shi, and W. Lu, "Generation of adaptive learning path based on concept map and immune algorithm," 2017 12th International Conference on Computer Science and Education (ICCSE), 2017, pp. 409–414..

[4] Zhao X, Xu D, Long S, "Collaborative Recommendation: A New Perspective for Personalized Learning path Generation," Distance Education in China, 2017, 2017(5): 24-34.

[5] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," User Modeling and User-Adapted Interaction, vol. 4, no. 4, pp. 253–278, 1994.

[6] H. Cen, K. Koedinger, and B. Junker, "Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement," in Intelligent Tutoring Systems, 2006, pp. 164–175.

[7] P. I. Pavlik, H. Cen, and K. R. Koedinger, "Performance Factors Analysis -A New Alternative to Knowledge Tracing," in Proceedings of the 2009 Conference on Artificial Intelligence in Education ,2009, pp. 531–538.

[8] C. Piech et al., "Deep Knowledge Tracing," Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1. Cambridge,MA,USA:MIT Press, 2015:505-513.

[9] ASSISTments Data. (2015). Retrieved March 07, 2016, from https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data/skill-builder-data-2009-2010

[10] C.K. Yeung and D.Y. Yeung, "Addressing Two Problems in Deep Knowledge Tracing via Prediction-Consistent Regularization," arXiv:1806.02180 [cs], Jun. 2018.